计算机科学与技术学院

*schol of Computer Science & Technology*

2014 年 度 研 究 生

# 核心期刊 论文集

# 前言

为体现苏州大学计算机科学与技术学院研究生的风采，展示我院研究生的学术成果，方便大家学习与交流，我院研究生会对计算机科学与技术学院的研究生发表在 2014 年度 SCI，中国计算机协会（CCF）及学院认定的一类核心期刊的论文进行了收集并汇编成册。

本论文集共收录了优秀论文 26 篇，并且按照我院各课题组研究方向进行了论文甄选和分类。

本论文集得到了学院领导的亲切关怀和支持，同时也得到了有关老师和研究生的鼎力帮助与指导，在此向各位老师和同学表示衷心的感谢！

计算机科学与技术学院研究生分会

2015 年 4 月

# 目录

## Section – 1 高性能计算



## Section – 2 机器学习理论及应用

## Section – 3 中文信息处理与自然语言理解



## Section – 4 先进数据分析

| | | | |
|---|---|---|---|
| An Evolution-Based Robust Social Influence Evaluation Method in Online Social Networks | Feng Zhu, Guanfeng Liu, Lei Zhao, Xiaofang Zhou | WISE | 179-195 |

## Section – 5 软件形式化和自动推理

| | | | |
|---|---|---|---|
| Study of Active Learning-based Trademark Number Recognition Method | Yujie Shi, Jian Wu, Victor S. Sheng, Zhiming Cui, Pengpeng Zhao | Journal of Algorithms & Computational Technology | 196-209 |

## Section – 6 图像处理与模式识别

| | | | |
|---|---|---|---|
| Improved packing of protein side chains with parallel ant colonies | Lijun Quan, Qiang Lu, Haiou Li, Xiaoyan Xia, Hongjie Wu | BMC Bioinformatics | 210-221 |

Journal of Computational Information Systems 10: 2 (2014) 747–754
Available at http://www.Jofcis.com

# Research on Verification of Properties for CPS Based on Statistical Model Checking ⋆

Mingcai CHEN [1,2], Guangquan ZHANG [1,3,*], Hui WEI [1], Yuzhen SHAO [1], Chengkai XU [1], Linfeng ZHENG [1]

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*

[2] *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China*

[3] *State Key Laboratory of Computer Science, Chinese Academy of Science, Beijing 100190, China*

**Abstract**

Cyber-Physical System (CPS) is a complex network system with the integration of computation and physical process. CPS software have many characters such as openness, tightly coupling with hardware, real-time, resulting in the verification of the key properties of CPS as a major challenge. This paper propose a new approach to verify the properties of CPS software. First, CPS software are modelled by using the network of extended hybrid automation. Then, we construct the sample of CPS model for statistical model checking, and then verify the properties of CPS model using statistical model checking. This method can avoid the problem of state space explosion effectively. At last, a case study is given to illustrate the validity of this method.

*Keywords*: Cyber-Physical System; Statistical Model Checking; Verification; Hybrid Automata

## 1 Introduction

With the development of embedded technology, computer and network technology, as well as hardware performance and data processing capabilities on the rise, computer systems become more information and intelligence. Cyber-Physical Systems (CPS) come into being as a new intelligent system which has attracted high attention of governments, academia and industry.

CPS is a complex embedded network system that combines computing and physical process. CPS will be widely used in monitoring and control of critical infrastructure, national defense systems, health care, intelligent transportation and so on, therefore ensuring the safety and reliability of CPS is important. The physical world, however, is not entirely predictable, as well as

⋆Project supported by the Natural Science Foundation of Jiangsu Province (BK2011281), Applied Foundation Research Program of Suzhou (SYG201241), Post Graduate Research and Innovation Program of Jiangsu Province (CXLX12_0809, CXLX13_820), the Students' Innovation and Entrepreneurship Training Program of Jiangsu Province (2012yb010), the Students' Academic Research Fund of Soochow University (KY2013053A).

*Corresponding author.
*Email address:* gqzhang@suda.edu.cn (Guangquan ZHANG).

1553–9105 / Copyright © 2014 Binary Information Press
DOI: 10.12733/jcis9617
January 15, 2014

device failure, the robustness and security of the entire CPS system is challenging [1]. Formal verification can guarantee the reliability of the system to a certain extent, which verifies the system in the design stage to find errors earlier. Model checking is a common formal methods, with the characteristics of high degree of automation. Model checking verifies whether the system satisfies the specification by traversing the state space of the system, limited by the size of state space. While the size of states space for CPS can be enormous in reality, coupled with the uncertainty of physical environment and the device failure. Under that, traditional model checking techniques for CPS verification is powerless. Statistical model checking (SMC) [2] is a new technology to verify large complex systems. The core idea of SMC is constructing the simulation of the execution of system, and then determines whether the system satisfies specific properties with a confidence by adopting statistical evaluation method.

Edward A. Lee led the Ptolemy project [3], study concurrent real-time, embedded systems modeling, simulation and design. What the project main concern is the concurrent combination of the components, and to solve the problem of heterogeneous computing model mix through the hierarchical combination of multiple computation models. But the work focused on modeling and simulation of the CPS, less involved in the verification of the properties of the CPS.

SMC was originally proposed by Younes, acceptance sampling was used to verify the properties of discrete event systems [4]. The error control of SMC was discussed and statistical verification for unbounded until operator property was studied in [5]. Bayesian statistics was used in SMC of stochastic systems in [6]. In [7], cross-entropy techniques and importance sampling techniques was used to study SMC in order to solve the problem of computation time explosion caused by rare events.

Those efforts focus on studying CPS modeling and simulation, however, less for verification of CPS properties. According to those algorithm for SMC, the model use mostly CTMC, DTMC and MDP. However, due to the characteristics of CPS, those models are not well expressing CPS. Compared to those work, a new approach to CPS properties validation is proposed in this paper. Section 2 presents essential knowledge of CPS properties verification. Section 3 describes statistics method of CPS properties verification. Section 4 gives a case study to illustrate the effectiveness of the proposed method. Section 5 concludes and proposes future research directions.

## 2 Extended Hybrid Automation

Hybrid automata is a formal model of description both discrete and continuous dynamic systems. Hybrid automata is the expansion based on the finite automata with a set of variables, its location for the continuous evolution and its transition for discrete change of states. As depict in Fig. 1, it usually uses directed graphs to represent a hybrid automata, the vertex for location, edge for a discrete transition.

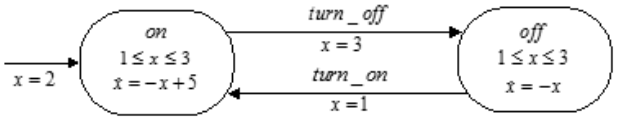



Fig. 1: Thermostat hybrid automaton model

Here is the basic definition of hybrid automata.

Table 1: The operations of port

| | |
|---|---|
| datatype Read( ) | void Write(datatype x) |
| { | { |
| return value; | value = x; |
| } | } |

**Definition 1** *hybrid automation machine HA is a six-tuple* $HA =< Loc, Var, Lab, E, Act, Inv >$.

A finite set $Loc$ of vertices, called locations. A finite set $Var$ of real-value variables. A valuation of variable $x$ is a function $v(x)$: $Var \rightarrow R$, map each variable to a real-value. A pair $< l, v >$ present for a state of hybrid system, where $l \in Loc$, $v \in V$. $\sum$ is used to present the set of all the states. A finite set $Lab$ of synchronization labels. A finite set $E$ of edges. A edge $e =< l, a, \mu, l' >$ contains a source location $l \in Loc$, a target location $l' \in Loc$, a synchronization label $a \in Lab$, and a transition relation $\mu \in V \times V$. A labeling function $Act$ that assigns each location $l \in Loc$ to a set of activities, $Act(l)$: $R^{\geq 0} \rightarrow V$ that maps time t to valuation $V$. A labeling function $Inv$ that assigns each location $l \in Loc$ to a invariant $Inv(l) \in V^2$.

CPS components are not completely isolated but a unified whole interrelated. Therefore, in order to describe the CPS more accurately, classic hybrid automata need to be extended. This paper introduces the concept of communication port to link different parts CPS together.

**Definition 2** *Communication port Port is a tuple* $Port =< pid, value, dom, options >$.

*pid* is the unique identifier of port. The system references the function of port through *pid*. *value* is the data of port. *dom* is data type of *value*. A finite set *options* that contains the operations allowed by port. As show in Fig. 1, the port is allowed to execute two kind of operations, *Read* and $Write$.

For example, the end to sensor communication port connected to the sensor is allowed to run both *Read* and $Write$. The other end connected to the computing unit is only allowed to run *Read*.

**Definition 3** *Extended hybrid automata is a tuple* $EHA =< HA, P >$.

The definition of $HA$ is the same as definition 1, $P = \{p_1, p_2, \cdots, p_n\}$ is a finite set of ports related with $HA$.

The core idea of SMC is that it determines whether the system hold the specific properties with a certain degree of confidence using statistical evaluation techniques on the basis of the sample of system through simulation of system. The hybrid automation used here is defined below, i.e. the sample of system implementation.

**Definition 4** *A run trace of hybrid automata is a finite or infinite sequence like this:*

$$\sigma_0 \mapsto \sigma_1 \mapsto \sigma_2 \mapsto \cdots$$

Where $\sigma_i =< l_i, t_i >$, $l_i \in Loc$, $t_i \geq 0$ present the time system stay in $l_i$.

The algorithm of SMC for CPS software is specifically addressed in the next section.

# 3 Statistical Verification for CPS Model

The problem of stochastic system $M$ about temporal logical formula $\phi$, i.e, is to compute the probability of $M \vDash \phi$. The existing methods to solve such problem are mainly two kinds: numerical methods and statistical methods, such as SMC. Numerical methods typically have high accuracy, but subject to the restriction of the size of the state space. SMC treats this as the problem of statistical inference, and solves the problem using reasonable sampling from the model simulation path, to avoid restrictions on the size of the state space. The statistical validation framework of CPS model is shown in Fig. 2.

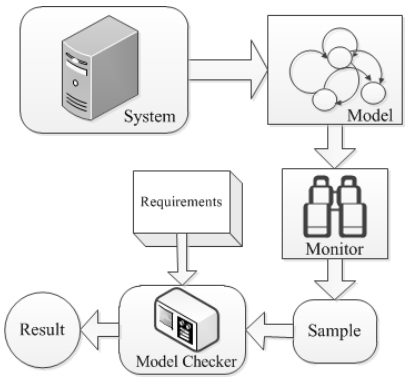



Fig. 2: A validation frameworks of CPS software properties

## 3.1 Property specification language

Probabilistic linear temporal logic (PBLTL) formula is used to describe the properties, which is the extensions of the LTL.

With the model $M$ and the set $Var$ of real-value variables, a Boolean predicate over $Var$ is a constraint of the form $y \sim v$, where $y \in Var$, $\sim \in \{\leq, \geq, =\}$ and $v \in R$.

**Definition 5** *The property formula is defined as fellow:*

$$\phi ::= y \sim \ v | (\phi_1 \vee \phi_2) | (\phi_1 \wedge \phi_2) | \neg\phi | (\phi_1 U^t \phi_2)$$

The semantics of the formula BLTL is defined on the execution traces of hybrid automata. That a run trace of $M$ hold the property $\phi$ is denoted by $\sigma \vDash \phi$. In this paper, the suffix of a trace $\sigma$ started at $i$-th step is denoted by $\sigma^i$. Especially, $\sigma^0$ denotes the origin trace $\sigma$. The value of variable $y$ in trace $\sigma$ at $i$-th step is expressed as $V(\sigma, i, y)$. The semantic of BLTL can be interpreted as fellow.

(1)$\sigma^k \vDash y \ \sim \ v$ if and only if $V(\sigma, k, y) \sim \ v$.

(2)$\sigma^k \vDash \phi_1 \vee \phi_2$ if and only if $\sigma^k \vDash \phi_1$ or $\sigma^k \vDash \phi_2$.

(3)$\sigma^k \vDash \phi_1 \wedge \phi_2$ if and only if $\sigma^k \vDash \phi_1$ and $\sigma^k \vDash \phi_2$.

(4)$\sigma^k \vDash \neg\phi_1$ if and only if $\sigma^k \vDash \phi_1$ does not hold (i.e. $\sigma^k \nvDash \phi_1$).

(5)$\sigma^k \vDash \phi_1 U^t \phi_2$ if and only if there exists $i \in N$ such that (a) $\Sigma_{0 \leq l \leq i} t_{k+1} \leq t$, (b) $\sigma^{k+i} \vDash \phi_2$, (c) for each $0 \leq j \leq i$, $\sigma^{k+j} \vDash \phi_1$.

Since it is impossible to obtain a sample unlimited executed, and that only limited prefix in an execution trace can determine the trajectory meets the properties formula has been demonstrated in [6]. The trajectory prefix length is decided by the boundary of the property formula.

**Definition 6** *The boundary of formula BLTL is defined as follows.*

$\#(y \sim v) := 0$ $\#(\neg\phi_1) := \#(\phi_1)$

$\#(\phi_1 \vee \phi_2) := \max(\#(\phi_1), \#(\phi_2))$

$\#(\phi_1 \wedge \phi_2) := \max(\#(\phi_1), \#(\phi_2))$

$\#(\phi_1 U^t \phi_2) := t + \max(\#(\phi_1), \#(\phi_2))$

On the basis of those above, the definition of PBLTL formula is introduced below.

**Definition 7** *PBLTL formula is a form of the equation $P_{\geq\theta}(\phi)$, where $\phi$ is BLTL formula, $\theta \in (0,1)$ is known as the probability threshold.*

That the model $M$ satisfies the PBLTL property $\phi$ can be denoted by $M \vDash P_{\geq\theta}(\phi)$. The formula $M \vDash P_{\geq\theta}(\phi)$ holds if and only if the probability that a run trace of $M$ satisfies $\phi$ is greater or equal to $\theta$. Here only discuss the case of relationship ($\geq$), greater or equal, and the relationship ($<$) can get through equation $P_{<\theta}(\phi) = 1 - P_{\geq\theta}(\phi)$.

## 3.2 The run sample of CPS model

SMC uses the run trajectory of the system model as an input in statistical verification phase. So how to get the implementation of the model sample is a critical issue in SMC. A sample of CPS model execution is depicted as the following form.

$$(s_0, t_0), (s_1, t_1), (s_2, t_2), \cdots$$

Where $s_i =< l_i, v_i >$ present for a state of CPS, $t_i$ for the time system stay in $s_i$.

This paper constructs a monitoring component to retrieve samples of the system model execution trace. The monitoring component records the changes of system state by monitor the entire discrete events in system. The monitoring component generates algorithm as follows.

## 3.3 Verification of CPS model

With successfully applied in many fields, SMC received a lot of attentions. It tries to calculate probability which any execution trace of automata meets the PBLTL property $p$. There exist two core Bayesian methods proposed in [6]: interval estimation and hypothesis testing. The difference between the two methods with the traditional model checking is that the trajectory which not meet the property $\phi$ is not a counter-example of the model but the evidence of $p < 1$. In this paper, we apply Bayesian statistical model checking to the CPS model verification.

Assume CPS model is denoted by $M$, $\sigma$ for a run trace of $M$, $\phi$ for the under-validate property formula, $\theta \in (0,1)$ for the probability threshold, $p$ for the probability of a trace of $M$ satisfies

Table 2: CPS model sample algorithm

| Input: Extended hybrid automation M, the length of trace n<br>Output: A sample for the run of M |
|---|
| (1) $\sigma$:= <>; |
| (2) s:=<>; |
| (3) v:=<>; |
| (4) clock t; |
| (5) while length($\sigma$) < n do |
| (6) t:=0; |
| (7) wait for an event happened; |
| (8) l:= the location of current; |
| (9) v:= the value of variant of M; |
| (10) s:=<l,v>; |
| (11) $\sigma := \sigma\bullet < s, t >$; |
| (12) end while |
| (13) return $\sigma$; |

Table 3: the algorithm of SMC for CPS model

| Input: Model M, $\phi$, $\theta$, $\alpha$, $\beta$, $T$<br>Output: The verification result |
|---|
| (1) n:=0; |
| (2) x:=0; |
| (3) loop |
| (4) $\sigma$:= draw sample trace from M; |
| (5) if $\sigma \models \phi$ then |
| (6) x:=x+1; |
| (7) end if |
| (8) beta:= BayesFactor(n,x); |
| (9) If beta > T then |
| (10) return accept H0 |
| (11) else if beta < 1/T then |
| (12) return accept H1; |
| (13) end if |
| (14) end loop |

property formula $\phi$. And then the validation problem to solve can denoted by $M \vDash P_{\geq\theta}(\phi)$, i.e, $M \vDash P_{\geq\theta}(\phi)$ holds when $p \geq \theta$, otherwise not hold. Then two mutex hypotheses are given out.

$$H_0 : p \geq \theta \text{ and } H_1 : p < \theta$$

With a sample $d = \{\sigma_1, \sigma_2, \sigma_3, \cdots\}$ of CPS model, random variable $X_i$ denote the result whether the trace $\sigma_i$ satisfy $\phi$ or not, its value as 1 or 0.

$$X_i = \begin{cases} 1 & \sigma_i \vDash \phi \\ 0 & \sigma_i \nvDash \phi \end{cases}$$

Since those traces are from the same model, a set of independent and identically distributed observations $\{x_1, x_2, x_3, \cdots, x_n\}$ can be gained. In that the assumptions mutual exclusion, assuming the prior probability satisfy $P(H_0) + P(H_1) = 1$. According to the theory of Bayesian posterior probability $P(H_i|d) = \frac{P(d|H_i)P(H_i)}{P(d)}$, $i \in \{0, 1\}$. For each sample $d$, $P(d) = P(d|H_0) + P(d|H_1) > 0$ always holds.

**Definition 8** *The Bayes factor of samples d and assumptions, $H_0$ and $H_1$, is defined as* $\beta = \frac{P(d|H_0)}{P(d|H_1)}$.

The Bayesian factor $\beta$ can be regarded as evidence in support of $H_0$, and its reciprocal $\frac{1}{\beta}$ as the evidence in support of $H_1$. Assume $T$ is the threshold to accept the evidence of assumptions. An efficient method to calculate proposed in [6].

Where $\sigma \models \phi$ can be easily verified using traditional methods.

# 4 Case Study

In this section, the speed control subsystem in automobile is taken as an example, model the automobile speed control subsystem software with extended hybrid automaton model, and verify the key property of the system with SMC. The automobile speed control subsystem application scenario is shown in Fig. 3. Users send a throttle command to the engine, and then the engine outputs a rotational speed to the actuator. And the actuator will convert the rotational speed of the engine to the output torque which can drive the automotive movement. The throttle sensor real-time transfers status information of throttle to drive controller. The vehicle speed sensor real-time report speed to the drive controller, and then the drive controller issues a control command to the controller to change the running speed of the vehicle according to the throttle command and the current vehicle speed and the control rule.

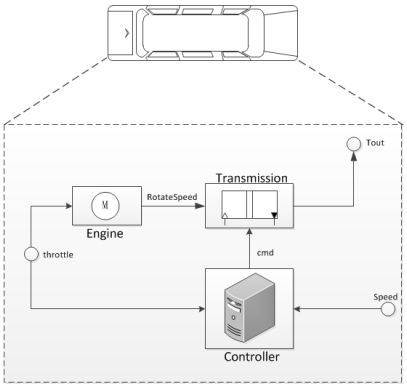



Fig. 3: The application scene graph of Automobile speed control subsystem

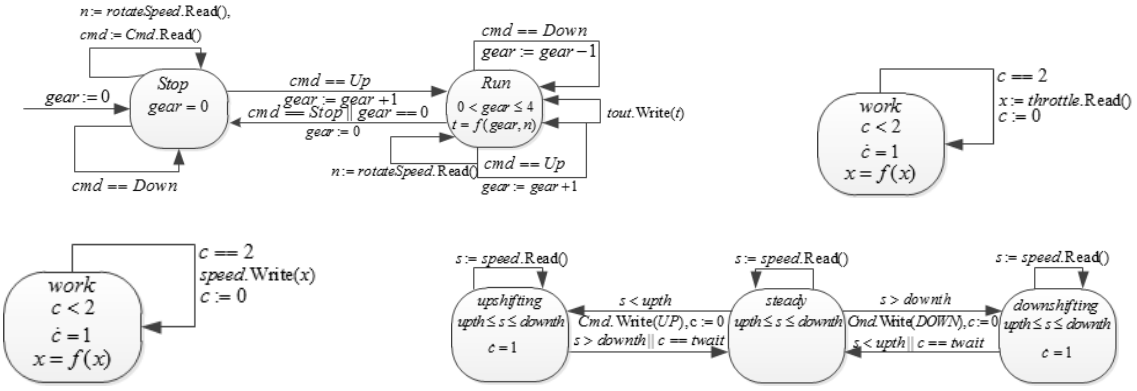



Fig. 4: (a) Transmission services; (b) Throttle services; (c) Engine services; (d) Drive control service

The vehicle speed control subsystem model is shown in Fig. 4. Security of the automobile speed control subsystem is critical, whether it is reliable is directly related to the life and property safety of the driver, especially his braking performance. The problem which this section attempts to verify is that when the automobile is traveling at normal speed, the probability which the speed of the car is reduced to 0 is at least 98% within 2.5s after the driver command emergency brake. Namely the assumptions are $H_0 : M \vDash P_{\geq\theta}(F^t(\neg(cmd = -3)\ U^{2.5} s = 0))$. The above results are based on the verification results obtained in the interval length of 0.04 and 0.98, the condition of the degree of confidence.

Table 4: The statistical verification results of Automobile speed control subsystem

| $\theta$ | 1.00 | 0.99 | 0.98 | 0.97 | 0.96 | 0.95 |
|---|---|---|---|---|---|---|
| Probability | [0.90,0.94] | [0.88,0.92] | [0.86,0.90] | [0.84,0.88] | [0.82,0.86] | [0.80,0.84] |
| hold/total | 788/857 | 740/822 | 722/821 | 683/794 | 617/735 | 589/719 |

# 5 Conclusion

Because of the complexity of CPS, the verification problem of CPS software properties has been well received attention by the academic. Compared with traditional software, CPS software has its own characteristics: tightly coupled with the hardware and openness. Because of huge system state space, verifying CPS software with the traditional formal verification method will inevitably face the system state space explosion. In this paper, the network of extended hybrid automation is taken as CPS software model and verify its key properties with SMC, which effectively avoids the risk of the state space explosion and provides a new way to CPS software properties verification. In future work, it may be considered that combining the traditional method and statistical model checking with some strategies can make the new method own characteristics of the high accuracy of the numerical methods and statistical methods, without limit of the size of state space.

# References

[1] K. Zhang, G. Zhang, M. Chen, et al. Model supporting CPS software system evaluation on trustworthiness [J]. Journal of Computational Information Systems, 8 (16), 2012, 6773-6780.

[2] K. G. Larsen. Statistical model checking, refinement checking, optimization, ... for stochastic hybrid systems [C]. Proceedings of 10th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS), 2012, 7-10.

[3] P. Derler, E. A. Lee, A. S. Vincentelli. Modeling cyber-physical systems [J]. Proceedings of the IEEE, 100 (1), 2012, 13-28.

[4] H. L. S. Younes, R. G. Simmons. Probabilistic verification of discrete event systems using acceptance sampling [C]. Proceedings of 14th International Conference on Computer Aided Verification (CAV), 2002, 223-235.

[5] H. S. Younes. Error control for probabilistic model checking [C]. Proceedings of 7th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI), 2006, 142-156.

[6] A. P. Paolo Zuliani, Edmund M. Clarke. Bayesian statistical model checking with application to Stateflow/Simulink verification [C]. Proceedings of Hybrid Systems: Computation and Control, 2010, 243-252.

[7] E. M. Clarke, P. Zuliani. Statistical model checking for cyber-physical systems [C]. Proceedings of 9th International Symposium on Automated Technology for Verification and Analysis (ATVA), 2011, 1-12.

# PTAS for Minimum *k*-Path Connected Vertex Cover in Growth-Bounded Graphs

Yan Chu, Jianxi Fan*, Wenjun Liu, and Cheng-Kuan Lin

Soochow University, School of Computer Science and Technology,
215006 Suzhou, China
{20124227005,jxfan,20114027003,cklin}@suda.edu.cn

**Abstract.** In the paper, we present a polynomial-time approximation scheme (PTAS) for the minimum *k*-path connected vertex cover (M*k*PCVC) problem , which can be used to solve security problems in wireless sensor networks (WSNs), under fixed $k \geq 2$. In contrast to previously known approximation schemes for M*k*PCVC problem, our approach does not need location data of the vertices, and it can be applied to growth-bounded graphs. For any $\varepsilon_1>0$, the algorithm returns a $(1+\varepsilon_1)$-approximation M*k*PCVC. We have proved the correctness and performance of the algorithm and shown its runtime is $r \cdot n^{O(f(r))}$, where $f(r)$ is a polynomial function, $r = O((1/\varepsilon)\cdot\ln(1/\varepsilon))$ and $\varepsilon$ is only dependent on $k$ and $\varepsilon_1$.

**Keywords:** PTAS, k-path connected vertex cover, growth-bounded graphs, bounded degree, wireless sensor networks.

## 1 Introduction

It is required to ensure secure communications in most of the applications in wireless sensor networks (WSNs). However, conventional security techniques cannot be directly employed in WSNs, because sensor nodes have limited computation, power and communication capabilities, and they are usually deployed in accessible areas, where they can be easily captured or attacked [1-2]. Furthermore, it is unrealistic to make all nodes anti-tamper because of the high costs [1-2]. Hence, it is a challenge to design security protocols for WSNs.

The Canvas scheme [3-9] can provide data origin authentication in a sensor network. The *k*-generalized Canvas scheme [6] improves the Canvas scheme and ensures data integrity in the communication graph as long as at least one node on each path of the length *k*-1 is not captured. There should be two different kinds of sensor devices – protected and unprotected, and it is more difficult to capture or attack a protected one. Hence, when deploying and initializing a sensor network, it is necessary to put at least one protected node on each path of the length *k*-1 [6]. Thus, we can reduce the costs by minimizing the number of protected nodes [6], which can be abstracted as follows:

* Corresponding author.

X.-h. Sun et al. (Eds.): ICA3PP 2014, Part I, LNCS 8630, pp. 114–126, 2014.

We call $P$ a $k$-path when the path $P$ contains $k$ vertices. Given a graph $G = (V, E)$ and a fixed integer $k \geq 2$. Let $C$ be a subset of $V$, we call $C$ a $k$-path vertex cover if each $k$-path in $G$ contains at least one vertex in $C$. The minimum connected $k$-path vertex cover (M$k$PCVC) problem asks to find the minimum cardinality of a connected $k$-path vertex cover in $G$ [1-2].

## 2 Related Work

M$k$PCVC problem has been studied a lot. Boštjan et al. [1] proved that the minimum $k$-path vertex cover (M$k$PVC) problem is NP-complete for any fixed integer $k \geq 2$. Then Tu and Zhou [10] gave a 2-approximation minimum 3-path vertex cover. Liu et al. [2] pointed out that M$k$PCVC problem is NP-complete and gave a $(1+\varepsilon)$-approximation M$k$PCVC for any fixed integer $k \geq 2$. Liu et al. [2] needed location information of all the vertices and repeatedly used grid-based separation and shifting strategy to get the final result. However, approaches based on shifting strategy are inherently central and can not be efficiently adapted to distributed works [11].

In the absence of position information, Nieberg et al. [12] first proposed a polynomial-time approximation scheme (PTAS) for the minimum dominating set problem in unit disk graphs. Later Kuhn et al. [11] gave local approximation schemes for the maximum independent set problem and the minimum dominating set problem in growth-bounded graphs. Then Gafeller et al. [13], Gao et al. [14] and Liu et al. [15] improved the algorithm in [12] respectively. Gafeller et al. [13] gave a $(1+\varepsilon)$-approximation minimum connected dominating set. Gao et al. [14] provided a $(1+\bar{\varepsilon})$-approximation minimum $d$-hop connected dominating set, where $\bar{\varepsilon}$ is only dependent on $d$, $\varepsilon$ and $f$. Liu et al. [15] separately proposed PTASs for the minimum vertex cover, weighted vertex cover and connected vertex cover problems.

In this paper, we give a $(1+\varepsilon_1)$-approximation M$k$PCVC under the constraint of bounded degree for any $\varepsilon_1>0$ based on the algorithms in [11-15].

The rest of the paper is organized as follows. In Section 3, we give some preliminaries which will be needed later. In Section 4, we present the algorithm and give the proof of its correctness, time complexity and performance. Finally, the conclusions and future works are drawn in Section 5.

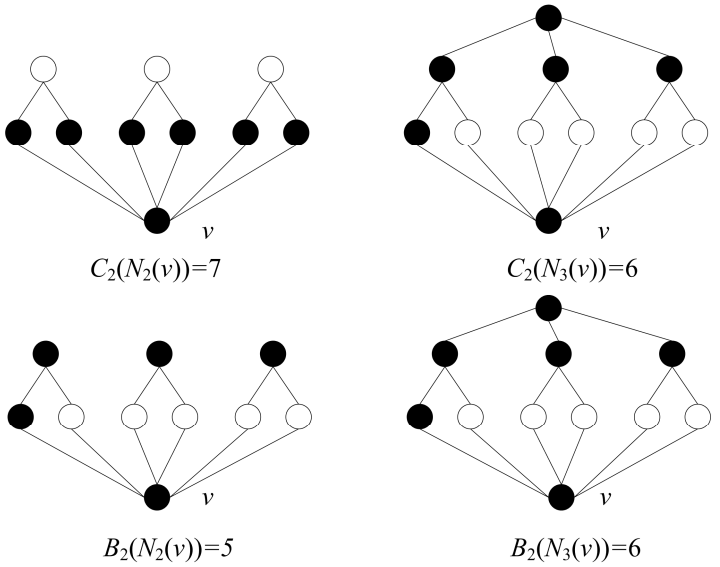
## 3 Preliminaries

In this Section, we first provide some definitions. Then, we give several related results.

For a given graph $G = (V, E)$, if the maximum vertex degree $\Delta$ in $G$ is upper bounded by a constant, then $G$ is under the constraint of bounded degree [16]. Let $S$ be the subset of $V$, then the subgraph of $G$ induced by $S$, denoted by $G[S]$, is the graph with vertices in $S$ and edges with both ends in $S$ [15].

**Definition 1.** [12] *For any vertex $v \in V(G)$, we define $N(v)$ as the set of the vertex $v$ and its adjacent neighbors, and $N_r(v)$ as the set of $v$ and its r-hop neighbors* (*nodes that reach $v$ via at most r-1 intermediate nodes*). *For any set $S \subseteq V(G)$, let $N_r(S) = \{N_r(v) | v \in S\}$.*

**Definition 2.** [6] *For any set $C \subseteq V(G)$, we call $C$ a k-path vertex cover of $G$ if each k-path in $G$ contains at least one vertex in $C$. Moreover, if the graph $G[C]$ is connected, then we call $C$ a k-path connected vertex cover. For any set $S \subseteq V(G)$, MkPCVC of $G[S]$* (*see Section I*) *is denoted by $C_k(S)$.*




$C_2(N_2(v))=7$ $C_2(N_3(v))=6$

$B_2(N_2(v))=5$ $B_2(N_3(v))=6$

**Fig. 1.** An example to illustrate the difference of *C* and *B*

Liu et al. [15] used $\left|C_2\left(N_{r_i+1}(v_i)\right) \setminus N_{r_i}(v_i)\right| \le \left|C_2\left(N_{r_i+1}(v_i)\right)\right| - \left|C_2\left(N_{r_i}(v_i)\right)\right|$ to prove the performance of his algorithm. In fact, his proof is not rigorous, especially for the *k*-path connected vertex cover problem. For example, $\left|C_2\left(N_3(v)\right) \setminus N_2(v)\right| > \left|C_2\left(N_3(v)\right)\right| - \left|C_2\left(N_2(v)\right)\right|$ as shown in Figure 1. Therefore, we first present the concept of *k*-path bounded connected vertex cover in Definition 3 to solve this problem.

**Definition 3.** *For any set $B \subseteq N_r(v)$, we call $B$ a k-path bounded connected vertex cover of $G[N_r(v)]$ if $B$ is a k-path vertex cover of $G[N_r(v)]$ and each connected component in $G[B]$ contains at least one vertex in $N_r(v) \setminus N_{r-1}(v)$. For any set $S \subseteq V(G)$, the minimum k-path bounded connected vertex cover* (*MkPBCVC*) *of $G[S]$ is denoted by $B_k(S)$.*

Since $B_k\left(N_{r_i+1}(v_i)\right)\bigcap N_{r_i}(v_i)$ is one $k$-path bounded connected vertex cover of $G\left[N_{r_i}(v_i)\right]$, $B_k\left(N_{r_i}(v_i)\right)\le B_k\left(N_{r_i+1}(v_i)\right)\bigcap N_{r_i}(v_i)$. Then we have

$$\left|B_k\left(N_{r_i+1}(v_i)\right)\setminus N_{r_i}(v_i)\right|\le\left|B_k\left(N_{r_i+1}(v_i)\right)\right|-\left|B_k\left(N_{r_i}(v_i)\right)\right|. \quad (1)$$

**Definition 4.** *For any set $I\subseteq V(G)$, we call $I$ an independent set if every two vertices in $I$ are not adjacent to each other. Moreover, we call $I$ a maximal independent set (MIS) if $I\cup\{u\}$ is no longer an independent set, for any vertex $u\in V(G)\setminus I$. For any set $S\subseteq V(G)$, a MIS of $G[S]$ is denoted by $I(S)$.*

Clearly, the subset $C\subseteq V(G)$ is a vertex cover of $G$ if and only if $V(G)\setminus C$ is an independent set. Hence we can get a vertex cover after we find a MIS of $G$. Similarly, we can also get a $k$-path vertex cover after we find a $k$-hop MIS ($k$-MIS) which is defined in Definition 5.

**Definition 5.** [14] *For any set $I_k\subseteq V(G)$, we call $I_k$ a $k$-hop independent set if there is no $k$-path in $G[I_k]$. Moreover, we call $I_k$ a $k$-MIS if $I_k\cup\{u\}$ is no longer a $k$-hop independent set for any vertex $u\in V(G)\setminus I_k$. For any set $S\subseteq V(G)$, a $k$-MIS of $G[S]$ is denoted by $I_k(S)$.*

**Definition 6.** [12] *For any two vertices $u$ and $v$ in $G$, $dist(u, v)$ denotes the length of the shortest path from $u$ to $v$ in $G$. For any two subsets $S_1$ and $S_2$ of $V(G)$, $dist(S_1, S_2) = min\{dist(u, v)|u\in S_1, v\in S_2\}$. If $S_1\cap S_2\neq\phi$, then $dist(S_1, S_2) = 0$.*

Obviously, for any two subgraphs $G[S_1]$ and $G[S_2]$ of $G$, if $dist(S_1, S_2) = k$, we can add $k$ -1 vertices to connect $G[S_1]$ with $G[S_2]$.

**Definition 7.** [11] *We call a graph $G$ a growth-bounded graph if there exists a polynomial function $f(r)$ such that for every $v\in V(G)$ and $r\ge 1$, the size of the largest independent set in $N_r(v)$ is at most $f(r)$.*

Growth-bounded graphs generalize different classes of graphs including unit disk graphs, unit ball graphs and coverage area graphs. In the remainder of this section, we give two lemmas which are used in the following sections in our analysis. Lemma 1 can be derived from [15], which is used in the proof of Lemma 5. Lemma 2 , used in both Lemma 8 and Lemma 10, was applied in [2].

**Lemma 1.** [15] *Given a growth-bounded graph $G$ with polynomial function $f$ and maximum vertex degree $\Delta$, for any vertex $v\in V(G)$ and $r\ge 1$, it holds that* $\left|B_k\left(N_r(v)\right)\right|\le(1+\Delta)\cdot f(r)$.

**Lemma 2.** [2] *Given a k-path vertex cover of G denoted by C(V), if G[C(V)] is not connected, then there exists two connected components G[$S_1$] and G[$S_2$] in G[C(V)] such that dist($S_1$, $S_2$) ≤ k.*

## 4 A PTAS for M*k*PCVC Problem

Our work is based on [11-15], which have given approximate algorithms for optimization problems in growth-bounded graphs. The main idea can be described as follows: First, the growth-bounded graph *G* is divided into several disjoint clusters with different radius. Then, they compute the optimal solution set of each cluster and merge all the sets into one, which is their final solution. We use the same partition scheme to compute M*k*PCVC in the growth-bounded graph *G*. The difference is that we use a new criterion to determine the radius of each cluster so that the problem of (1) can be solved. In addition, we let the clusters overlap more to ensure the connectivity of our solution.

**Algorithm 1** PTAS for MkPCVC Problem
**Input:** a growth-bounded graph *G* with bounded degree constraint, initial parameters *k* and $\varepsilon > 0$
**Output:** a *k*-path connected vertex cover of *G*

1 $C \leftarrow \Phi$, $V_C \leftarrow V$;
2 { For analysis: $i = 1$; }
3 **while** $V_C$ is not empty **do**
4 choose a vertex $v$ from $V_C$;
5 $r \leftarrow 0$;
6 find the smallest radius $r$ such that $\left|B_k\left(N_{r+2k}(v)\right)\right| \le (1+\varepsilon)\cdot\left|B_k\left(N_r(v)\right)\right|$;
7 $C = C \bigcup C_k\left(N_{r+2k}(v)\right)$;
8 $V_c = V_c \setminus N_r(v)$;
9 { For analysis: $v_i = v$, $r_i = r$, $S_i = N_r(v)$, $T_i = N_{r+2k}(v)$, $i = i + 1$; }
10 **end while**

We now analyze the performance of Algorithm 1. First, we prove that the radius $r_i$ has an upper bound to show that Algorithm 1 can be done in polynomial time by Lemma 3. Then, Lemma 4 and Lemma 5 prove that the final solution *C* is a *k*-path connected vertex cover of *G*. Theorem 1 provides the runtime of our algorithm. Next, we analyze its approximation performance. For each cluster $S_i$, Lemma 6 gives the relation between $|C_k(S_i)|$ and $|B_k(S_i)|$. Lemma 7 and Lemma 8 provide the relation between $|B_k(S_i)|$ and $|C^* \cap S_i|$ in two different conditions, and it is not considered in [13-15] in condition that $S_i$ meets the bounds of the clusters which have been eliminated before (see in Figure 3). Then Lemma 9 gets the relation between $\sum |B_k(S_i)|$ and $|C^*|$ based on Lemma 7 and Lemma 8. In the end, Theorem 2 computes the relation

between $|C|$ and $|C^*|$ (i.e., the performance of Algorithm 1) based on Lemma 6 and Lemma 9.

Liu et al. [15] has proved the radius $r \leq r(\varepsilon)$ for vertex cover, weighted vertex cover and connected vertex cover problems. Next we use a similar strategy to prove the correctness of Lemma 3.

**Lemma 3.** *For given growth-bounded graph G with bounded degree constraint* $\Delta$*, the radius* $r_i$ *has an upper bound* $r(f, \varepsilon)$ *such that* $\left|B_k\left(N_{r+2k}(v_i)\right)\right| \leq (1+\varepsilon)\cdot\left|B_k\left(N_r(v_i)\right)\right|$*, where f is a polynomial function of G.*

*Proof.* Assume that there is no upper bound of $r_i$, i.e., $\left|B_k\left(N_{r_i+2k}(v_i)\right)\right| > (1+\varepsilon)\left|B_k\left(N_{r_i}(v_i)\right)\right|$ for any positive integer $r_i \geq 1$.

If $r_i = 2k\cdot t$ where $t$ is a positive integer, then

$$\left|B_k\left(N_{r_i+2k}(v_i)\right)\right| > (1+\varepsilon)\left|B_k\left(N_{r_i}(v_i)\right)\right| > \ldots > (1+\varepsilon)^{\frac{r_i}{2k}+1}\left|B_k\left(N_0(v_i)\right)\right| \geq (1+\varepsilon)^{t+1};$$

if $r_i = 2k\cdot t+1$, then

$$\left|B_k\left(N_{r_i+2k}(v_i)\right)\right| > (1+\varepsilon)\left|B_k\left(N_{r_i}(v_i)\right)\right| > \ldots > (1+\varepsilon)^{\frac{r_i+2k-1}{2k}}\left|B_k\left(N_1(v_i)\right)\right| \geq (1+\varepsilon)^{t+1};$$

if $r_i = 2k\cdot t+2$, then

$$\left|B_k\left(N_{r_i+2k}(v_i)\right)\right| > (1+\varepsilon)\left|B_k\left(N_{r_i}(v_i)\right)\right| > \ldots > (1+\varepsilon)^{\frac{r_i+2k-2}{2k}}\left|B_k\left(N_2(v_i)\right)\right| \geq (1+\varepsilon)^{t+1};$$

...

if $r_i = 2k\cdot t+2k-1$, then

$$\left|B_k\left(N_{r_i+2k}(v_i)\right)\right| > (1+\varepsilon)\left|B_k\left(N_{r_i}(v_i)\right)\right| > \ldots > (1+\varepsilon)^{\frac{r_i+1}{2k}}\left|B_k\left(N_{2k-1}(v_i)\right)\right| \geq (1+\varepsilon)^{t+1}.$$

Since $\left|B_k\left(N_{r_i+2k}(v_i)\right)\right| \leq (1+\Delta)\cdot f(r_i+2k)$ by Lemma 1, we have

$$(1+\Delta)\cdot f(r_i+2k) \geq \left|B_k\left(N_{r_i+2k}(v_i)\right)\right| \geq (1+\varepsilon)^{t+1} \geq (1+\varepsilon)^{\frac{r_i}{2k}} \tag{2}$$

for any positive integer $r_i \geq 1$. In fact, (2) has to be violated when $r_i$ is large enough, contradicting to our assumption that $\left|B_k\left(N_{r_i+2k}(v_i)\right)\right| > (1+\varepsilon)\left|B_k\left(N_{r_i}(v_i)\right)\right|$ for any positive integer $r_i \geq 1$. □

Assume that the integer $m$ is the highest order of polynomial function $f(r)$, then $f(r+2k) \leq A_1\cdot r^m$, where $A_1$ is a constant. Thus we have $(1+\varepsilon)^{\frac{r}{2k}} \leq (1+\Delta)\cdot A_1\cdot r^m$ for (2). Let $C_2 = (1+\Delta)\cdot A_1$, then we have

$$(1+\varepsilon)^{\frac{r}{2k}} \leq A_2\cdot r^m \quad . \tag{3}$$

The smaller $\varepsilon$ is, the larger the upper bound of $r$ is. Hence, we can consider the upper bound $r(\varepsilon)$ when $0 < \varepsilon < \frac{1}{4k(A_2+3m)} < \frac{1}{2}$. If $\varepsilon \geq \frac{1}{4k(A_2+3m)}$, then $r \leq r(\varepsilon)$.

Nieberg et al. [17] has provided the upper bound of $r$ when $m = 2$. We can aloso use the same method to prove that the upper bound of $r$ is $O((1/\varepsilon)\cdot ln(1/\varepsilon))$ when $m \geq 0$.

**Lemma 4.** *For any two vertices $v_i$ and $v_j$ chosen in Algorithm 1, if $T_i \bigcap S_j \neq \phi$, then $G[C_k(T_i)]$ is connected with $G[C_k(T_j)]$.*

*Proof.* Assume that the vertex $u_1$ is contained in $\left(T_i \setminus N_{r_i+2k-1}(v_i)\right) \bigcap S_j$. Then, there exists one $k$-path $e_1$ in $T_i \setminus N_{r_i+k}(v_i)$ such that $u_1$ is one end of $e_1$. Since the $k$-path $e_1$ is covered by $C_k(T_i)$, there is one vertex $u_2$ in $e_1$ such that $u_2$ is contained in $C_k(T_i)$. Then, we have one $k$-path $e_2$ in $C_k(T_i)$ and $u_2$ is one end of $e_2$. Obviously, $e_2$ is also a $k$-path in $T_j$, so $e_2$ is covered by $C_k(T_j)$. Thus, there is one vertex $u_3$ in $e_2$ such that $u_3 \in C_k(T_i) \bigcap C_k(T_j)$. Hence, $C_k(T_i) \bigcap C_k(T_j) \neq \phi$, i.e., $G[C_k(T_i)]$ is connected with $G[C_k(T_j)]$. □

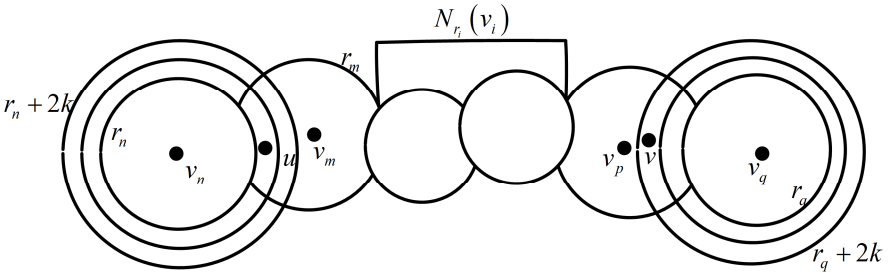



**Fig. 2.** This is an example to illustrate the clusters chosen in Algorithm 1

**Lemma 5.** *The final solution C is a k-path connected vertex cover of G.*

*Proof.* We first prove that $C$ is a $k$-path vertex cover of $G$. Assume that $C$ is not a $k$-path vertex cover of $G$. Then, there exists at least one $k$-path $e_u \subseteq G$ such that $e_u$ is not covered by $C$. Let $e_u = (u_1, u_2, u_3, \ldots, u_k)$, then $C$ does not contain any vertex in $\{u_1, u_2, u_3, \ldots, u_k\}$. The set $V_C$ is empty upon completion of Algorithm 1. So, there exists a vertex $v_i$ and a $j$ $(1 \leq j \leq k)$ such that $u_j \in S_i$ and $\{u_1, u_2, u_3, \ldots, u_k\} \setminus u_j$ are contained in $T_i$. As a result, $e_u$ must be contained in $G[(T_i)]$ and $e_u$ is covered by $C_k(T_i) \subseteq C$, contradicting to our assumption.

Next we prove that the induced graph $G[C]$ is connected, i.e., $u$ and $v$ lie in the same connected component for any two vertices $u$ and $v$ in $G[C]$.

Assume that $u \in S_m \bigcap C_k(T_n)$ and $v \in S_p \bigcap C_k(T_q)$. As shown in Figure 2, there must be some sets $S_i$ so that $G[S_n]$ is connected with $G[S_q]$ through $G[S_m], G\left[\bigcup S_i\right]$ and $G[S_q]$. $G[C_k(T_n)]$ is connected with $G[C_k(T_q)]$ through $G[C_k(T_m)], G\left[\bigcup C_k(T_i)\right]$

and $G[C_k(T_q)]$ can be easily derived from Lemma 4. Hence, $u$ and $v$ are connected in $G[C]$. (Here, $S_m$ and $S_n$ can be the same cluster, so can $S_p$ and $S_q$.) □

**Theorem 1.** *The runtime of Algorithm 1 is* $r \cdot n^{O(f(r))}$, *where* $r = O((1/\varepsilon)\cdot\ln(1/\varepsilon))$, $n = |V|$.

*Proof.* The subset $C \subseteq V$ is a $k$-path vertex cover of $G$ if and only if $V \setminus C$ is a $k$-hop independent set. If we select a $k$-hop independent set $I_k$ of $G[T_i]$, then $T_i \setminus I_k$ should be a $k$-path vertex cover of $G[T_i]$. By Definition 7, the size of the maximal independent set of $G[T_i]$, denoted by |MIS|, is no larger than $f(r_i)$. Since $|\text{MIS}_k| \le |\text{MIS}| \le f(r_i)$, we can find all the $k$-path vertex covers of $G[T_i]$ by exhausting search within polynomial time $n_i^{O(f(r_i))}$ where $n_i = |T_i|$, and then we can pick out $C_k(T_i)$ and $B_k(T_i)$. By Lemma 3, we know that the radius $r_i$ has an upper bound $r(\varepsilon)$ where $r(\varepsilon) = O((1/\varepsilon)\cdot\ln(1/\varepsilon))$. Thus, the time complexity of Algorithm 1 is $\sum_i r_i \cdot n_i^{O(f(r_i))} \le r \cdot n^{O(f(r))}$. □

**Lemma 6.** $\left|C_k(T_i)\right| \le (1+\varepsilon_2)\left|B_k(S_i)\right|$ *where* $1+\varepsilon_2 = 1+k\cdot\varepsilon$.

*Proof.* By Lemma 2, for any $k$-path vertex cover, if we add $k$-1 certain vertices, the number of connected components will be reduced by one. Let $d_i$ be the number of connected components in $B_k(T_i)$, then we can add $(d_i\text{-}1)\cdot(k\text{-}1)$ certain vertices to make $G[B_k(T_i)]$ connected. Thus we have

$$\left|C_k(T_i)\right| \le \left|B_k(T_i)\right| + (d_i - 1)(k-1) \quad . \tag{4}$$

According to Definition 3, we have

$$d_i \le \left|B_k(T_i) \bigcap \left(T_i \setminus N_{r_i+2k-1}(v_i)\right)\right| \le \left|B_k(T_i) \bigcap (T_i \setminus S_i)\right| \quad . \tag{5}$$

Since $B_k(T_i) \cap S_i$ is a $k$-path bounded connected vertex cover of $G[S_i]$ and $B_k(S_i)$ is the one with minimum cardinality, $\left|B_k(S_i)\right| \le \left|B_k(T_i) \cap S_i\right|$.

As $\left|B_k(T_i) \cap (T_i \setminus S_i)\right| = \left|B_k(T_i)\right| - \left|B_k(T_i) \cap S_i\right|$, we have

$$\left|B_k(T_i) \cap (T_i \setminus S_i)\right| \le \left|B_k(T_i)\right| - \left|B_k(S_i)\right| \quad . \tag{6}$$

By (5), (6) and (7), we have

$$\left|C_k(T_i)\right| \le \left|B_k(T_i)\right| + \left(\left|B_k(T_i)\right| - \left|B_k(S_i)\right|\right)\cdot(k-1) \quad . \tag{8}$$

Furthermore, we have

$$\left|B_k(T_i)\right| \le (1+\varepsilon)\left|B_k(S_i)\right| \quad . \tag{9}$$

By (8) and (9), we have $\left|C_k(T_i)\right| \le (1+k\cdot\varepsilon)\left|B_k(S_i)\right|$. □

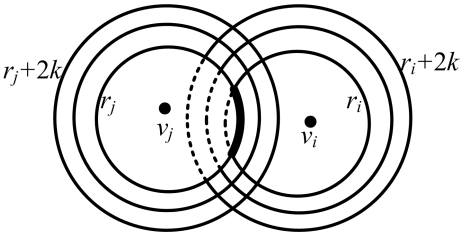



**Fig. 3.** This is an example to illustrate the clusters $S_i$ and $S_j$ chosen in Algorithm 1

**Lemma 7.** *For given graph G, if* $N_{r_i-1}(v_i)$ *and* $G \setminus S_i$ *are connected completely through the vertices in* $S_i \setminus N_{r_i-1}(v_i)$*, i.e., $S_i$ doesn't meet the bound of other partitions during its construction in Algorithm 1, then* $\left|B_k(S_i)\right| \le \left|C^* \cap S_i\right|$*, where $C^*$ is the optimum k-path connected vertex cover of G.*

*Proof.* Clearly, $C^* \cap S_i$ is a *k*-path vertex cover of $S_i$. Since $G[C^*]$ is a connected graph, both $C^* \cap N_{r_i-1}(v_i)$ and $C^* \cap (G \setminus S_i)$ are connected in $G[C^*]$. And both $C^* \cap N_{r_i-1}(v_i)$ and $C^* \cap (G \setminus S_i)$ are connected through the vertices in $C^* \cap \left(S_i \setminus N_{r_i-1}(v_i)\right)$, i.e., each connected component in $G[C^* \cap S_i]$ contains at least one vertex in $S_i \setminus N_{r_i-1}(v_i)$, because $N_{r_i-1}(v_i)$ and $G \setminus S_i$ are connected through the vertices in $S_i \setminus N_{r_i-1}(v_i)$. By Definition 3, $C^* \cap S_i$ is a *k*-path bounded connected vertex cover of $G[S_i]$. Since $B_k(S_i)$ is the one with minimum cardinality, $\left|B_k(S_i)\right| \le \left|C^* \cap S_i\right|$. □

In step 8 in Algorithm 1, every time one partition will be eliminated from *G*, such as $S_j$. Thus, the *r-hop* neighbors of $v_i$ in *G*, $N'_{r_i}(v_i)$, may be overlapped with $S_j$ which has been eliminated before. As a result, in the partition computed in Algorithm 1, $S_i$ may be different from $N'_{r_i}(v_i)$. As shown in Figure 3, $N_{r_i-1}(v_i)$ and $G \setminus S_i$ are connected through the vertices in $S_i \setminus N_{r_i-1}(v_i)$ and $N_{r_j+1}(v_j) \setminus S_j$ (see the heavier line in Figure 3). Then we have Lemma 8.

**Lemma 8.** *If* $N_{r_i-1}(v_i)$ *and* $G \setminus S_i$ *are connected through the vertices in* $S_i \setminus N_{r_i-1}(v_i)$ *and* $N_{r_j+1}(v_j) \setminus S_j$ *(see the heavier line in Figure 3, denoted by $N_j$), i.e., $S_i$ meets the bound of $S_j$ during its construction in Algorithm 1, then*
$\left|B_k(S_i)\right| \le \left|C^* \cap S_i\right| + (k-1)^2 \left|B_k(T_j) \cap S_i\right|$.

*Proof.* By Lemma 7, $C^* \cap S_i$ is a $k$-path vertex cover of $S_i$. Since $N_{r_i-1}(v_i)$ and $G \setminus S_i$ are connected through the vertices in $N_{r_i}(v_i) \setminus N_{r_i-1}(v_i)$ and $N_j$, each connected component in $C^* \cap S_i$ contains at least one vertex in $N_{r_i}(v_i) \setminus N_{r_i-1}(v_i)$ or $N_j$. Next, we need to modify $C^* \cap S_i$ to compare $|B_k(S_i)|$ with $|C^* \cap S_i|$.

After Algorithm 2, $C'$ has $t+|Y|$ connected components connected to $N_j$, and the other connected components are connected to $N_{r_i}(v_i) \setminus N_{r_i-1}(v_i)$. Moreover, we have $t+|Y| \le t+|Z| \le (k-1)|B_k(T_j) \cap S_i|$. Similar to the proof of (5), we can add at most $(k-1)^2 |B_k(T_j) \cap S_i|$ vertices to make $C'$ to be a $k$-path bounded connected vertex cover of $S_i$. Therefore, we have

$$|B_k(S_i)| \le |C'| + (k-1)^2 |B_k(T_j) \cap S_i| \le |C^* \cap S_i| + (k-1)^2 |B_k(T_j) \cap S_i|. \quad \square$$

---

**Algorithm 2** Scheme to modify $C^* \cap S_i$

**Input:** $C^* \cap S_i$, a vertex $v$ not contained in $C^* \cap S_i$

**Output:** $C'$(a $k$-path vertex cover of $S_i$ with $t+|Y|$ connected components connected to $N_j$, and $|C'| \le |C^* \cap S_i|$)

---

$C' \leftarrow C^* \cap S_i$, $Y \leftarrow \{v\}$, $Z \leftarrow \Phi$;
**while** $|Z| < |Y|$ **do**
  $C' = C' \setminus Y$, $C' = C' \cup Z$;
  $t \leftarrow 0$, $Y \leftarrow \Phi$, $Z \leftarrow \Phi$;
  **for** those components which contain vertices in $N_j$
    **if** the size of the component is larger than or equal to $k$
      **then** $t \leftarrow t+1$ and there exists a vertex in this component which is contained in $B_k(T_j) \cap S_i$;
    **else**
      **then** let the vertex in this component with minimum degree and not contained in $N_j$ be $y$, and put $y$ into the set $Y$(if there is only one vertex contained in $N_j$ in this component, then put this vertex into the set $Y$); for those $k$-paths only covered by $y$ in $G[S_i]$, there exists one vertex, denoted by $z$, contained in $B_k(T_j) \cap S_i$ in each $k$-path. By Lemma 3.2, if we add $k$-2 certain vertices, the vertex $z$ can be connected to another connected component in $C^* \cap S_i$. Put all these vertices into the set $Z$;
  **end for**
**end while**

---

Lemma 8 proves the conclusion under the condition that $S_i$ meets the bound of only one partition $S_j$ during its construction in Algorithm 1. If $S_i$ meets more than one partition, then we have Lemma 9 and the proof is similar.

**Lemma 9.** *Let* $S_1' = \bigcup_{i=1}^{j} S_i$ , $T_1' = \bigcup_{i=1}^{j} T_i$ , $B_k(T_1') = \bigcup_{i=1}^{j} B_k(T_i)$ , $S_2' = \bigcup_{i=j+1}^{q} S_i$ , $B_k(S_2') = \bigcup_{i=j+1}^{q} B_k(S_i)$. *If* $S_{j+1}$, …, $S_q$ *are the partitions which only meet the bound of* $S_1'$ *during its construction, then* $\left|B_k(S_2')\right| \le \left|C^* \cap S_2'\right| + (k-1)^2 \left|B_k(T_1') \cap S_2'\right|$.

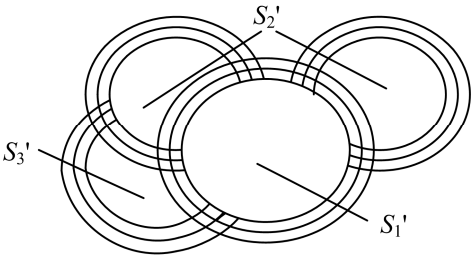



**Fig. 4.** This is an example to illustrate $S_1$', $S_2$' and $S_3$'

**Lemma 10.** $\sum_i \left|B_k(S_i)\right| \le (1+\varepsilon_3)\left|C^*\right|$ *holds in Algorithm 1, where* $1+\varepsilon_3 = \dfrac{1}{1-(k-1)^2\varepsilon}$ *and* $C^*$ *is the optimum k-path connected vertex cover of G.*

*Proof.* As shown in Figure 4, denote the partitions which don't meet the bound of other partitions during its construction in Algorithm 1 by $S_1$, $S_2$, …, $S_j$ and let $S_1' = \bigcup_{i=1}^{j} S_i$ , $T_1' = \bigcup_{i=1}^{j} T_i$ , $B_k(S_1') = \bigcup_{i=1}^{j} B_k(S_i)$, and $B_k(T_1') = \bigcup_{i=1}^{j} B_k(T_i)$. Denote the partitions which only meet the bound of $S_1'$ during its construction by $S_{j+1}$, …, $S_q$ and let $S_2' = \bigcup_{i=j+1}^{q} S_i$ , $T_2' = \bigcup_{i=j+1}^{q} T_i$ , $B_k(S_2') = \bigcup_{i=j+1}^{q} B_k(S_i)$ and $B_k(T_2') = \bigcup_{i=j+1}^{q} B_k(T_i)$. Denote the partitions which only meet the bound of $S_1'$ and $S_2'$ during its construction by $S_{q+1}$, …, $S_m$ and let $S_3' = \bigcup_{i=q+1}^{m} S_i$ , $T_3' = \bigcup_{i=q+1}^{m} T_i$ ,

$B_k(S_3') = \bigcup_{i=q+1}^{m} B_k(S_i)$, $B_k(T_3') = \bigcup_{i=q+1}^{m} B_k(T_i)$; … By analogy, assume that the final groups are $S_n'$ and $T_n'$. By Lemma 7, we have $\left|B_k(S_1')\right| \le \left|C^* \cap S_1'\right|$. By Lemma 9, we have

$$\left|B_k(S_2')\right| \le \left|C^* \cap S_2'\right| + (k-1)^2 \left|B_k(T_1') \cap S_2'\right|;$$

$$\left|B_k(S_3')\right| \le \left|C^* \cap S_3'\right| + (k-1)^2 \left|\left(B_k(T_1') \cup B_k(T_2')\right) \cap S_3'\right|;$$

…

$$\left|B_k(S_i')\right| \le \left|C^* \cap S_i'\right| + (k-1)^2 \cdot \left|\left(B_k(T_1') \cup B_k(T_2') \cup \ldots \cup B_k(T_{i-1}')\right) \cap S_i'\right|;$$

…

$$\left|B_k(S_n')\right| \le \left|C^* \cap S_n'\right| + (k-1)^2 \cdot \left|\left(B_k(T_1') \cup B_k(T_2') \cup \ldots \cup B_k(T_{n-1}')\right) \cap S_n'\right|.$$

Thus,

$$\begin{aligned}
&\left|B_k(S_1')\right| + \left|B_k(S_2')\right| + \ldots + \left|B_k(S_n')\right| \le \left(\left|C^* \cap S_1'\right| + \ldots + \left|C^* \cap S_n'\right|\right) \\
&+ (k-1)^2 \left(\left|B_k(T_1') \cap S_2'\right| + \ldots + \left|\left(B_k(T_1') \cup \ldots \cup B_k(T_{n-1}')\right) \cap S_n'\right|\right) \\
&\le \left(\left|C^* \cap S_1'\right| + \ldots + \left|C^* \cap S_n'\right|\right) \\
&+ (k-1)^2 \left(\left|B_k(T_1') - B_k(S_1')\right| + \ldots + \left|B_k(T_{n-1}') - B_k(S_{n-1}')\right|\right) \\
&\le \left(\left|C^* \cap S_1'\right| + \ldots + \left|C^* \cap S_n'\right|\right) + (k-1)^2 \varepsilon \left(\left|B_k(S_1')\right| + \ldots + \left|B_k(S_n')\right|\right).
\end{aligned}$$

Hence, $\sum_i \left|B_k(S_i)\right| \le \sum_i \left|B_k(S_i')\right| \le \frac{1}{1-(k-1)^2\varepsilon} \cdot \sum_{i=1}^{n} \left|C^* \cap S_i'\right| \le (1+\varepsilon_3)\left|C^*\right|$. □

**Theorem 2.** *For given graph $G$, $|C| \le (1+\varepsilon_1)\cdot|C^*|$ holds in Algorithm 1, where $1+\varepsilon_1 = (1+\varepsilon_2)\cdot(1+\varepsilon_3)$ and $C^*$ is the optimum $k$-path connected vertex cover of $G$.*

*Proof.* $\sum_i \left|C_k(T_i)\right| \le (1+\varepsilon_2) \cdot \sum_i \left|B_k(S_i)\right|$ by Lemma 6, and $\sum_i \left|B_k(S_i)\right| \le (1+\varepsilon_3) \cdot \left|C^*\right|$ by Lemma 10. Therefore, we have $\sum_i \left|C_k(T_i)\right| \le (1+\varepsilon_2)\cdot(1+\varepsilon_3)\cdot\left|C^*\right|$. □

## 5 Conclusion

In the paper, we present a PTAS for M$k$PCVC problem, which can be applied in designing security protocols for WSNs [6], in growth-bounded graphs under the constraint of bounded degree. In contrast to [2], our algorithm only relies on the adjacent relation between vertices and doesn't need the geometric representation of the vertices. Also, our algorithm can be easily extended to a distributed one based on a greedy method. However, there may only be one active cluster in the graph in each

round, which leads to a linear time complexity [11]. Hence, improving our algorithm into a totally distributed one and applying it to WSNs will be our future work.

**Acknowledgments.** This work is supported by National Natural Science Foundation of China (No. 61170021), Application Foundation Research of Suzhou of China (No.SYG201240)**.**

## References

1. Brešar, B., Kardoš, F., Katrenič, J., Semanišin, G.: Minimum k-path vertex cover. Dis. Appl. Math. 159, 1189–1195 (2011)
2. Liu, X., Lu, H., Wang, W., Wu, W.: PTAS for the minimum *k*-path connected vertex cover problem in unit disk graphs. J. Global Opt. 56, 449–458 (2013)
3. Gollmann, D.: Protocol analysis for concrete environments. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 365–372. Springer, Heidelberg (2005)
4. Menezes, A.J., Van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (2010)
5. Novotný, M.: Formal analysis of security protocols for wireless sensor networks. Tatra Mt. Math. Publ. 47, 81–97 (2010)
6. Novotný, M.: Design and analysis of a generalized canvas protocol. In: Samarati, P., Tunstall, M., Posegga, J., Markantonakis, K., Sauveron, D. (eds.) WISTP 2010. LNCS, vol. 6033, pp. 106–121. Springer, Heidelberg (2010)
7. Vogt, H.: Integrity preservation for communication in sensor networks. Tech. Rep. 434, ETH Zurich, Institute for Pervasive Computing (2004)
8. Vogt, H.: Exploring message authentication in sensor networks. In: 1st European Workshop Security Ad-Hoc Sensor Networks (2004)
9. Vogt, H.: Increasing Attack Resiliency of Wireless Ad Hoc and Sensor Networks. In: 2nd International Workshop on Security in Distributed Computing Systems, pp. 179-184 (2005)
10. Tu, J., Zhou, W.: A factor 2 approximation algorithm for the vertex cover P3 problem. Info Proc. Lett. 111, 683–686 (2011)
11. Kuhn, F., Moscibroda, T., Nieberg, T., Wattenhofer, R.: Local approximation schemes for ad hoc and sensor networks. In: DIALM-POMC Cologne, Germany, pp. 97-103 (2005)
12. Nieberg, T., Hurink, J.: A PTAS for the minimum dominating set problem in unit disk graphs. In: Approximation and Online Algorithms, pp. 296-306 (2006)
13. Gfeller, B., Vicari, E.: A Faster Distributed Approximation Scheme for the connected Dominating Set Problem for Growth-Bounded Graphs. In: 6th International Conference on Ad-Hoc, Mobile, and Wireless Networks, pp. 59-73 (2007)
14. Gao, X., Wang, W., Zhang, Z., Zhu, S., Wu, W.: A PTAS for minimum *d*-hop connected dominating set in growth-bounded graphs. Opt Lett 4, 321–333 (2010)
15. Liu, Y., Fan, J., Wang, D., Du, H., Zhang, S., Lv, J.: Approximation Algorithms for Vertex Cover Problems in WSN Topology Design. Ad Hoc and Sensor Wireless Networks (accepted)
16. Wang, Z., Wang, W., Kim, J.M., Thuraisingham, B., Wu, W.: PTAS for the minimum weighted dominating set in growth bounded graphs. J. Global Opt. 54, 641–648 (2012)
17. Nieberg, T., Hurink, J.L., Kern, W.: A robust ptas for maximum weight independent sets in unit disk graphs. In: Hromkovič, J., Nagl, M., Westfechtel, B. (eds.) WG 2004. LNCS, vol. 3353, pp. 214–221. Springer, Heidelberg (2004)

Journal of Discrete Algorithms 26 (2014) 1–6

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

# On the metric dimension of HDN

Dacheng Xu, Jianxi Fan *

*School of Computer Science and Technology, Soochow University, Suzhou 215006, China*

ARTICLE INFO

*Article history:*
Received 23 March 2012
Received in revised form 17 September 2013
Accepted 18 September 2013
Available online 2 October 2013

*Keywords:*
Metric basis
Metric dimension
HDN1($n$)
HDN2($n$)

ABSTRACT

The concept of metric basis is useful for robot navigation. In graph $G$, a robot is aware of its current location by sending signals to obtain the distances between itself and the landmarks in $G$. Its position is determined uniquely in $G$ if it knows its distances to sufficiently many landmarks. The metric basis of $G$ is defined as the minimum set of landmarks such that all other vertices in $G$ can be uniquely determined and the metric dimension of $G$ is defined as the cardinality of the minimum set of landmarks. The major contribution of this paper is that we have partly solved the open problem proposed by Manuel et al. [9], by proving that the metric dimension of HDN1($n$) and HDN2($n$) are either 3 or 4. However, the problem of finding the exact metric dimension of HDN networks is still open.



## 1. Introduction

If we treat a robot moving from one position to another in Euclidean space as a point, then we can study the navigation in a graph [6,8]. In a graph $G$, a robot is aware of its current location by sending signals to obtain the distances between itself and the landmarks in $G$. Obviously, for any graph $G$, it is significant to find out the position and the minimum number of landmarks such that all other vertices in $G$ can be uniquely determined by their distances to the landmarks. In fact, the set composed of these landmarks is the metric basis of $G$, and the cardinality of this set is the metric dimension of $G$. The concept of metric basis is also used in chemistry [3,6] and combinatorial optimization [11].

Khuller et al. [8] proved that the metric dimension of path and $d$-dimensional grid is 1 and $d$, respectively. They also showed that the problem of deciding whether the metric dimension of $G$ is less than or equal to $k$ is NP-complete. Chartrand et al. [3] studied the metric dimension of complete graphs $K_n$, complete bigraphs $K_{m,n}$, trees, unicyclic graphs, and ordinary connected graphs. Fehr et al. [5] considered the metric dimension of Cayley digraphs. Javaid et al. [7] investigated the metric dimension of a family of circulant graphs $C_n(1,2)$, and Imran et al. [6] extended the research of [7]. They proved the metric dimension of $C_n(1,2,3)$ is 4 when $n \equiv 2,3,4,5 \pmod 6$ and gave an upper bound of the metric dimension of $C_n(1,2,3)$ when $n \equiv 0,1 \pmod 6$. Other related research results about the metric dimension appeared in [1,13,14].

Stojmenovic [12] proposed a family of variants of meshes and tori, which includes honeycomb meshes, honeycomb tori, and others. Compared with 2D-mesh and tori, honeycomb networks have better topological properties. The degree, diameter, and bisection width of honeycomb mesh are 3, $1.63\sqrt{N}$, and $0.82\sqrt{N}$, respectively, where $N$ is the number of vertices. The hexagonal mesh [4,10] is the dual of honeycomb mesh. Manual et al. [9] proved the metric dimension of honeycomb mesh is 3 by proving that the metric dimension of hexagonal mesh is 3. They also introduced two hex derived networks: HDN1($n$) and HDN2($n$), where $n \geqslant 2$. They have some superior performances over honeycomb meshes. The diameter of HDN1($n$) and HDN2($n$) are both $0.67\sqrt{N}$, less than honeycomb meshes. Manual et al. [9] proposed an open problem to show that the

* Corresponding author.
*E-mail addresses:* xudacheng06@163.com (D. Xu), jxfan@suda.edu.cn (J. Fan).

1570-8667/$ – see front matter 
http://dx.doi.org/10.1016/j.jda.2013.09.002

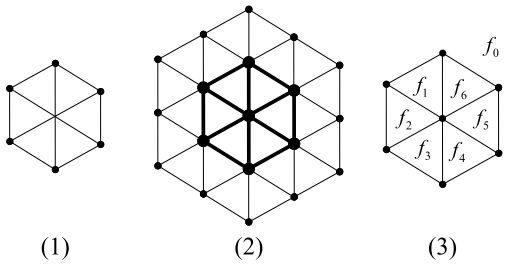



**Fig. 1.** Hexagonal meshes: (1) HX(2), (2) HX(3), and (3) all faces in HX(2).

metric dimension of HDN1($n$) and HDN2($n$) are between 3 and 5. In this paper, we have partly solved this open problem with a more tight bound that the metric dimension of HDN1($n$) and HDN2($n$) are either 3 or 4.

This work is organized as follows. Section 2 provides the definitions of metric basis, metric dimension, hexagonal network, HDN1($n$), and HDN2($n$). Detailed proofs that the metric dimension of HDN1($n$) and HDN2($n$) are either 3 or 4 are given in Section 3. We make a conclusion in Section 4.

## 2. Preliminaries

For an undirected graph $G = (V(G), E(G))$, $V(G)$ is the set of vertices and $E(G)$ is the set of edges. For two vertices $u$ and $v \in V(G)$, the distance between $u$ and $v$ is $d(u, v)$ which is defined as the length of a shortest $u - v$ path in $G$. Clearly, $d(u, v) = d(v, u)$. An $r$-neighborhood of $v$, denoted by $N_r(v)$, is a subset $U$ of $V(G)$, which means the distance between every vertex in $U$ and $v$ is $r$, i.e. $N_r(v) = \{u \in V(G) \mid d(u, v) = r\}$. Clearly, if $u \in N_{r1}(v) \cap N_{r2}(w)$, then $d(u, v) = r1$ and $d(u, w) = r2$. The degree of vertex $v$ is denoted by $\deg(v)$.

If $W \subset V(G)$ such that for any two vertices $u$ and $v \in V(G)$ there exists a vertex $w \in W$ such that $d(u, w) \neq d(v, w)$, then $W$ is a **locating set** of $G$. The vertices in $W$ are called locating landmarks. A locating set containing a minimum number of landmarks is a **minimum locating set** of $G$, which is called a **metric basis** of $G$. The cardinality of the metric basis is called the **metric dimension**, which is denoted by $\mathrm{md}(G)$.

Chen et al. [4] proposed the **hexagonal mesh**. A hexagonal mesh is made up with a set of triangles as shown in Fig. 1. 1-dimensional hexagonal mesh does not exist. A 2-dimensional hexagonal mesh HX(2) is composed of six triangles (see Fig. 1(1)). A 3-dimensional hexagonal mesh HX(3) is obtained from HX(2) by adding a layer of triangles around the boundary of HX(2) (see Fig. 1(2)). Similarly, HX($n$) is obtained from HX($n-1$) by adding a layer of triangles around the boundary of HX($n-1$).

A plane graph $G$ partitions the rest of the plane into a number of arcwise-connected open sets, which are called the **faces** of $G$ [2]. If two faces are adjacent, then they have at least one common edge. Every plane graph has one and only one unbounded face, called the **outer face**. Fig. 1(3) shows that HX(2) has seven faces $f_0, f_1, \ldots, f_6$ where $f_0$ is the outer face and $f_1$ is adjacent to $f_0$, $f_2$ and $f_6$.

In plane graph HX($n$), suppose any arbitrary face $f$ is bijective to one vertex $f^*$ except the outer face. If $f^*$ is located in the face $f$ and we connect the vertices of $f$ with $f^*$, then we get HDN1($n$). Fig. 2(1) demonstrates HDN1(3). Assume that $f$ is adjacent to $f_1, f_2, \ldots, f_k$ and $f_1^*, f_2^*, \ldots, f_k^*$ are bijective to $f_1, f_2, \ldots, f_k$, respectively. If we join the vertices of $f$ and $f_1^*, f_2^*, \ldots, f_k^*$ with $f^*$, then we get HDN2($n$). Clearly, HDN1($n$) is a subgraph of HDN2($n$). Fig. 2(2) shows HDN2(3). HDN1($n$) and HDN2($n$) are collectively known as HDN($n$), where $n \geqslant 2$.

In [9], a convenient coordinate system for HX($n$) was introduced. Actually, the coordinate system applies for HDN($n$) too (see Fig. 2), where $x$, $y$ and $z$ axes parallel to three edge directions of the hexagon. Lines that parallel to the coordinate axes $x$, $y$ and $z$ are $x$-lines, $y$-lines, and $z$-lines, respectively. All vertices on the $x$-lines have the same $x$-coordinate, all vertices on the $y$-lines have the same $y$-coordinate, and all vertices on the $z$-lines have the same $z$-coordinate. Thus, every vertex in HDN($n$) is assigned a single coordinate $(x, y, z)$. In Fig. 2, the coordinate of $A, B, C, D$, and $E$ are $(3, 6, 3)$, $(-3, -6, -3)$, $(-3, 3, 6)$, $(-6, -3, 3)$, and $(6, 3, -3)$, respectively.

For HDN($n$), $n \geqslant 2$, let $\gamma = (3(n-1), 3(n-1), 0)$, $\beta = (3(n-1), 0, -3(n-1))$, $\alpha = (0, -3(n-1), -3(n-1))$, $\eta = (-3(n-1), -3(n-1), 0)$, $\sigma = (-3(n-1), 0, 3(n-1))$, $\mu = (0, 3(n-1), 3(n-1))$, and $O = (0, 0, 0)$ (see Fig. 3).

For any other fundamental graph theoretical terminology, please refer to [2].

## 3. The metric dimension of HDN

In this section, we firstly prove $\mathrm{md}(\mathrm{HDN}(n)) \geqslant 3$. Then, we give a locating set $W$ of HDN such that $|W| = 4$. Finally, we come to a conclusion $3 \leqslant \mathrm{md}(\mathrm{HDN}(n)) \leqslant 4$.

Suppose that $W$ is a locating set of $G$. If $u \in V(G)$, $v \in W$, and $u \neq v$, then $d(u, v) > d(v, v)$. Thus, if we want to prove $W = \{w_1, w_2, \ldots, w_k\}$ is a locating set of $G$, we only need to prove that for any two vertices $u, v \in V(G) \backslash W$, there is a $w_i$ such that $d(u, w_i) \neq d(v, w_i)$, $1 \leqslant i \leqslant k$.

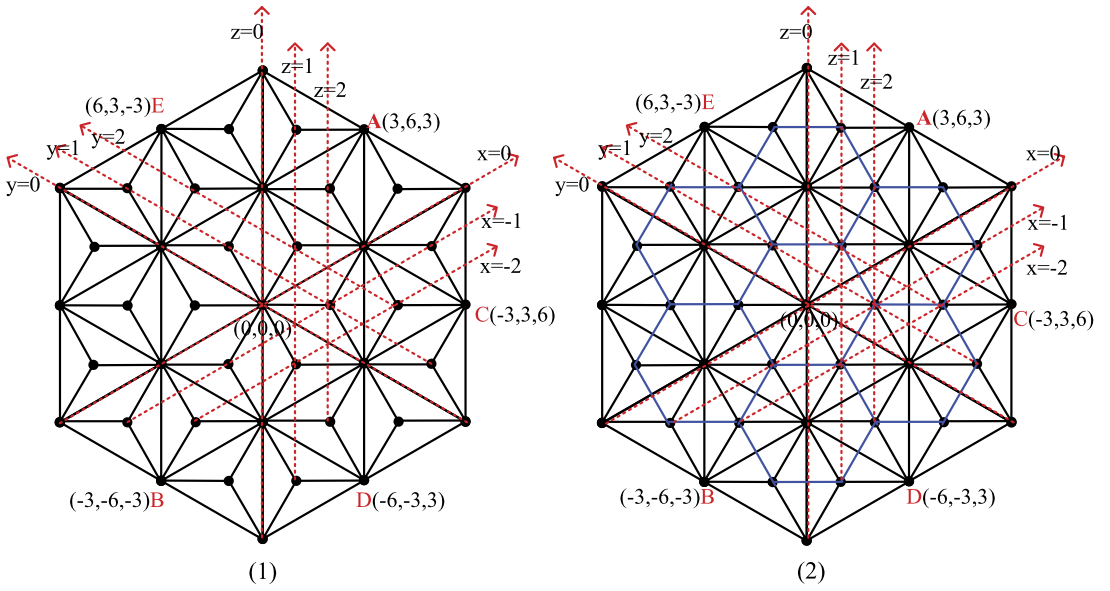



**Fig. 2.** HDN networks: (1) HDN1(3) and (2) HDN2(3).

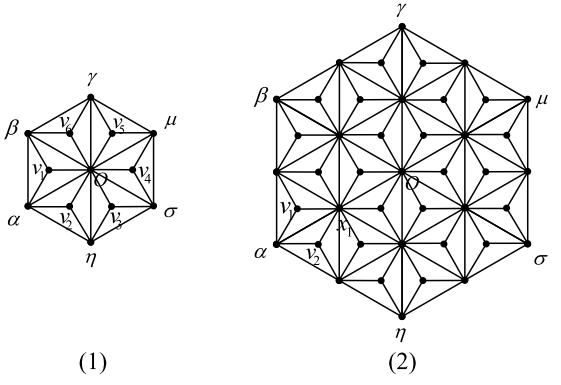



**Fig. 3.** HDN1($n$) networks: (1) HDN1(2) and (2) HDN1(3).

**Theorem 3.1.** *(See [8].) The metric dimension of a graph $G$ is 1 iff $G$ is a path.*

**Theorem 3.2.** *(See [8].) Let $G = (V, E)$ be a graph with metric dimension 2 and let $\{a, b\} \subset V$ be a metric basis in $G$. The following are true*:

1. *There is a unique shortest path $P$ between $a$ and $b$.*
2. *The degrees of $a$ and $b$ are at most 3.*
3. *Every other vertex on $P$ has degree at most 5.*

**Lemma 3.3.** md(HDN1(2)) $\geqslant 3$.

**Proof.** We prove this lemma by contradiction. Suppose md(HDN1(2)) $\leqslant 2$. However, according to Theorem 3.1, md(HDN1(2)) $\neq 1$. Thus, md(HDN1(2)) $= 2$. Fig. 3(1) demonstrates HDN1(2). Assume that $\{a, b\}$ is a metric basis of HDN1(2). According to Theorem 3.2, we have $a, b \in \{v_1, v_2, \ldots, v_6\}$, where $v_1 = (1, -1, -2)$, $v_2 = (-1, -2, -1)$, $v_3 = (-2, -1, 1)$, $v_4 = (-1, 1, 2)$, $v_5 = (1, 2, 1)$, and $v_6 = (2, 1, -1)$ (see Fig. 3(1)). Without loss of generality, let $a = v_1$. Depending on the distinct values of $b$, we have the following five cases.

Case 1. $b = v_2$. Since $d(\alpha, v_1) = d(O, v_1) = 1$ and $d(\alpha, v_2) = d(O, v_2) = 1$, we get a contradiction.
Case 2. $b = v_3$. Since $d(v_4, v_1) = d(v_6, v_1) = 2$ and $d(v_4, v_3) = d(v_6, v_3) = 2$, we get a contradiction.
Case 3. $b = v_4$. Since $d(v_2, v_1) = d(v_5, v_1) = 2$ and $d(v_2, v_4) = d(v_5, v_4) = 2$, we get a contradiction.
Case 4. $b = v_5$. Similar to Case 2.
Case 5. $b = v_6$. Similar to Case 1.
Thus, the assumption is not true and md(HDN1(2)) $\geqslant 3$. □

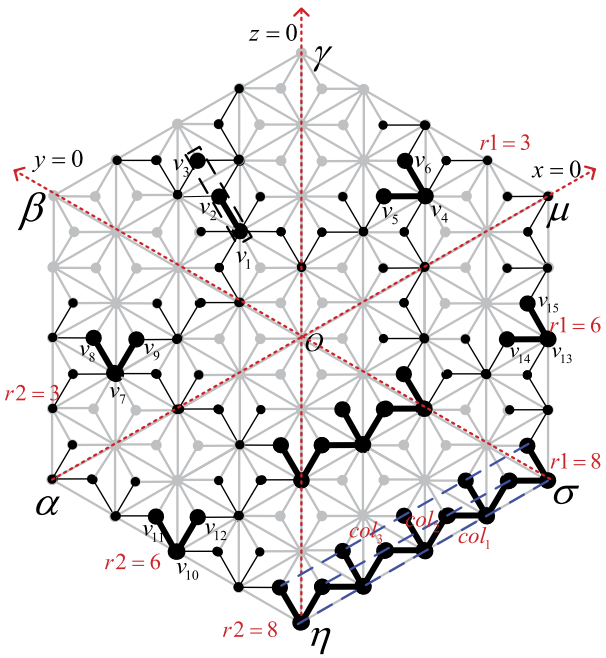



**Fig. 4.** $N_{r1}(\gamma) \cap N_{r2}(\beta)$ in HDN1(5) where $r1, r2 \in \{1, 2, \ldots, 8\}$ and $r1 + r2 > n - 1$. If two different vertices $x$ and $y$ are connected by thick solid black lines or in the same box with black dashed lines, then we have $d(x, w) = d(y, w)$, where $w \in \{\gamma, \beta\}$.

**Lemma 3.4.** $\mathrm{md}(\mathrm{HDN1}(n)) \geqslant 3$ *for* $n \geqslant 3$.

**Proof.** We prove this lemma by contradiction. Suppose $\mathrm{md}(\mathrm{HDN1}(n)) \leqslant 2$ for some $n \geqslant 3$. However, according to Theorem 3.1, $\mathrm{md}(\mathrm{HDN1}(n)) \neq 1$. Thus, $\mathrm{md}(\mathrm{HDN1}(n)) = 2$. Fig. 3(2) shows HDN1(3). Assume that $\{a, b\}$ is a metric basis of HDN1$(n)$ for $n \geqslant 3$. According to Theorem 3.2, we have $\deg(a) = \deg(b) = 3$. Clearly, $d(a, b) \geqslant 2$. Assume that vertex $c \in V(\mathrm{HDN1}(n))$ is on a shortest path between $a$ and $b$ such that $c$ is adjacent to $a$ or $b$. According to Theorem 3.2, we have $\deg(c) \leqslant 5$. Thus, $c \in \{\gamma, \beta, \alpha, \eta, \sigma, \mu\}$. Without loss generality, let $c = \alpha$, then $a, b \in \{v_1, v_2\}$, where $v_1 = (1, -3(n-1)+2, -3(n-1)+1)$ and $v_2 = (-1, -3(n-1)+1, -3(n-1)+2)$ (see Fig. 3(2)). Obviously, $d(\alpha, v_1) = d(x_1, v_1) = 1$ and $d(\alpha, v_2) = d(x_1, v_2) = 1$, where $x_1 = (0, -3n + 6, -3n + 6)$. Thus, the assumption is not true. □

According to Lemma 3.3 and Lemma 3.4, we have the following corollary.

**Corollary 3.5.** $\mathrm{md}(\mathrm{HDN1}(n)) \geqslant 3$ *for* $n \geqslant 2$.

**Lemma 3.6.** $\mathrm{md}(\mathrm{HDN2}(n)) \geqslant 3$ *for* $n \geqslant 2$.

**Proof.** By contradiction, suppose $\mathrm{md}(\mathrm{HDN2}(n)) \leqslant 2$. However, according to Theorem 3.1, we have $\mathrm{md}(\mathrm{HDN2}(n)) \neq 1$. Thus, $\mathrm{md}(\mathrm{HDN2}(n)) = 2$. Assume that $\{a, b\}$ is a metric basis of HDN2$(n)$. By Theorem 3.2, we have $\deg(a) \leqslant 3$ and $\deg(b) \leqslant 3$. Nevertheless, the degree of all the vertices in HDN2$(n)$ are greater than 4, a contradiction. Hence, the lemma holds. □

**Lemma 3.7.** *For* HDN1$(n)$, $n \geqslant 2$, *we have* $|N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r3}(\eta) \cap N_{r4}(\sigma)| \leqslant 1$, *where* $r1, r2, r3,$ *and* $r4 \in \{1, 2, \ldots, 2(n-1)\}$.

**Proof.** Again by contradiction, suppose that $|N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r3}(\eta) \cap N_{r4}(\sigma)| \geqslant 2$. Then there exists two vertices $u$ and $v$ such that $u, v \in N_{r1}(\gamma) \cap N_{r2}(\beta)$ and $u \neq v$. In HDN1$(n)$, $n \geqslant 2$, if $r1 + r2 \leqslant n - 1$, then $|N_{r1}(\gamma) \cap N_{r2}(\beta)| \leqslant 1$, a contradiction. Thus, in the following, we only consider the case for $r1 + r2 > n - 1$.

With respect to the values of $r1$ and $r2$, we have the following six different cases. In Cases 1–5, $|N_{r1}(\gamma) \cap N_{r2}(\beta)| \leqslant 3$, while in Case 6, $|N_{r1}(\gamma) \cap N_{r2}(\beta)| \geqslant 5$. Without loss of generality, in the following proof we can only deal with these cases.

**Case 1**: $1 \leqslant r1 \leqslant n-1$, $1 \leqslant r2 \leqslant n-1$, and $r1+r2 > n-1$. Obviously, $u$ and $v$ are inside of $\Delta\beta O\gamma$ and the coordinates of $u$ and $v$ satisfy that $x \geqslant 0$, $y \geqslant 0$, and $z \leqslant 0$. Let $v_1 = (x, y, y-x)$, $v_2 = (x+2, y+1, y-x-1)$, and $v_3 = (x+4, y+2, y-x-2)$, where $x = 3k$ with $0 \leqslant k \leqslant n-2$ and $y = 3h$ with $0 \leqslant h \leqslant k$. Clearly, $u, v \in \{v_1, v_2, v_3\}$ (see Fig. 4).

Subcase 1.1. $u = v_1$ and $v = v_2$. Then, $d(v_1, \eta) = d(v_2, \eta) - 1$.

Subcase 1.2. $u = v_1$ and $v = v_3$. Then, $d(v_1, \eta) = d(v_3, \eta) - 2$.

Subcase 1.3. $u = v_2$ and $v = v_3$. Then, $d(v_2, \eta) = d(v_3, \eta) - 1$.

Thus, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r3}(\eta)$.

**Case 2**: $1 \leqslant r1 \leqslant n-1$ and $n-1 < r2 \leqslant 2(n-1)$. Obviously, $u$ and $v$ are inside of $\Delta\gamma O\mu$ and the coordinates of $u$ and $v$ satisfy that $x \geqslant 0$, $y > 0$, and $z > 0$. Let $v_4 = (x, y, y-x)$, $v_5 = (x+1, y-1, y-x-2)$, and $v_6 = (x+2, y+1, y-x-1)$, where $x = 3k$ with $0 \leqslant k \leqslant n-2$ and $y = 3h$ with $k < h \leqslant n-1$. Clearly, $u, v \in \{v_4, v_5, v_6\}$ (see Fig. 4).

Subcase 2.1. $u = v_4$ and $v \in \{v_5, v_6\}$. Then, $d(u, \sigma) = d(v, \sigma) - 1$. Thus, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r4}(\sigma)$.

Subcase 2.2. $u = v_5$ and $v = v_6$. Then, $d(v_5, \eta) = d(v_6, \eta) - 1$. Thus, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r3}(\eta)$.

**Case 3**: $n-1 < r1 \leqslant 2(n-1)$ and $1 \leqslant r2 \leqslant n-1$. Obviously, $u$ and $v$ are inside of $\Delta\beta O\alpha$ and the coordinates of $u$ and $v$ satisfy that $x \geqslant 0$, $y < 0$, and $z < 0$. The proof is analogous to Case 2. Let $v_7 = (x, y, y-x)$, $v_8 = (x+2, y+1, y-x-1)$, and $v_9 = (x+1, y+2, y-x+1)$, where $x = 3k$ with $0 \leqslant k \leqslant n-2$ and $y = -3h$ with $0 < h \leqslant n-1-k$. Clearly, $u, v \in \{v_7, v_8, v_9\}$ (see Fig. 4).

Subcase 3.1. $u = v_7$ and $v \in \{v_8, v_9\}$. Similar to Subcase 2.1, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r3}(\eta)$.

Subcase 3.2. $u = v_8$ and $v = v_9$. Similar to Subcase 2.2, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r4}(\sigma)$.

**Case 4**: $n-1 < r1 \leqslant 2(n-1)$, $n-1 < r2 \leqslant 2(n-1)$, and $r1 > r2$. Obviously, $u$ and $v$ are inside of $\Delta\alpha O\eta$ and the coordinates of $u$ and $v$ satisfy that $x < 0$, $y < 0$, and $z < 0$. Let $v_{10} = (x, y, y-x)$, $v_{11} = (x+2, y+1, y-x-1)$, and $v_{12} = (x+1, y+2, y-x+1)$, where $x = -3k$ with $0 < k \leqslant n-2$ and $y = -3h$ with $k < h \leqslant n-1$. Clearly, $u, v \in \{v_{10}, v_{11}, v_{12}\}$ (see Fig. 4).

Subcase 4.1. $u = v_{10}$ and $v \in \{v_{11}, v_{12}\}$. Then, $d(u, \eta) = d(v, \eta) - 1$. Thus, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r3}(\eta)$.

Subcase 4.2. $u = v_{11}$ and $v = v_{12}$. Then, $d(v_{11}, \sigma) = d(v_{12}, \sigma) + 1$. Thus, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r4}(\sigma)$.

**Case 5**: $n-1 < r1 \leqslant 2(n-1)$, $n-1 < r2 \leqslant 2(n-1)$, and $r1 < r2$. Obviously, $u$ and $v$ are inside of $\Delta\mu O\sigma$ and the coordinates of $u$ and $v$ satisfy that $x < 0$, $y > 0$, and $z > 0$. The proof is analogous to Case 4. Let $v_{13} = (x, y, y-x)$, $v_{14} = (x+1, y-1, y-x-2)$, and $v_{15} = (x+2, y+1, y-x-1)$, where $x = -3k$, $0 < k \leqslant n-2$ and $y = 3h$, $1 \leqslant h \leqslant n-1-k$. Clearly, $u, v \in \{v_{13}, v_{14}, v_{15}\}$ (see Fig. 4).

Subcase 5.1. $u = v_{13}$ and $v \in \{v_{14}, v_{15}\}$. Similar to Subcase 4.1, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r4}(\sigma)$.

Subcase 5.2. $u = v_{15}$ and $v = v_{14}$. Similar to Subcase 4.2, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r3}(\eta)$.

**Case 6**: $n-1 < r1 \leqslant 2(n-1)$, $n-1 < r2 \leqslant 2(n-1)$, and $r1 = r2$. The coordinates of $u$ and $v$ satisfy that $x < 0$, $y \leqslant 1$, and $z \geqslant -1$. Let line segments $col_1$, $col_2$, and $col_3$ parallel to $x$-axis, respectively (see three dashed lines in Fig. 4). Vertices on line segment $col_1$ are $(x, 0, -x), (x, -3, -3-x), \ldots, (x, -3h, -3h-x)$ in turn with $0 \leqslant h \leqslant -x/3$, where $x = -3k$ and $0 < k \leqslant n-1$. Vertices on line segment $col_2$ are $(x+1, -1, -2-x), (x+1, -4, -5-x), \ldots, (x+1, -3j+2, -3j+1-x)$ in turn, where $0 < j \leqslant -x/3$. Vertices on line segment $col_3$ are $(x+2, 1, -1-x), (x+2, -2, -4-x), \ldots, (x+2, -3m+1, -3m-1-x)$ in turn, where $0 \leqslant m \leqslant -x/3$. Obviously, $u$ and $v$ belong to the vertices of line segments $col_1$, $col_2$, or $col_3$. Let $u = (x_1, y_1, z_1)$ and $v = (x_2, y_2, z_2)$, we have the following subcases.

Subcase 6.1. $u \in \{col_1, col_2\}$ such that $x_1 = x_2 - 1$, $y_1 = y_2 + 1$, and $z_1 = z_2 + 2$. Then, $d(u, \sigma) = d(v, \sigma) - 1$. Thus, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r4}(\sigma)$.

Subcase 6.2. $u \in col_1$ such that $x_1 = x_2 - 2$, $y_1 = y_2 + 2$, and $z_1 = z_2 + 4$. Then, $d(u, \sigma) = d(v, \sigma) - 2$. Thus, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r4}(\sigma)$.

Subcase 6.3. $u$ and $v$ do not satisfy the relationship in Subcases 6.1, 6.2. It is easily to verify that $d(u, \eta) \neq d(v, \eta)$. Thus, we have $u, v \notin N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r3}(\eta)$.

In summary, there is no pair of $u$ and $v$ such that $u, v \in N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r3}(\eta) \cap N_{r4}(\sigma)$ and $u \neq v$. Thus, we get a contradiction. □

**Lemma 3.8.** *$\{\gamma, \beta, \eta, \sigma\}$ is a locating set of* HDN1$(n)$ *for $n \geqslant 2$.*

**Proof.** We prove this lemma by contradiction. Suppose that $\{\gamma, \beta, \eta, \sigma\}$ is not a locating set for HDN1$(n)$. Then, there exists one pair of vertices $u$ and $v$ such that $u, v \in V(\text{HDN1}(n))$ with $u \neq v$ and $d(u, w) = d(v, w)$ for any $w \in \{\gamma, \beta, \eta, \sigma\}$.

For $r1, r2, r3, r4 \in \{1, 2, \ldots, 2(n-1)\}$, if $d(u, \gamma) = d(v, \gamma) = r1$, $d(u, \beta) = d(v, \beta) = r2$, $d(u, \eta) = d(v, \eta) = r3$, and $d(u, \sigma) = d(v, \sigma) = r4$, then $u, v \in N_{r1}(\gamma)$, $u, v \in N_{r2}(\beta)$, $u, v \in N_{r3}(\eta)$, and $u, v \in N_{r4}(\sigma)$, i.e. $u, v \in N_{r1}(\gamma) \cap N_{r2}(\beta) \cap N_{r3}(\eta) \cap N_{r4}(\sigma)$. By Lemma 3.7, we know that such $u$ and $v$ do not exist. Thus, we get a contradiction. □

Using the same method as in the proof of Lemma 3.7 and Lemma 3.8, we can prove that $\{\gamma, \beta, \eta, \sigma\}$ is a locating set of HDN2$(n)$.

**Lemma 3.9.** *$\{\gamma, \beta, \eta, \sigma\}$ is a locating set of* HDN2$(n)$ *for $n \geqslant 2$.*

According to Lemmas 3.8 and 3.9, we have the following two corollaries.

**Corollary 3.10.** md(HDN1$(n)$) $\leqslant 4$ *for $n \geqslant 2$.*

**Corollary 3.11.** md(HDN2$(n)$) $\leqslant 4$ *for $n \geqslant 2$.*

According to Corollary 3.5, Lemma 3.6, Corollary 3.10, and Corollary 3.11, we have the following theorem.

**Theorem 3.12.** $3 \leqslant \mathrm{md}(\mathrm{HDN}(n)) \leqslant 4$ *for* $n \geqslant 2$.

## 4. Conclusion

In this paper, we have partly solved the open problem proposed in [9]. We firstly prove $\mathrm{md}(\mathrm{HDN}) \geqslant 3$. Then we give a locating set $W$ of HDN such that $|W| = 4$. Thus, we have obtained $3 \leqslant \mathrm{md}(\mathrm{HDN}(n)) \leqslant 4$ for $n \geqslant 2$.

By enumeration, it's easy to prove that the metric dimension of HDN1(2) is 4. We conjecture that the metric dimension of HDN networks is 4.

## Acknowledgements

We would like to express our warmest gratitude to the anonymous reviewers for their review comments, which helped us in improving the quality of the original manuscript. This work is supported by National Natural Science Foundation of China (61170021), Specialized Research Fund for the Doctoral Program of Higher Education (20103201110018), Program for Science and Technology Innovative Research Team of Soochow University (SDT2012B02) and sponsored by Qing Lan Project.

## References

[1] S. Arumugam, V. Mathew, The fractional metric dimension of graphs, Discrete Math. 312 (9) (2012), http://dx.doi.org/10.1016/j.disc.2011.05.039.
[2] J.A. Bondy, U.S.R. Murty, Graph Theory with Applications, Macmilan, New York, 1997.
[3] G. Chartrand, L. Eroh, M.A. Johnson, O.R. Oellermann, Resolvability in graphs and the metric dimension of a graph, Discrete Appl. Math. 105 (2000) 99–113.
[4] M.-S. Chen, K.G. Shin, D.D. Kandlur, Addressing, routing, and broadcasting in hexagonal mesh multiprocessors, IEEE Trans. Comput. 39 (1990) 10–18.
[5] M. Fehr, S. Gosselin, O.R. Oellermann, The metric dimension of Cayley digraphs, Discrete Math. 306 (2006) 31–41.
[6] M. Imran, A.Q. Baig, S.A.U.H. Bokhary, I. Javaid, On the metric dimension of circulant graphs, Appl. Math. Lett. 25 (2012) 320–325.
[7] I. Javaid, M.T. Rahim, K. Ali, Families of regular graphs with constant metric dimension, Util. Math. 75 (2008) 21–33.
[8] S. Khuller, B. Raghavachari, A. Rosenfeld, Landmarks in graphs, Discrete Appl. Math. 70 (1996) 217–229.
[9] P. Manuel, B. Rajan, I. Rajasingh, C. Monica M, On minimum metric dimension of honeycomb networks, J. Discrete Algorithms 6 (2008) 20–27.
[10] F.G. Nocetti, I. Stojmenovic, J. Zhang, Addressing and routing in hexagonal networks with applications for tracking mobile users and connection rerouting in cellular networks, IEEE Trans. Parallel Distrib. Syst. 13 (2002) 963–971.
[11] A. Sebö, E. Tannier, On metric generators of graphs, Math. Oper. Res. 29 (2004) 383–393.
[12] I. Stojmenovic, Honeycomb networks: Topological properties and communication algorithms, IEEE Trans. Parallel Distrib. Syst. 8 (1997) 1036–1042.
[13] I. Tomescu, Discrepancies between metric dimension and partition dimension of a connected graph, Discrete Math. 308 (2008) 5026–5031.
[14] I.G. Yero, J.A. Rodriguez-Velazquez, A note on the partition dimension of Cartesian product graphs, Appl. Math. Comput. 217 (2010) 3571–3574.

Journal of Computational Information Systems 10: 22 (2014) 9679–9686
Available at http://www.Jofcis.com

# Adaptive Environment Perception Architecture Model for Internet of Things ⋆

Chengkai XU [1], Mei RONG [2,3,*], Guangquan ZHANG [1,3], Yulei GU [1,3], Yuerong SUN [1]

[1] *School of Computer Science and Technology, Soochow University, Suzhou 215006, China*

[2] *Shenzhen Tourism College, Ji'nan University, Shenzhen 518053, China*

[3] *State Key Laboratory of Computer Science, Chinese Academy of Science, Beijing 100190, China*

**Abstract**

With the rapid development of the Internet of Things, adaptive technology for the Internet of Things has gained a great deal of attention. Environment perception as a technology to support adaptive networking system has been widely used. Through analyzing adaptive Internet of Things system, a model of independent environment perception structure is proposed. The structure is divided into three layers. In the collection layer, targeted environmental information is perceived according to the environmental model. In the interconnection layer, other environmental information can be received via the Internet. In the release layer, environmental information is stored and recalled. Finally, environmental information is modeled through timed automata. And the operation of perceptual structure correctness is verified with an example of the greenhouse system by UPPAAL.

*Keywords*: Environment Perception; Internet of Things; Adaptive; Timed Automata

## 1 Introduction

The Internet of things is a network to achieve intelligent identification, location, tracing, monitoring and management, which can make anything connected to the Internet for information exchange through a sensing device and the contract agreement. Its basic characteristic is the transmission, intelligent processing and environment perception. Its core is the interaction between entities including people. Therefore, emphasize the "fusion" between system and environment, the environment perception and the adaptability have become an important demand of the Internet of things system. The environment is a kind of information that can depict the

⋆Project supported by the Natural Science Foundation of Jiangsu Province (BK2011281), Applied Foundation Research Program of Suzhou (SYG201241), Post Graduate Research and Innovation Program of Jiangsu Province (CXLX13_820, KYLX_1247), the Students' Innovation and Entrepreneurship Training Program of Soochow University (2014xj030).

*Corresponding author.
*Email address:* rongmei@sz.jnu.edu.cn (Mei RONG).

1553–9105 / Copyright © 2014 Binary Information Press
DOI: 10.12733/jcis13106
November 15, 2014

entity's state. Entity contains people, places, objects and so on, so the environment information includes with the information related to software implementation, especially the information that can influence system execution. It contains the system's state information, personal information of the user, and any information involved in the interaction with the user and the system [1].

Environment perception [2] is the basic function of the Internet of things and the foundation of things linked. With the rapid development of the Internet of things, the environment perception technology has gained wide attention. Environment perception is core content of pervasive computing, and the basis of the system self-adaptive. The core of the Internet of things is the interaction between entities, the human and object, so the Internet of things system needs self-adaption for the changed environment. Research on adaptive system for Internet of things related to the field of artificial intelligence and software engineering, contains the requirements engineering, control theory and intelligent Agent technology, and so on. Therefore, traditional software development techniques is not adaptive to provide theoretical guarantees and key technology development for the Internet of things system.

Adaptive system for Internet of things runs in the open, complex and evolving environment, systems need to be changed itself running status according to the environmental information, to achieve operational objectives of system. With the help of the control theory to realize the adaptive software system for the Internet of things, the adaptive feedback loop is a key design [3]. This feedback loop contains four processes. In the monitoring phase, sensing the environment data. In the analysis phase, analyzing environmental data and get the key information. In the decision-making stage, determining the adaptive strategy of the system. In the implementation stage, executing the system target instruction.

## 2 Related Work

Environment perception is the basic feature of the Internet of things and the basis for the normal operation of the Internet of things system, provides a source of information for adaptive system for Internet of things. The purpose of environment perception is to get environmental information according to the system requirement, so we must specify the content of the environment, correctly model and describe environmental information. The literature [4] divides environment into three types (such as time, place and the user), and puts forward the method of context aware service recommendation collaborative filtering. Literature [5] proposes a new environmental modeling technology, through the classification of environmental information and formal description, to achieve the intelligent interactive system application in the Internet of things.

Many scholars believe that under the open dynamic environment context aware framework is an important aspect of the environment perception technology, study environment awareness through the establishment and design of environment perception structure. Paper [6] proposes an supporting framework based on dynamic binding. It regards environment as the first class abstraction, and provides a language to abstract and describe the environment in which self-adaptive multi-agent organizations are situated to perceive the environment efficiently. Literature [7] presents a framework of the open environment characteristics. The framework introduces the predicate detection technology, which supports efficient realize the environmental mechanism and building of trusted software system.

In addition, there are many literatures about environment perception technology referencing the

middleware platform and wireless sensor network. The literature [8] introduces an adaptive evolutionary method of wireless sensor network. Depends on the adaptive wireless sensor network, it can improve the efficiency of environmental information collection. Literature [9] analyzes mobile network, presents a unified architecture model according to the environmental data asynchronous distribution .

In the following sections, we design an environment perception structure by using the separation method of the perception module and adaptive mechanism. This method is conducive to reduce coupling of adaptive layer and perception layer, reduce the complexity of adaptive layer. According to the information model to collect environmental information in the perceptual structure, it improves the quality and pertinence of environmental information. And then it sends information to the Internet layer to complete the environmental information storage. It completes the verification on the perceived structure through the model checking tool UPPAAL. At last, through the example of intelligent greenhouse system, it describes the perceptual structure specific operation process.

# 3 Adaptive Environment Perception Module

## 3.1 Adaptive feedback loop

According to the dynamic changing environment adjusting their behavior to adapt to the changing environment is the development target of adaptive system. In order to achieve this goal, the feedback loop is introduced in the adaptive system. The system is divided into two parts: the perception module and adaptive module. User and environment are two entities in system. The loop consists of four processes: perception, analysis, decision-making, execution, as shown in Fig. 1.

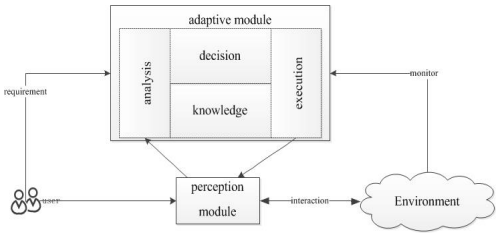



Fig. 1: Feedback loop adaptive system

The perception module and adaptive module constitute a completely adaptive system. In the perception module, system monitors the changing environment data. Adaptive module is decision-making module of the system. It is the core part of the system. In adaptive module, system deals with environmental data to get control command, then completes the adaptive process of system. Separating the perception module and adaptive module, using the independent environment perception structure, mainly due to the following reasons. 1) Changing environment information is transferred to adaptive module as a message, and then system gives instruction to guidance other parts to accomplish the adaptive target, improve the efficiency of the system. 2) Separating the environment perception part is good for information sharing in system modules. 3) Reduce the coupling degree of perception module and adaptive module, and system adaptive layer complexity.

## 3.2 Environment perception structure

The Internet of Things system's goal is to make the information exchange between entities easily. The perception module obtaining information is the foundation of Internet of Things to achieve entities communication. And it is an indispensable mechanism. According to the system's decision, the perception module completes the acquisition and release of environmental information. Environmental information includes software state, network environmental data, information of user and system interaction etc.

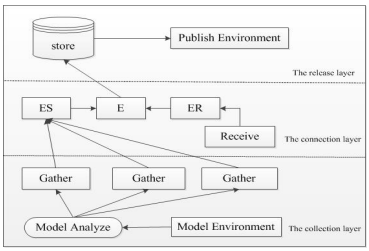



Fig. 2: Sensing structure

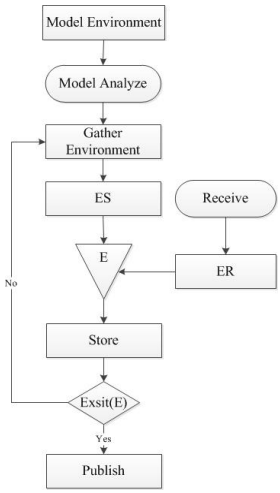



Fig. 3: Environment perception process

As a separate data-getting structure, the perception module is divided into three layers: the acquisition layer, the Internet layer, the distribution layer. It is shown in Fig. 2. The design method is conducive to the realization of the perception module expansion and maintenance. It improves the operation efficiency of module. The role and function of the hierarchy is as follows.

1) The collection layer. This layer provides the environmental information modeling and acquisition function. Environmental information model is used to guide the environmental information collection, including information species, information description etc. The acquisition component obtains acquisition plan by analyzing environmental information model. According to the information model, the layer call the sensor, and send environmental information to the Internet layer.

2) The connection layer. The layer obtains the data information through Internet. The information receiver acquires network information, and analyzes the information, combines with the information of acquisition layer getting environment information E. Then it sends the environmental information to the release layer.

3) The release layer. The layer provides storage and release function of environmental information. The environmental information is stored in the memory. According to the call instruction this layer sends the environment information to system adaptive module and releases the environment information.

According to the environment model collecting information can improve the information pertinence. And it can make different calling strategies based on different kinds of information,

reduce redundant data, improve the information quality and the efficiency of the system. The specific operation process is shown in Fig. 3. So we can get the specific operation process of the environment perception module. It is shown in Algorithm 1.

**Algorithm 1** The operation process of perception module

```
Function Auto-run;
begin:
while true do
    Model(Environment);
    Analyze(Model);
    ES = Sensor(Environment);
    if ∃(ER) then
        ER = Receive(InternetEnvironment);
    end if
    E = ER ∪ ES;
    Store(E);
    while ¬(E) do
        ES = Sensor(Environment);
        ER = Receive(InternetEnvironment);
        E = ER ∪ ES;
        Store(E);
    end while
    Publish(E);
end while
EndAuto − run
```

# 4 Entity Modeling Based on Timed Automata

## 4.1 Timed automata

Here we model the environmental entity and perceive behavior by the timed automata as formal modeling tool. Through the model checking tool UPPAAL achieving time automata behavior simulation and property detection. The symbols are as follows. $Chan$ is a set of channel names. $Act$ is a set of actions, including input, output and internal, three kinds of action, $Act = \{a?|a \in Chan\} \cup \{a!|a \in Chan\} \cup \{R\}$. $Clock$ is all clock variables.

**Definition 1** *A timed automaton is a six tuple $T_a = (S, s_0, C, A, I, \sigma)$, $S$ is a finite set of states, $s_0 \in S$is the initial state; $C$ is the clock set; $A \in Act$ is a finite set of actions; $\sigma S \times A \times B(C) \times 2^C \times S$ is the set of edges, used to describe the state migration, $B(C)$ a set of enable conditions, describing the time constraint of transfer, $2^C$ the clock set of transfer, the mapping of $I: L \to B(C)$ assigned a clock constraint for each state.*

We can use the timed automata to characterize Information processing behavior of subsystem. It communicates with other automata through channel. Timed automata network builds the system into the network model of multiple parallel time automata. Timed automata network share some clock variables and data variables, but they have their own state. It is equivalent to n parallel synthesis of $T_{a_i}(1 \le i \le n)$ in the general clock and behavior set. $N_{T_a} = T_{a_1}|T_{a_1}|\cdots|T_{a_1} = (S, s_0, C, A, I, \sigma)$, $S$ is the state vectors, $s_0$ initial state vector.

### 4.2 Entity modeling

**Definition 2** *Environmental information collection is* $CoInfos = \{D_i | i \in N\}$, *among it* $D_i = \{d_{i1}, d_{i2}, \cdots, d_{im}\}$, $d_{im} = \langle tid, attr, values, op, dom \rangle$, $m \in N$. *tid is a unique identifier for the environment entity, attr represents a collection of attributes of environmental entities, op is set of the operation, dom represents a mapping from attribute set to the set of data types.*


(a) temperature (b) humidity (c) system state

Fig. 4: Automata description of environmental property

Here describing temperature and humidity in the greenhouse system. The temperature is shown in Fig. 4(a): $Temperature \equiv \langle Temp, \{up, down\}, value, \{showTemp\}, dom_{Temp} \rangle$. Simulate temperature through the probability, so control the temperature bellow 40. If the temperature is above 40 then system will take urgent treatment. The humidity is described in Fig. 4(b): $Humidity \equiv \langle Hum, \{up, down\}, value, \{showHum\}, dom_{Hum} \rangle$. If the humidity is not normal, it need take measures. The value of temperature and humidity are transferred by $showTemp!$ and $showHum!$. The information receiving component is shown in Fig. 4(c): $systState \equiv \langle systState, \{normalRun, busyRun\}, value, \{showState\}, dom_{sys} \rangle$. When the state value is less than 6, that the system is normal. Or the system is overloaded or system failure.

In the structure, the perception component is: $Sensor \equiv \langle \{run\}, \{run\}, c, A, I, \sigma \rangle$. Information receiving module is: $\mathrm{R}ecieve \equiv \langle \{recieve\}, \{recieve\}, c, A, I, \sigma \rangle$. Information receiving module simulates to receive network and system state information storage components: $StoreInfo \equiv \langle \{store\}, \{store\}, c, A, I, \sigma \rangle$ the storage component is responsible for the environmental information to be saved, and call information according to the instruction, so it contains the environmental information acceptance and release two processes. These will be described in the next section.

The normal operation of interaction between entities represents the perception structure's operation. Thus to verify the correctness of perception model we need to validate the perception structure activity and state reachability. The validation process and results in UPPAAL are described in the next section.

## 5 Case Study

Taking the greenhouse system as an example, we will simulate operation process of the perception structure in this section. The normal growth of crops in Greenhouse needs to maintain appropriate temperature and humidity. When the greenhouse's temperature and humidity change, system needs to make reasonable control to maintain the proper temperature and humidity. System control temperature and humidity through the control entity shown in Fig. 5, that includes two state $Won$ and $Woff$.

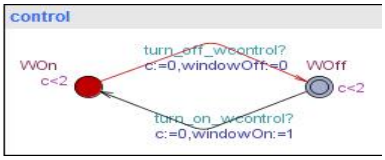
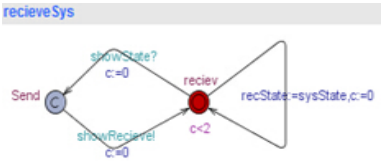


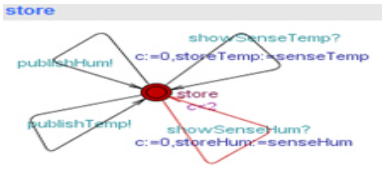

Fig. 5: The window entity description

In this scenario, we model the entity components and environmental entity of environment perception structure. The timed automata network model of environment perception structure is shown in Fig. 6. When the environment changes, perception component get the information as Fig. 6(a). The information receiving component receives the network and system state information as Fig. 6(b). The storage component obtains the information from perception component, stores it. It is shown in Fig. 6(c). In the Fig. 6(d), system calls environmental information according to the current system state information, then makes decisions.

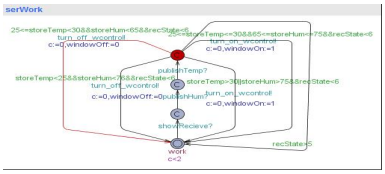

(a) sensor




(b) receiving component




(c) store




(d) call information

Fig. 6: Automata description of environmental property

According the automaton network of the environment perception, we use the tool UPPAAL to verify the system activity and state reachability. Network and system state is also the environmental factor. When the state of the system is higher than 6, it represents system may be in busy state. So the state verification: $E <> senseTemp > 25$, $E <> senseHum > 65$, $E <> recState > 5$. The results is shown in Fig. 7. And activity can be described through the following properties. a) The memory can store the environmental data: $Temperature.senseTemp \wedge Humidity.senseHum \rightarrow Store$. b) System control the widow according system state and environment information: $Store.storeTemp \wedge Store.storeHum \wedge System.recState \rightarrow window$. Its result is shown in Fig. 8. From the trace, those properties are satisfied, so activity is verified. It can guarantee the correctness of the perceptual structure.

# 6 Conclusion

Environment perception is the base of the Internet of Things system adaptive. So it requires accurate describe of the environment information. In this paper, a structure of three layers structure of environmental awareness is presented. The environment and controlling entities are

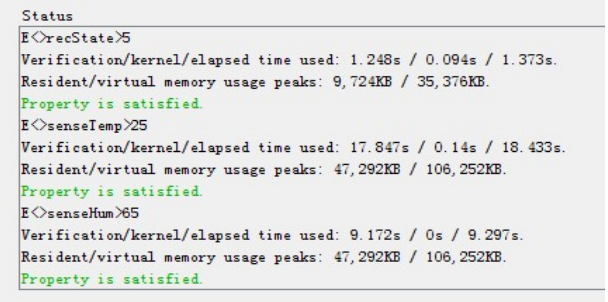



Fig. 7: Verification of temperature and humidity

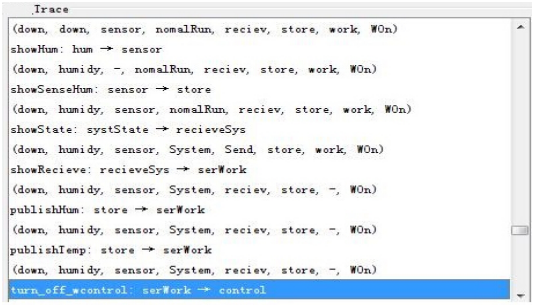



Fig. 8: Simulation trace

modeled by timed automata.At last, this paper verify the reachability and active of perception structure by UPPAAL through the Greenhouse example. There are still shortcomings in the model description of environmental information. The correctness of environmental information authentication and the security of data transmission still need further in-depth research.

# References

[1] E. Pascalau, G. J. Nalepa, K. Kluza. Towards a better understanding of context-aware applications [C]. 2013 Federated Conference on Computer Science and Information Systems (FedCSIS). Piscataway, NJ, USA: IEEE, 2013: 959-962.

[2] A. K. Dey, G. D. Abowd, D. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications [J]. Human-Computer Interaction, 2001, 16(2-4): 97-166.

[3] Y. Brun, G. D. M. Serugendo, C. Gacek, H. Giese, H. Kienle, M. Litoiu, H. Mller, M. Pezzè, M. Shaw. Engineering self-adaptive systems through feedback loops [M]. Software Engineering for Self-Adaptive Systems. Berlin, Germany: Springer-Verlag, 2009.

[4] M. Hongyan, J. Ningkang, S. Wen, H. Linpeng. A Context-aware Modeling Framework for Pervasive Applications [C]. 2012 International Conference on Cloud and Service Computing (CSC 2012). Los Alamitos, CA, USA: IEEE Computer Society, 2012: 40-44.

[5] Y. Song, G. Zeng, H. Pu. Research on the context model of intelligent interaction system in the Internet of Things [C]. 2011 International Symposium on Information Technology in Medicine and Education. Piscataway, NJ, USA: IEEE, 2011: 379-382.

[6] D. Meng-gao, M. Xin-jun, G. Yi, Q. Zhi-chang. Representing and Perceiving Environment of Complex Self-Adaptive Multi-Agent [J]. Journal of Computer Research and Development, 2012, 49(2): 402-412.

[7] H. Yu, Y. Jian-Ping, M. Xiao-Xing, T. Xian-ping, L. Jian. Monitoring Properties of Open Environments [J]. Journal of Software, 2011, 22(5): 865-876.

[8] C. Wang-hu, L. Jing, S. Yong-fu, L. Haoyu, W. Run-ping, C. Zheng-bao. An Adaptive Revolution Approach of WSN for Information Gathering [J]. Journal of Computational Information Systems. 2013, 9(17): 6759-6766.

[9] P. Bellavista, A. Corradi, M. Fanelli, L. Foschini. A survey of context data distribution for mobile ubiquitous systems [J]. ACM Comput. Surv., 2012, 44(4): 1-45.

International Journal of Database Theory and Application
Vol.7, No.3 (2014), pp.13-22
http://dx.doi.org/10.14257/ijdta.2014.7.3.02

# A Succinct String Dictionary Index in External Memory

Guoqing Zhang[1], Mei Rong[2*] and Guangquan Zhang[1,3]

[1]*School of Computer Science & Technology, Soochow University, Suzhou, 215006, China*
[2]*Shenzhen Tourism College, Jinan University, Shenzhen, 518053, China*
[3]*State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Science, Beijing, 100190, China*
**rongmei@sz.jnu.edu.cn*

***Abstract***

*With the coming of the big data age, more and more string dictionaries need to be processed. The existing string dictionary indexes are either too space-consuming, or lack of locality of reference, making them inapplicable in the external memory environment. Targeted with these problems, first we design a new succinct representation of Patricia trie using LOUDS encoding. Then applying it to external memory indexing problem, we propose a new string dictionary index SB-trie, which is not only succinct on space, but also has good locality of reference, making it I/O efficient in external memory environment. Experiments show that SB-trie consumes less space and has greater searching performance in disk environment.*

***Keywords:*** *String Dictionary; Succinct Data Structure; Trie; Big Data Processing*

## 1. Introduction

Recently the rapid development of Internet and mobile technology leads us to the big data age. In the big data age more and more data need to be processed, especially text data. As the basis of text index, string dictionary index is ubiquitously applied in fields like RDF graph, IP datagram classification, search engine and bioinformatics computing *etc*.

Confronted with the challenge of large scale text data, there are 2 categories of solutions. The first is to design a more efficient external memory data structure, increasing the locality of reference. All data is stored in external memory and the needed data is fetched into main memory in little chunks on demand so that the I/O operations are efficient. The second is to compress the data so that under the same resources condition more data can be stored and processed.

On the external memory data structure category, the followings are the recent progresses. Ferragina etc., combining Patricia trie and B+tree, proposed String B-tree [1], which resolves the decrease of performance when the string keys are too long. Because of the independent storage of label strings, each search operation in the String B-tree node needs 2 I/O operations, which worsens the performance when string keys are shorter than 1000 bytes. And because of the use of implicit pointers when storing trie, the space consumption is still too large. Askitis *etc*. adapted the Burst trie to external memory and proposed the B-trie [2]. It adopts a 2 levels structure, the first or root level is an array based trie, the second or leaf level uses a simple mapping structure based on binary search algorithm. Normally the leaf nodes are not fully filled, the space efficiency is not as well as B+tree. A solution commonly used in industry is to compress the B+tree node using front coding to achieve the overall space reduction. The Cache Oblivious String B-tree [3] proposed by Ferragina *etc.* is theoretically analyzed and proved that it not only has good locality of reference characteristics, but also is compressed. But due to the complexity of the structure and the

ISSN: 2005-4270 IJDTA

related algorithms, it has not been implemented and practically tested. The common problem of above indexes is the space consumption.

On the compressed index category, there are also some progresses recently. Klein *etc*. combined front coding and Huffman coding and proposed a new method to match string characters in compressed state directly[4]. Grossi *etc.* proposed path decomposed trie to achieve space compression [5]. Arz *etc.* proposed a method to compress string dictionary based on the classical LZ compression algorithm [6]. Brisaboa *etc.* surveyed 4 methods to compress the index of string collection and conducted a full experiments to compare the performance and trade-offs of these methods [7]. But all the methods are based on the main memory and weak on locality of reference, making them unable to adapt to external memory easily.

Targeted with the problems of these indexes, we first design a new method to succinctly represent a Patricia trie. Then applying it to the external memory index problem, we propose a new succinct string dictionary index in external memory, Succinct B-trie (SB-trie for short), which not only has great locality of reference, but also is succinct on space consumption. Experiments show that compared with existing indexes, this index consumes much less space and has good search performance.

In the following sections of this paper, we will explain the details of the new data structure and the relevant algorithm. Section 2 is the introduction of some basic data structures and algorithms used in SB-trie. Section 3 introduces the succinct representation of Patricia trie and the relevant algorithms. Combining things introduced in Section 2 and 3, the SB-trie and its algorithm is described in Section 4. Section 5 is the experiment and the analysis. Section 6 concludes this paper.

## 2. Basic Data Structures and Algorithms

### 2.1. Bit array

Suppose B[1, n] is a bit array of length n, there are following operations:

- B[i]: the i-th bit in the bit array, $1 \leqslant i \leqslant n$.
- Rank(B, i, b): the count of bit b in sub-array B[1, i], $1 \leq i \leq n$.
- Select(B, i, b): the position of the i-th bit b in the bit array B.

It is first proposed by Jacobson[8], then enhanced by Raman[9] etc. All the operations only cost o(n) extra space and O(1) time complexity.

### 2.2. Trie

Trie is an ordered multi-way tree, node of which consists of a value and multiple branches. Every branch corresponds to a distinct character label. String key is stored on the path from the leaf node to the root node and the corresponding value is stored in the leaf node.

Figure 1 is a typical trie. String keys sharing the same prefix cluster to share nodes. Node sharing lessen node counts and reduces the space consumption. The effect on space-reducing is really similar to the front coding, but the difference is that trie supports fast search while front coding not. Another key characteristic of trie is that trie supports searching pattern P in O(|P|) time complexity, namely it depends only on the length of the pattern to be searched and is independent of the key count stored in the trie.

Given a string key $K_i$, $K_i$ can be a prefix of another string key $K_j$, so the corresponding value of a string key can occur at any node in the trie. In order to avoid ambiguity, we define tail node as follow:

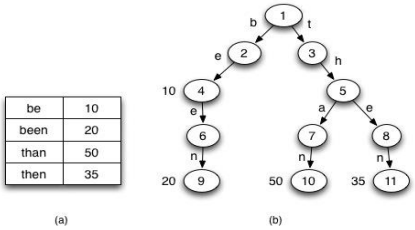



**Figure 1. (a) A Simple String Dictionary, (b) The Corresponding Trie**.

**Definition 1:** the tail node of a string key in a trie is the node from which to the root node stored the string key. The value of the string key stored right in the tail node.

Algorithm 1 is the exact matching algorithm of trie. The invariant of the loop in step 3-7 is that the prefix *P[1, i]* is in the trie and *node* is the tail node of the prefix. Step 8 test the value of the node to determine whether *P* is truly in the trie, or *P* is just a prefix of some other string keys. Taking the trie in Figure 1 for example, given *P* = than, *node* will be node 1, 3, 5 ,7 10 in order, finally the value corresponding to *P* will be found in node 10.

**Algorithm 1:** Exact Match
Input: the trie rooted at *root*, the pattern to be searched *P*.
Output: the value corresponding to *P*, null if not exists.

```
1)  node := root;
2)  i := 1;
3)  WHILE i <= |P|
4)      node := find the branching node corresponding to character P[i] in node.
5)      IF node is null, RETURN null.
6)      i := i + 1
7)  END WHILE
8)  RETURN the value in node.
```

Each node in a trie corresponds to a distinct string, so given a node, we can compute the corresponding string. Algorithm 2 is it. The invariant of the loop in step 3-8 is that the concatenated string from node *tail* to node *node* is the string *key*. Let's see the example in Figure 1. Given *tail* is node 11, *node* will be node 11, 8, 5, 3, 1 in order. Prepending each character in *key*, the resulting *key* is 'then'. Same as Algorithm 1, this algorithm has time complexity O(|P|).

**Algorithm 2:** ComputeStringKey
Input: a trie rooted at *root*, a node in the trie *tail*.
Output: the string key *P* corresponding to node *tail*.

```
1)  node := tail
2)  key := ""
3)  WHILE node != root
4)      parent := Parent(node)
5)      ch :=  find the label character associated with node in parent.
6)      key := ch + key
7)      node := parent
8)  END WHILE
9)  RETURN key
```

### 2.3. The LOUDS Representation of Trie

The traditional pointers based representation of trie costs too much space, especially on 64 bit computers. The LOUDS (Level-Order Unary Degree Sequence) [11] representation is a new succinct representation of trie. Next is the details of the representation.

Given a trie node with 3 branches, the node can be represented by code 1000. The first 1 represent the node itself and the following 3 0s represent 3 branches of the node. Traversing the trie in level order and concatenating the code of each node, we get a bit array. For the convenience of computing, a super root will be added as the parent node of trie root node. So

code 10 will be appended to the bit array. The resulting bit array is just the main part of the LOUDS representation of the trie.

The bit array only records the structure of the trie, we need more data structures to record the remaining information:

- Labels: an array of characters. Used to record the character labels of each trie node. Also concatenated in level order of the trie nodes.
- Values: an array of integers. Used to record the values of each trie node in level order.

Making use of the characteristic of LOUDS representation, the following operations can be supported at the cost of O(1) time complexity. Here we identify each trie node by the first bit 1.

- Branch(id, i), the i-th branching node of the node id: louds.select1(louds.rank0(id + i)).
- Parent(id), the parent node the node id: louds.select1( louds.rank1( louds.select0( louds.rank1(id) - 1 ))).
- Degree(id), the degree of node id: louds. select1( louds.rank1(id) + 1 ) - id - 1.

## 3. The Succinct Representation of Patricia Trie

This section first introduces the exact match and lower bound algorithm of Patricia trie, which will be used in the SB-trie introduced in the next section. We then introduce the succinct representation of Patricia trie and the branch matching algorithm in the succinct representation.

### 3.1. Patricia Trie

Patricia trie is a special trie, it enhances the traditional trie on the space consumption aspect. The traditional trie consists of many non-branching child nodes, which have no contribution to index when searching. When the non-branching nodes becoming a significant fraction, the overall space is wasted by the large portion of pointers. Patricia trie collapses the chain of non-branching nodes of traditional trie into a single branch node with all the character labels concatenated into a string label. Figure 2 is the Patricia trie transformed from the trie in Figure 1.

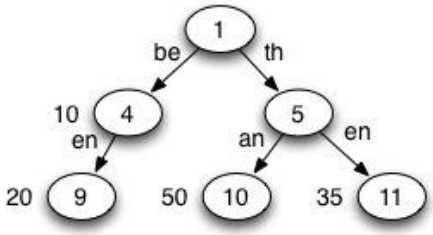



**Figure 2. The Patricia Trie of the Trie in Figure 1**

The exact match algorithm of Patricia trie is similar to trie's. The difference is that now the branch match is against string label instead of character label. Algorithm 3 is the algorithm. Started from the root node, the pattern is used to match the branch nodes continuously. The invariant of the loop is that the prefix P[1, i – 1] equals the string concatenated from the string labels from *node* to *root.* See the example in Figure 2. Given P = than, *node* will be node 1, 5, 10 in order. Finally the value is found in node 10. The time complexity is O(|P|) as well.

**Algorithm 3:** exact match in Patricia trie
Input: a Patricia trie rooted at *root*, the pattern to be searched *P*.
Output: the value corresponding to *P*, null if not exists.

```
1)  node := root
2)  i := 1
3)  WHILE i < |P|
4)      node, i := MatchChild (node, P, i)
5)      IF node is null, RETURN null
6)  END WHILE
7)  RETURN the value in node.
```

Besides the exact match algorithm, there will be a lower bound match algorithm needed.

**Definition 2:** Given a trie constructed from string key collection $S = \{s_1, s_2, .., s_n\}$, the string keys are sorted in lexicographically ascending order. Given the pattern to be searched P, the lower bound match finds the first string key not less than P in S.

The lower bound match makes use of the following property:

**Property 1:** The string keys printed in depth first order of the trie are in lexicographically ascending order.

The definition and property can be extended to Patricia trie naturally. Algorithm 4 is the lower bound match algorithm for Patricia trie. The first part of the algorithm is same as the exact match algorithm. After the loop in step 3-7, the string in *node* is the longest exact matched string in the Patricia trie. At this time, the Patricia trie is split into 3 parts. The first part is the sub-tree rooted at *node*. And the remaining part of Patricia trie was split into 2 parts by the path from *node* to *root*. All string keys in the left part are lexicographically less than *P*, while all string keys in the right part are greater than *P*. Then there are following cases:

- $i > |P|$, namely *P* is exactly matched, then the least string key in the sub-trie rooted at *node* is the result.
- $I <= |P|$, namely the prefix $P[1, i-1]$ is exactly matched. Then we need to find the branching node *next* whose string label is the first label not less than $P[i, |P|]$ via LowerBoundChild operation.
  - If *next* is found, then the least string key in the sub-trie rooted at *next* is the result.
  - If *next* is not found, namely all string keys in the sub-trie rooted in *node* are less than *P*. then the least string key of right part of the Patricia trie is the result.
    - If not found, namely the right part is empty. Matching failed.
    - If found, it is the result.

**Algorithm 4:** LowerBound

Input: a Patricia trie rooted at *root*, the pattern to be searched *P*.

Output: the first string key not less than *P* and the corresponding value. null if not found.

```
1)  node := root
2)  i := 1
3)  WHILE i <= |P|
4)      next, i := MatchChild (node, P, i)
5)      IF next is null, BREAK
6)      node := next
7)  END WHILE
8)  IF i <= |P|
9)      next := ChildLowerBound(node, P[i, |P|])
10)     IF next is null
11)             next := GeneralizedSibling(node)
12)     IF next is null, RETURN null
13)     node := next
14) node := LeftMostNode(node)
15) RETURN the string key and value in node
```

### 3.2. The Succinct Representation of Patricia Trie

Though Patricia trie tries to reduce space cost, it can't be represented succinctly and easily due to the label of each branch is variable length string. We decompose a Patricia trie to 2 tries, which can be represented by LOUDS encoding method.

Suppose the string label of a Patricia trie node is s = as', a is a character and s' is the remaining string of s excluding a. Temporarily ignore the s' part of label, Patricia trie can be represented by LOUDS representation of trie. All the non-empty s' forms another string set, which can be represented by another trie, which we call label trie, which can also be represented by LOUDS representation. Then the missing s' information can be represented by a integer node identifier of the label trie. Just like to represent the value of each trie node, same method can be applied to the node identifier of label trie. This results in an array of integer called Links.

Because the string label of Patricia trie is split into 2 parts, the branch matching algorithm also needs 2 steps, the first step is same as traditional trie matching, to find the branch node corresponding to the first character. The second step is to compute the s' of the branch node and verify that s' is exactly matched as well. Algorithm 5 is the pseudo-code.

**Algorithm 5:** MatchChild
Input: the current node *id*, the whole pattern to be matched *P*, the current position of *P* to be matched *i*.
Output: the branching child node of node *id* corresponds to *P[i, |P|]*, and the new position.

```
1)  offset := louds.rank0(id)
2)  degree := Degree(id)
3)  rank := find position of P[i] in character array Labels[offset, offset + degree – 1]
4)  IF rank is null, RETURN null, i.
5)  child := Branch(id, rank)
6)  label := compute the s' part of node child
7)  IF label is prefix of P[i + 1, ...]
8)      RETURN child, i + length(label)
9)  RETURN null, i
```

Algorithm 6 is the ChildLowerBound algorithm. The first part is similar with Algorithm 5, the steps of 7-10 ensure that the string label is not less than *P*. If it is less than *P*, find the right sibling node of the child node.

**Algorithm 6:** ChildLowerBound
Input: current node *id*, the pattern to be searched *P*.
Output: the first child node whose string label is not less than *P*.

```
1)  offset := louds.rank0(id)
2)  degree := Degree(id)
3)  rank, ch := find the first character not less than P[1] in character array Labels[offset, offset + degree – 1],
    return the position and the character.
4)  IF rank is null, RETURN null
5)  child := Branch(id, rank)
6)  label := compute the s' part of string label of node child
7)  label := ch + label
8)  IF P <= label
9)      RETURN child
10) RETURN Sibling(child)
```

## 4. SB-trie

This section introduces how to integrate the succinct representation of Patricia trie and the relevant algorithms to become the whole SB-trie succinct index in external memory.

### 4.1. The Construction of SB-trie

SB-trie's overall structure is similar to B-trie. There is one central root node and a number of leaf nodes. Each node is logically a Patricia trie decomposed into a index trie and a label trie. Then all the label tries are merged together into a unifying label trie. So SB-trie is composed of a root index trie, a number of leaf index trie and a unifying label trie. And all tries are represented by the LOUDS succinct representation.

The details of construction procedure are as follow. Suppose the input string key collection is $K = \{K_1, .., K_n\}$, the keys are sorted in lexicographically ascending order. First partition K into several groups, each of which contains consecutive keys. The string keys of each group are used to construct a Patricia trie which is transformed into a index trie and a label trie. And each group becomes a leaf node of SB-trie. Then we take the greatest string key of each group as the representative of the node, insert it into a new string dictionary along with the identifier of the responding leaf node. The new string dictionary is used to construct another Patricia trie and then transformed into a index trie and label trie. The index trie becomes the root of the SB-trie. Finally we merge all the label tries into a unifying label trie and stored in LOUDS representation. Figure 3 is a simple diagram of the overall structure of SB-trie.

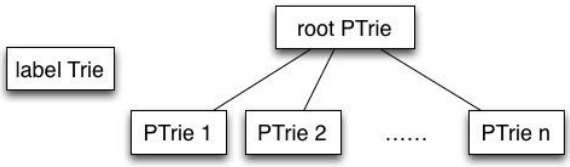



**Figure 3. The Layout of SB-trie**

### 4.2. Search in SB-trie

Algorithm 7 is the pattern searching algorithm of SB-trie. First in step 1, a lower bound match introduced in Section 3.1 is performed to find the leaf node in which *P* possibly occur. If the leaf node is not found, it means *P* is greater than all the string keys stored in the SB-trie, search failed and null is returned. Otherwise the leaf node is found, then the leaf node is fetched into main memory and the exact match introduced in Section 3.1 is performed to find whether *P* really occurs in it. If we cache the root node in main memory, the search algorithm only incurs one I/O operation, so the time complexity is O(1) in external memory model.

**Algorithm 7:** Find
Input: the root of SB-trie *root*, the pattern to be searched *P*
Output: the value corresponding to *P* if exists, null otherwise.
1) *offset* := LowerBound(*root*, *P*)
2) IF *offset* is null, RETURN null
3) *leafNode* := fetch the leaf node at *offset* into main memory.
4) RETURN ExactMatch(*leafNode* , *P*)

## 5. Experiments

We implemented SB-trie and conducted experiments to compare the time and space performance with other existing string dictionary indexes. We adopt the layout used in paper[4] to implement a simple and efficient B+tree for string keys. For front coded B+tree, we just compress each B+tree node and we adopt the one for front coded data structure in paper[6] as search algorithm. For String B-tree we use an open source static implementation[11]. At last, Leveldb is a lightweight key/value store developed by Google. The SB-trie implementation makes use of the bit array with rank/select operation support from sdsl and DAC code is adapted and optimized from the one in paper.

### 5.1. Experiment Environment

The experiment platform is as follow: Intel Core 2 Duo 2.4Ghz CPU, 2G RAM, 160G 5400rpm hard drive, Ubuntu 12.04 64bit LTS OS, FAT32 file system. And all code is written in C/C++ and compiled by g++ 4.6.3 with –O3 optimization.

The data used is the titles of all English Wikipedia entries on 2012/12/01, totally 9864458 items and 201.7MB. it is referred as 10000 kilo items approximately in the following figures. The other smaller scale data are extracted from the beginning part of it.

### 5.2. Experiment Results and Analysis

A comparison experiment is performed with the 5 indexes run on 5 different scale of data. In order to reduce the disturbance of cache, before each run of program, the experiment partition is remounted and the RAM is shuffled by running an auxiliary program. We run the search operation with string keys from the original data set as the search patterns. Every $1000^{th}$ string key of a set of string keys is used to search on the index of that set of string keys. The total run time is averaged to the search time cost per operation. Figure 4(a) shows the search time results, and Figure 4(b) shows the space ratio of each index on each data set. Next is the analysis of each index.

B+tree: because string keys are stored directly in B+tree nodes without any compression, the space consumption is roughly 140% of the size of original data set, regardless of the scale of original data. On the search time aspect, due to multi-level structure, with the increase of the scale of data, the search time cost becomes longer.

Front coded B+tree: due to the adoption of front coding method to compress B+tree nodes, the space consumption ratio decreases with the increase of the scale of data set. On the search time aspect, though the front coding method increase the computation complexity, the decreased space consumption reduced the need to perform I/O operations. The experiment results show that the reduced I/O time complements the increased CPU time, so the total performance is increased.

String B-tree: similar to B+tree, it includes multi-level index structure. The use of implicit pointer based trie makes it too costly on space, even worse than B+tree. Increased space consumption make it need to perform more I/O operation to fulfill search operation. Considering it is designed for long length string keys, and the string keys used in the experiments are mostly short, shorter than 1000 bytes, so the results are expected.

Leveldb: it use the Snappy general purpose compression library to reduce the index space. The effect is similar to front coded B+tree. On search time aspect, it is slower than SB-trie because it supports dynamic operations and other functionalities.

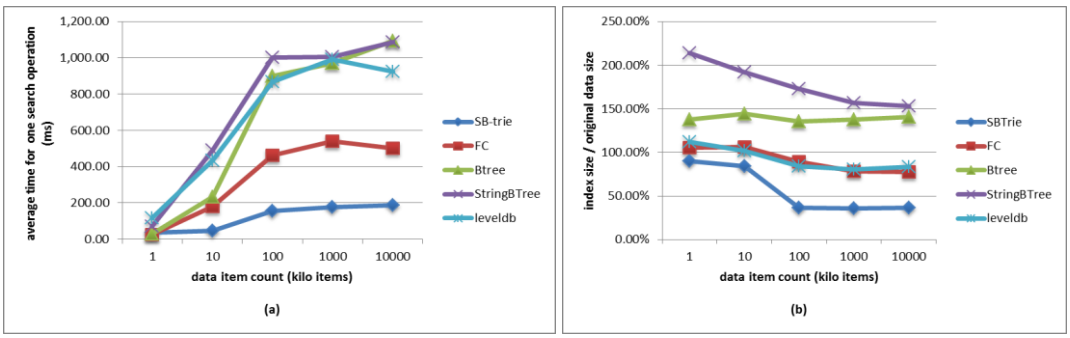



**Figure 4. Time(a) and Space(b) Performance on Different Scale of Data Set**

SB-trie: the experiments results show that the SB-trie outperforms all other indexes on the space aspect. When the scale of data set increases, the space ratio is close to 36%. The index of 201.7MB of original data set only costs 73.1MB. There are 2 main aspects achieving this result. The first is the use of succinct representation of Patricia trie in SB-trie nodes. Patricia trie reduces space cost by sharing same prefix of string keys, and the LOUDS use bit array to record the structure of trie and avoid using implicit pointers. The combination of these 2 methods greatly reduce the cost of the index. Lastly we merge all the label trie into a unifying label trie to reduce redundancy of storing the same string labels more than once. This method reduce the space cost even more.

On search time aspect, though the succinct representation of Patricia trie has greater search time complexity, the greatly reduced space cost reduce the node count, making it possible to use only one root node. So each search operation only needs one I/O operation. The reduced I/O time significantly complements the increased CPU time, so when the scale of data set become greater than 10000 items, the search efficiency outperforms other indexes as well.

## 6. Conclusion

In the big data age, large scale of data has been a challenge of string dictionary index problem. The existing string dictionary indexes are either too space-consuming, or inefficient on I/O operation when working in external memory, which makes it difficult to rise to the challenge. Targeted with these problems, first we design a new succinct representation of Patricia trie. Then applying it to external indexing algorithm, we propose a new string dictionary index data structure SB-trie, which is not only succinct on space, but also has good locality of access, making it I/O efficient in external memory environment. Experiments show that SB-trie consumes less space and has greater searching performance in disk environment. But SB-trie is only a static index, it does not support dynamic operations like insertion or deletion. So how to adapt it to support dynamic operation will be a more challenging problem.

## Acknowledgements

We are grateful to the authors of the open source libraries for kindly sharing them. The work of this paper is supported by the Natural Science Foundation of Jiangsu, China (Grant No. BK2011281) and the Applied Fundamental Research Program of Suzhou, China (Grant No. SYG201241).

## References

[1] P. Ferragina and R. Grossi, "The string B-tree: a new data structure for string search in external memory and its applications", Journal of the ACM (JACM), vol. 46, no. 2, **(1999)**, pp. 236-280.
[2] N. Askitis and J. Zobel, "B-tries for disk-based string management", The VLDB Journal, vol. 18, no. 1, **(2009)**, pp. 157-179.
[3] P. Ferragina, R. Grossi, A. Gupta, R. Shah and J. S. Vitter, "On searching compressed string collections cache-obliviously" in Proceedings of the twenty-seventh ACM SIGMOD-SIGACT- SIGART symposium on Principles of database systems, ACM, **(2008)**, pp. 181-190
[4] S. T. Klein and D. Shapira, "Compressed Matching in Dictionaries", Algorithms, vol. 4, no. 4, **(2011)**, pp. 61-74.
[5] R. Grossi and G. Ottaviano, "Fast compressed tries through path decompositions" in Proceedings of the 14th Meeting on Algorithm Engineering & Experiments, ALENEX, Kyoto, Japan: **(2012)**, pp. 65-74
[6] J. Arz and J. Fischer, "LZ-Compressed String Dictionaries", CoRR, **(2013)**.
[7] N. Brisaboa, R. Cánovas, F. Claude, M. Martínez-Prieto and G. Navarro, "Compressed string dictionaries", Experimental Algorithms, **(2011)**, pp. 136-147.
[8] G. Jacobson, "Space-efficient static trees and graphs", Foundations of Computer Science, **(1989)**, pp. 549-554.
[9] R. Raman, V. Raman and S. R. Satti, "Succinct indexable dictionaries with applications to encoding k-ary trees, prefix sums and multisets", ACM Trans. Algorithms, vol. 3, no. 4, **(2007)**, pp. 43-68.
[10] D. Arroyuelo, R. Cánovas, G. Navarro and K. Sadakane, "Succinct trees in Practice", Proceedings of the Twelfth Workshop on Algorithm Engineering and Experiments, ALENEX, Austin, Texas, USA, **(2010)**, pp. 84-97.
[11] J. Lemoine, "Text Mining Software - C++ string b-tree Library", http://wikipedia-clustering.speed, blue.org/strBTree.php, **(2013)**, pp. 7-20.

## Authors

**Guoqing Zhang**, he received his Bachelor of Science degree from Huaiyin Normal University, Jiangsu, P.R. China in 2010. Now he is a graduate student majoring in Computer Software and Theory at Soochow University in P.R. China. His current researches focus on data compression and data index algorithms.

**Mei Rong**, she received the PhD degrees in Computer Science from Chongqing University, in 1998, respectively. She is currently an associate professor in the Shenzhen Tourism College, Jinan University, China, and is the member of CCF. Her research interests include software engineering, formal methods and Cyber Physical Systems.

**Guangquan Zhang**, he received his Ph.D. of Computer Software and Theory from Chongqing University, P.R. China in 1999. He is a professor at School of Computer Science and Technology, Soochow University, China and a senior member of China Computer Federation. His research interests include software engineering, cloud computing, CPS and formal methods.

# 基于重复博弈的 Ad hoc 网络合作转发模型

张华鹏 张宏斌*
(苏州大学计算机科学与技术学院 苏州 215006)

**摘 要**：针对噪音环境下的 Ad hoc 网络合作问题，运用不完美信息重复博弈模型分析节点之间的交互过程，使用贝尔曼方程构造满足序贯均衡的合作激励机制。对于该机制，节点间无需交换观察信息，节省了节点能量和网络负担。与已有的序贯均衡策略相比，该机制避免使用对观测误差敏感的触发策略，提高了不完美信息环境下网络的合作率和节点的平均收益。仿真结果表明，使用贝尔曼方程构造的序贯均衡策略既提高了网络的合作率，又有很好的适应性。

**关键词**：Ad hoc；不完美信息；重复博弈；序贯均衡；合作
**中图分类号**：TP393 **文献标识码**： A **文章编号**：1009-5896(2014)03-0703-05
**DOI**: 10.3724/SP.J.1146.2013.00559

# Cooperative Forwarding Model Based on Repeated Game in Ad hoc Networks

Zhang Hua-peng Zhang Hong-bin
(*School of Computer Science & Technology*, *Soochow University*, *Suzhou* 215006, *China*)

**Abstract**: To motivate nodes to forward packets in Ad hoc networks under the condition of imperfect information, repeated game is often used to model the process of sequential interactions between nodes and the Bellman equations is applied to design strategy based on sequential equilibrium. The nodes need not to exchange their monitor information, so it saves nodes' energy and reduces the network burden. Compared with existing sequential equilibrium strategy, since the proposed strategy does not use trigger mechanism, which is sensible to observation error, it improves the cooperation rate and the average benefits of networks with imperfect information. Simulation compares the performance of two sequential equilibrium strategies. The results indicate that the proposed strategy not only improves the cooperation rate, but also has a good adaptability.

**Key words**: Ad hoc; Imperfect information; Repeated game; Sequential equilibrium; Cooperation

## 1 引言

Ad hoc网络是一种无需基础设施支持的自组织网络，网络的功能通过节点间合作实现。在实际的网络中，由于节点存在自私性使得它们为了保存能量而不愿参加与己无关的数据转发活动。经研究发现，节点的自私行为不仅严重降低网络整体性能，而且威胁网络安全[1]。因此，在一些领域的Ad hoc网络应用中，设计促进网络中节点合作的策略机制变得十分关键。

Ad hoc 网络中的合作激励机制分为两类：基于信用的机制和基于信誉的机制。基于信用的机制[2−6]将经济学中的激励手段应用到 Ad hoc 网络中，但该类机制需要使用防更改硬件或第 3 方信任机构，这影响了该类机制的推广应用。在基于信誉的机制中[1,7−13]，节点观察其它节点的行为，并根据信誉推断网络中的自私节点。

对于基于信誉的机制，可以使用博弈模型分析节点间的交互并给出相应的均衡策略[7,8,14,15]。文献[7]运用合作博弈模型对节点间交互进行建模，但节点在选择策略时需进行复杂的信息交换，消耗的能量较多。针对该问题，文献[8]采用非合作博弈模型分析节点间的交互。非合作博弈模型不仅不需要节点间交换信息，而且能够很好地反应 Ad hoc 网络节点的困境：是为了增加自身的信誉值选择合作还是为了节省能量选择不合作。文献[8]使用完美信息重复博弈模型分析节点间的交互，并给出满足子博弈完美均衡的策略组合。然而在 Ad hoc 网络中，由于无线信道容易受到干扰等特点，大多数情况下节点掌握的信息是不完美信息。针对该问题，文献[15]采用不完美信息重复博弈模型分析节点间的交互，

2013-04-24 收到，2013-11-08 改回
国家自然科学基金(61070169)和江苏省自然科学基金(BK2011 22394)资助课题
*通信作者：张宏斌 zhanghb@suda.edu.cn

并给出对应的序贯均衡策略。但该文献使用简单的触发机制构造序贯均衡策略，当观测误差较大时，网络的合作率较低。

本文在文献[15]的基础上，给出了不完美信息下另一种满足序贯均衡的策略组合。与文献[15]中的机制相比，本文使用贝尔曼方程构造策略，避免使用对观测误差非常敏感的触发机制，提高了不完美信息环境下网络的合作率。

## 2 不完美信息重复博弈

### 2.1 博弈模型

假定Ad hoc网络中的节点可以使用watchdog[1]等机制观察其它节点的行为。由于无线信道的特点，节点的观察存在误差，故节点掌握的信息是不完美信息。将一段时间内网络中任意一对节点之间的交互看作一轮两人静态博弈$G$。因此可以使用不完美信息重复博弈模型$\Gamma(G,\delta,\lambda)$分析Ad hoc网络中任意一对节点间的交互，其中$\delta$为贴现因子，说明了该对节点继续博弈的概率，$\lambda$为观测误差。

对于重复博弈$\Gamma$，博弈双方为网络中的任意一对节点，记为$i$和$j$。在单轮博弈$G$中，博弈双方同时从行为空间$A=\{F,D\}$选择行为，$F$为转发对方数据包，$D$为丢弃对方数据包；并观察对方的行为，$f$为观察到对方的转发行为，$d$为观察到对方的丢包行为，令$\Omega=\{f,\ d\}$。由watchdog等机制的原理可知：对于对方的转发行为$F$，博弈者依概率$1-\lambda$观察到$f$，依概率$\lambda$观察到$d$；对于对方的丢包行为$D$，博弈者仅能观察到$d$。博弈者的转发行为$F$消耗节点资源，对方转发自身数据包时节点会获得利益，假定数据包的价值大于转发行为的成本，归一化后的收益矩阵如图1所示。图中$g>0$，$l>0$。假定$u_i^t$表示在第$t$轮博弈中节点$i$的收益，那么节点$i$在重复博弈$\Gamma$中的收益为

$$U_i^t=\sum_{k=t}^{\infty}\delta^k u_i^k \tag{1}$$

| | $F$ | $D$ |
|---|---|---|
| $F$ | $l,l$ | $-l,g+l$ |
| $D$ | $g+l,-l$ | 0,0 |

图1 博弈$G$中的收益矩阵

令$a_i^t$表示在第$t$轮博弈中节点$i$的行为，$\omega_i^t$表示在第$t$轮博弈中节点$i$的观察结果，且边界条件为$\omega_i^0=f$。在第$t$轮博弈中，$i$的博弈历史为节点行为和观察信息序列，记为

$$h_i^t=\{a_i^\tau,\omega_i^\tau\}_{\tau=0}^{t-1} \tag{2}$$

令$H_i^t=(A\times\Omega)^t$表示在第$t$轮博弈中，$i$所有可能的博弈历史集合，那么节点$i$的策略$s_i$是博弈序列到节点行为的映射，$s_i:H_i^t\to A,\ (t=0,\ 1,\cdots)$。

### 2.2 序贯均衡

对于不完美信息重复博弈，序贯均衡[16]不仅满足一致性，而且满足序贯理性，即对于博弈双方的信念以及所有可能到达的信息集[17]，每个博弈方的策略都是针对其他博弈方策略的最佳对策。因此满足序贯均衡的激励机制比满足纳什均衡的机制有更好的稳定性和适用性。

**定义 1** 称$s=(s_i,s_j)$为无显著偏离(no observable deviation[18])的策略组合。若对于每一个博弈者$i$，其任何一个没有到达的信息集可以且仅可以通过博弈者$i$本身偏离策略$s_i$到达。

**定理 1** 对于不完美信息无限次重复博弈$\Gamma$，若纳什均衡策略$s$为无显著偏离的策略组合，且$s$在博弈$\Gamma$中没有到达的信息集上也构成纳什均衡策略，那么$s$是博弈$\Gamma$的序贯均衡策略。证明见文献[18]。

### 2.3 已有模型分析

文献[15]使用不完美信息重复博弈分析节点间交互，并提出满足序贯均衡的策略机制。文献[15]在构造策略时使用两种连续策略：触发策略$\sigma_F$和背叛策略$\sigma_D$。

$$\left.\begin{aligned}&\sigma_F:\begin{cases}a_i^0=F\\ a_i^t=\begin{cases}F,\ \omega_i^k=f,\ k=0,1,\cdots,t-1\\ D\ ,\ \text{其它}\end{cases}\end{cases}\\ &\sigma_D:\ a_i^t=D,\ \ t=0,1,\cdots\end{aligned}\right\} \tag{3}$$

触发机制对观察误差比较敏感，若节点的观察结果在$t$轮之前不全为$f$，那么在$t$轮之后的博弈中始终选择背叛行为$D$。因此，在观察误差较大的环境中，该类机制的合作率较低。本文采用基于贝尔曼方程的序贯均衡策略所采取方法并不是“触发机制”而是针对背叛策略，采取一定程度忍耐方式，从而抵消文献[15]中由于观测误差导致合作率不高的结果出现。第5节进一步分析该类机制，并通过仿真比较两种序贯均衡策略的性能。

## 3 序贯均衡策略

### 3.1 构造策略

假定博弈双方的行为$a_i^t$以及观察到对方的行为$\omega_i^t$是公共信息(public information)，在该假设下的博弈记为$\Gamma'(G,\delta,\lambda)$。针对博弈$\Gamma'$，求解节点$i$的策略$s_i$。在构造节点$i$的策略$s_i$时，应满足在博弈的任何阶段节点$j$选择$F$行为和$D$行为节点$j$所获得的总收益都相等。策略$s_i$仅根据节点$i$的观察信息$\omega_i^t$选择行为$a_i^t$。节点$j$按照同样方法构造策略$s_j$。易知$s=(s_i,s_j)$构成博弈$\Gamma'$的纳什均衡，对于策略组合$s$，节点仅根据自身的观察信息选择行为，因此$s$也是博弈$\Gamma$的纳什均衡。

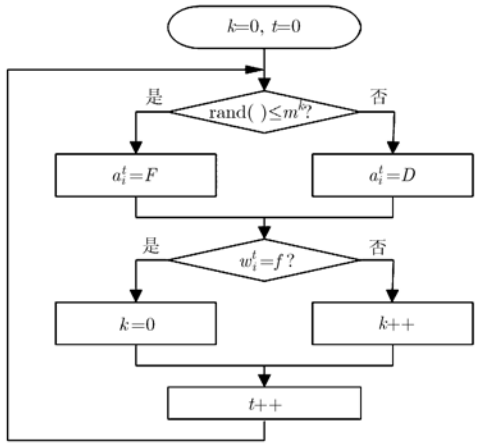



图 2 节点策略构造流程图

图 2 描述节点策略的构造。初始化阶段，节点根据网络确定收益矩阵中的参数$g$, $l$, 初始化参数$t$, $k$, $t$=0, $k$=0，并计算出数列$\{m^k\}$。在第$t$轮博弈中，节点首先确定自身的行为：以概率$m^k$选择合作行为$F$; 以概率 $1-m^k$选择不合作行为$D$。在每一轮博弈结束时，节点观察对方的行为：若观察到对方选择合作行为(即 $\omega_i^t = f$)，将观察到对方连续不合作次数归零($k = 0$)；若观察到对方选择不合作行为(即 $\omega_i^t = d$)，则$k = k+1$。构造策略$s$的关键是如何确定 $m^k$, $0 \le m^k \le 1$, $(k = 0, 1, \cdots)$，使得对方在任何博弈点选择$F$行为和$D$行为获得的总收益相等。

节点策略构造算法如表 1 所示。

**表 1 节点策略构造算法**

| | |
|---|---|
| 步骤 1 | 初始化参数： |
| | 根据网络设置 $g$, $l$; $t$=0, $k$=0; |
| 步骤 2 | 选择本轮行为： |
| | 以$m^k$的概率选择合作行为$F$, |
| | 以 $1-m^k$的概率选择不合作行为$D$; |
| 步骤 3 | 观察对方在本轮的行为： |
| | 若观察到对方的合作行为，则　$k$=0, |
| | 若观察到对方的不合作行为，则　$k$++; |
| 步骤 4 | 继续下一轮博弈: |
| | $t$++, |
| | 转步骤 2。 |

假定在第 $t$ 轮博弈中，节点连续 $k$ 次观察到对方的丢包行为($\omega_i^\tau = d$, $(\tau = t-1, t-2, \cdots, t-k+1)$ 且 $\omega_i^{t-k} = f$)，从本轮博弈开始，对方的总收益为

$$V^k = \sum_{\tau=t}^{\infty} \delta^{\tau-t} u^t \tag{4}$$

在首轮博弈中($t = 0$, $k = 0$)，节点“友好”地选择转发行为；节点观察到对方的转发行为时($\omega_i^t = f$, $k = 0$)，同样选择转发行为。根据对方选择转发行为 $F$ 和丢包行为 $D$ 时(对方)所得收益，可得贝尔曼方程：

$$\left.\begin{aligned} V^0 &= (1-\delta)(1-\lambda-l\lambda)+\delta\left[(1-\lambda)V^0+\lambda V^1\right], \\ &\quad \text{对方转发行为}F \\ V^0 &= (1-\delta)(1-\lambda)(g+1)+\delta V^1, \\ &\quad \text{对方丢包行为}D \end{aligned}\right\} \tag{5}$$

解该方程得

$$\left.\begin{aligned} V^0 &= \frac{1-(g+l+2)\lambda+(g+1)\lambda^2}{1-\lambda} \\ V^1 &= \frac{1-(1-\delta)(g+1)+(g-l-2\delta(g+1))\lambda+\delta(g+1)\lambda^2}{\delta(1-\lambda)} \end{aligned}\right\} \tag{6}$$

更一般的情况，当节点连续 $k$ $(k > 0)$次观察到对方的丢包行为时，根据对方选择转发行为和丢包行为时(对方)所得收益，可得贝尔曼方程：

$$\left.\begin{aligned} V^k &= (1-\delta)(m^k(1-(l+1)\lambda)-(1-m^k)l) \\ &\quad +\delta((1-\lambda)V^0+\lambda V^{k+1}),\ \text{对方选择转发行为} \\ V^k &= (1-\delta)(1-\lambda)m^k(g+1)+\delta V^{k+1}, \\ &\quad \text{对方选择丢包行为} \end{aligned}\right\} \tag{7}$$

解该方程可得

$$\left.\begin{aligned} m^k &= \frac{(1-\lambda)V^k+l-\delta(l+1)+\delta(g+l+2)\lambda-\delta(g+1)\lambda^2}{(1-\delta)(1-\lambda)(l+1-(g+1)\lambda)} \\ V^{k+1} &= \frac{l-g}{\delta(l+1-(g+1)\lambda)}V^k \\ &\quad -(g+1)\frac{l-\delta(l+1)+\delta(g+l+2)\lambda-\delta(g+1)\lambda^2}{\delta(l+1-(g+1)\lambda)} \end{aligned}\right\} \tag{8}$$

式(6)求出 $V^0$ 和 $V^1$ 的值，式(8)给出了 $V^{k+1}$ 与 $V^k$ 的递推关系，可得 $V^k(k = 0, 1, \cdots)$。式(8)给出了 $m^k$ 与 $V^k$ 的关系，因此可以由 $V^k$ 得到 $m^k(k = 0, 1, \cdots)$，进而得到节点的策略。

### 3.2 策略分析

**引理 1**　对于博弈 $\Gamma(G, \delta, \lambda)$，策略组合$s=(s_i, s_j)$为无显著偏离策略。

**证明：**以二元组$(a_i^t, \omega_i^t)$标识节点$i$的信息集，$a_i^t \in \{F, D\}$, $\omega_i^t \in \{f, d\}$。每一轮博弈结束时，节点可能到达的信息集有 4 种情况，分别为$\{F, f\}$, $\{F, d\}$, $\{D, f\}$和$\{D, d\}$。当节点$j$选择转发行为$F$时，节点$i$依概率$1-\lambda$观察到转发行为$f$, 依概率$\lambda$观察到丢包行为$d$，因此只要$j$在第$t$轮博弈中选择转发行为$F$的概率大于零(偏离)，节点$i$就可能通过偏离策略$s_i$到达所有的信息集。

由$s$可知，在首轮博弈中，节点$j$选择转发行为$F$，因此$i$可以通过偏离$s_i$到达所有信息集。假定在第$t(t>0)$轮博弈中，节点$j$选择转发行为的概率是$m^k$。若$m^k=0$，那么节点$i$可以通过在第$t-1$轮博弈中选择$F$行为使得$m^k>0$，因此在第$t$轮博弈中$i$可以通过偏离策略$s_i$到达所有的信息集。综上所述，节点$i$可以通过偏离策略$s_i$到达任何信息集。$j$与$i$采用相同策略，因此$j$也可以通过偏离$s_j$到达任何信息集，$s$无显著偏离。 证毕

**定理 2** 若 $s_i^*=s_i$, $s_j^*=s_j$，策略组合 $s^*=(s_i^*,s_j^*)$ 是不完美信息无限次重复博弈 $\Gamma$ 的序贯均衡策略。

**证明** 对于策略$s^*$，任何一个节点$i(j)$单独偏离策略$s_i^*(s_j^*)$其总收益不会增加，因此$s^*$是博弈$\Gamma$的纳什均衡策略。因为在博弈的任何阶段，任意一个博弈者$i(j)$偏离策略$s_i^*(s_j^*)$其总收益不会增加，因此$s^*$在没有到达的信息集上也是纳什均衡策略。由引理 1 及定理 1 可知，$s^*$是无限次重复博弈$\Gamma$的序贯均衡策略。 证毕

图 3 是$m^k$随$k$的变化趋势，参数$l=3$, $g=2$, $\delta=0.99$, $\lambda=0.03$。从图 3 中可以看出，随着连续观察到对方丢包次数的增加($k$的增加)，节点选择转发行为的概率($m^k$)逐渐降低，最终趋于稳定值$m$。从式(8)可以看出，博弈者$i$的对手收益也是随着自身丢包行为次数增加而减少。博弈者$i$也以较低的概率$m$选择转发行为$F$。由于博弈对称特性，博弈者$i$丢包行为次数将随之增加，也会导致博弈者$i$的收益减少。最后双方将会在一个较低的转发行为概率值$m^k$=$m$范围内达到平衡。但不会出现$m^k$=0 的情形。若对方始终选择丢包行为，博弈者$i$以较低的概率$m$选择转发行为$F$，降低了节点的损失。若对方为正常节点，由于信息的不完美性导致节点(连续)观察到丢包行为，博弈者$i$以较大概率$m^k$试图与对方恢复合作，提高了网络的合作率。

## 4 仿真

在仿真中，本文给出基于贝尔曼方程的序贯均衡策略，与文献[15]中的序贯均衡策略进行比较。对

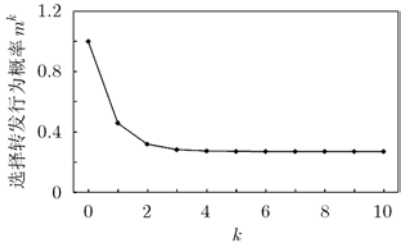



图3 $m^k$ 与 $k$ 的关系

于文献[15]中的模型，为了获得更好的收益，文献[15]将重复博弈 $\Gamma$ 分割为 $N$ 个重复博弈 $\Gamma_{NG}$。

$$N=\left\lfloor \lg\delta' / \lg\delta \right\rfloor \tag{9}$$

其中 $\delta'$ 为文献中模型对贴现因子的最低要求。第 $k(0\le k<N)$ 个重复博弈 $\Gamma_{NG}$ 发生在原博弈的如下阶段：$k$, $k+N$, $k+2N$, $\cdots$。通过分析可知，对于本文中的收益矩阵，$N$ 与 $l/(g+1)$成正比。对于参数 $g=2$, $l=2$, $N\approx 0.34/(1-\delta)$，其中 $1/(1-\delta)$ 是博弈 $\Gamma$ 的期望重复次数。因此，对于文献给出的序贯均衡策略，为了避免因使用简单触发机制造成的性能损失，文献[15]将博弈 $\Gamma$ 分割成多个重复博弈。这不仅增加了机制的复杂性，消耗节点更多的内存资源，而且性能提升有限。

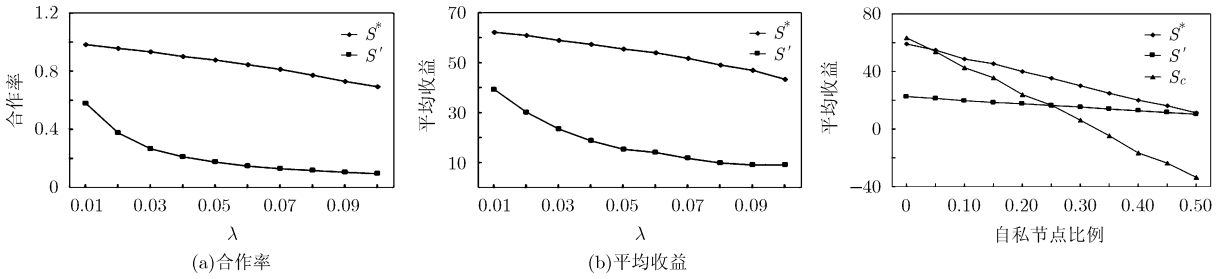
从合作率、平均收益和对不同网络环境的适应性 3 个方面考察两种序贯均衡策略的性能。仿真模拟了 4 种类型的策略，$S^*$为本文给出的序贯均衡策略；$S'$为文献[15]中的策略；$S_c$为合作策略，节点始终选择转发行为$F$; $S_d$为背叛策略，节点始终选择丢包行为$D$。

图 4 为分别使用策略$S^*$和$S'$时，网络的合作率、平均收益随观察误差的变化曲线。实验参数为：$g$=2, $l=2$, $\delta$=0.99。对于策略$S'$，作者使用简单的触发机制构造该策略，触发机制对观察误差非常敏感。因此，随着观察误差的增加，网络的合作率、节点的平均收益迅速降低。对于策略$S^*$，当观察出现误差时，节点以一定的概率恢复合作，提高了网络的合作率和节点的平均收益。仿真结果表明，网络采用策略$S^*$时，合作率、平均收益与观察误差呈现出近似的线性关系。随着观察误差的增加，合作率、平均收益没有显著下降。

图 5 反映了在不同的网络环境中，使用策略$S^*$, $S'$和$S_c$时节点的平均收益。实验参数为：$g=2$, $l=2$, $\delta=0.99$, $\lambda=0.03$。对于策略$S_c$，节点始终选择转发行为$F$，随着网络中自私节点比例的增加，节点的收益快速下降。策略$S^*$在面对自私节点时，以概率$m$选择转发行为$F$，一定程度上减少了损失。策略$S'$使用了触发机制，如果观察到自私节点的丢包行为$d$，那么节点在之后的博弈中始终选择不合作行为$D$。因此策略$S'$收到的影响最低。

## 5 结论

一种有效的合作激励机制可以防止节点自私行为的发生。本文应用不完美信息重复博弈模型分析 Ad hoc 网络中节点的交互，使用贝尔曼方程构造满足序贯均衡的合作激励机制。与已有的序贯均衡相比，本文给出的序贯均衡策略避免使用触发机制，提高在不完美信息环境下的性能。仿真结果表明，本文给出的序贯均衡策略不仅提高了网络的合作率和节点的平均收益，而且提高了机制的适应性。




(a)合作率　　(b)平均收益

图 4 合作率和平均收益与参数 $\lambda$ 的关系　　图 5 节点的平均收益随自私节点所占比例的变化

## 参 考 文 献


[1] Marti S, Giuli T J, Lai K, *et al*.. Mitigating routing misbehavior in mobile ad hoc networks[C]. Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCom'00), Boston, MA, 2000: 255–265.

[2] Mahmoud M M E A and Shen X. FESCIM: fair, efficient, and secure cooperation incentive mechanism for multihop cellular networks[J]. *IEEE Transactions on Mobile Computing,* 2012, 11(5): 753–766.

[3] Zhong S, Chen J, and Yang Y R. Sprite: a simple, cheat-proof, credit based system for mobile ad-hoc networks[C]. Proceedings of the IEEE INFOCOM'03, San Francisco, CA, 2003: 1987–1997.

[4] Cutillo L A, Molva R, and Önen M. PRICE: privacy preserving incentives for cooperation enforcement[C]. World of Wireless, Mobile and Multimedia Networks (WoWMoM), San Francisco, California, USA, 2012: 1–9.

[5] Kaushik R and Singhai J. Modspirite: a credit based solution to enforce node cooperation in an ad-hoc network[J]. *International Journal of Computer Science Issues*, 2011, 8(2/3): 295–302.

[6] Mahmoud M E and Shen X M. Credit-based mechanism protecting multi-hop wireless networks from rational and irrational packet drop[C]. Global Telecommunications Conference, Waterloo, Canada, 2010: 1–5.

[7] Saad W, Han Z, Debbah M, *et al*.. A distributed coalition formation framework for fair user cooperation in wireless networks[J]. *IEEE Transactions on Wireless Communications*, 2009, 8(9): 4580–4593.

[8] Milan F, Jaramillo J J, and Srikant R. Achieving cooperation in multihop wireless networks of selfish nodes[C]. Workshop on Game Theory for Networks (GameNets 2006), Pisa, Italy, 2006: 3–3

[9] Michiardi P and Molva R. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks[C]. Proceedings of the Communications and Multimedia Security Conference (CMS ' 02), Portoroz, Slovenia, 2002: 107–121.

[10] Buchegger S and Le Boundec J Y. Performance analysis of the CONFIDANT protocol (cooperation of nodes: fairness in dynamic ad-hoc networks)[C]. Proceedings of the International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc'02), Lausanne, Switzerland, 2002: 226–236.

[11] 许智君, 胡琪, 张玉军, 等. MANET 网络激励节点协作的信任评估路由协议[J]. 通信学报, 2012, 33(7): 27–35.

[12] Charilas D E, Georgilakis K D, and Panagopoulos A D. ICARUS: hybrid incentive mechanism for cooperation stimulation in ad hoc networks[J]. *Ad Hoc Networks*, 2012, 10(6): 976–989.

[13] Yu K and Chen X B. Analysis of reputation-based incentive mechanisms in Ad hoc networks[C]. Internet Conference on Software Engineering and Service Science (ICSESS). Beijing, China, 2012: 333–336.

[14] Jaramillo J J and Srikant R. A game theory based reputation mechanism to incentivize cooperation in wireless Ad hoc networks[J]. *Ad Hoc Networks*, 2010, 8: 416–429.

[15] Ji Z, Yu W, and Liu K J R. A belief evaluation framework in autonomous MANETs under noisy and imperfect observation: vulnerability analysis and cooperation enforcement[J]. *IEEE Transactions on Mobile Computing*, 2010, 9(9): 1242–1254.

[16] Kreps D M and Wilson R. Sequential equilibria[J]. *Econometrica*: *Journal of the Econometric Society*, 1982, 50(4): 863–894.

[17] Osborne M J. A course in game theory[M]. Cambridge, Mass.: MIT Press, 1994: 1–353.

[18] Kandori M and Matsushima H. Private observation, communication and collusion[J]. *Econometrica*, 1998, 66(3): 627–652.



张华鹏：　男，1988 年生，硕士，研究方向为无线传感器网络.

张宏斌：　男，1967 年生，博士，副教授，研究方向为无线传感器网络、复杂网络及仿真.

International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.6 (2014), pp.49-58
http://dx.doi.org/10.14257/ijmue.2014.9.6.06

# Modeling and Verifying of CPS Component Services Based on Hybrid Automata

Jianning Zhang[1], Guanquan Zhang[1,2]*,Rongjie Yan[2],Yi Zhu[3] and Xingjun Qi[1]

[1]*School of Computer Science & Technology, Soochow University, Suzhou, 215006, China*
[2]*State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Science, Beijing, 100190, China*
[3]*School of Computer Science and Technology, Jiangsu Normal University, Xuzhou, 221116, China*

**gqzhang@suda.edu.cn*

***Abstract***

*In recent years, the modeling and verifying of Cyber-Physical System (CPS) is now an important aspect of CPS researches. Because of the CPS' complex architecture, it may suffer from the state-space explosion problem when we verify CPS models by model checking methods. Therefore, we offer a method which models CPS with Component Services. The method treats the CPS components as a service provider, and models component services to further simplify the system's state-space. We verify the correctness of this model and solve the synchronous/asynchronous communication problems.*

***Keywords:*** *Cyber-physical System; component services; state-space explosion; model checking*

## 1. Introduction

Cyber-Physical Systems (CPSs) are integrations of the computation and physical processes. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa [1]. At present, applications of CPSs mainly include high confidence medical devices and systems, traffic control and safety, critical infrastructure control, advanced automotive and energy conservation [2-5]. Most of these systems are resource-constrained, and have a high requirement for real-time response and fault tolerance. The work in [6] names it as Performance Critical Systems (PCSs), and therefore the goal of this paper is to ensure the performance of CPS.

In model-based design (MBD) and model driven development (MDD), models play an important role in the design process, so we can analyze and verify the system's properties in the early time. The MBD and MDD methods can ensure the quality of the system, and efficiently reduce the time and cost of the system development. However, the quality of system is determined by the architecture of system [7], and hence applying the design and developing methods of the Model-based Architecture-driven to the CPS modeling can efficiently guarantee the system performance. Because it can automatically prove properties of systems, the Model Checking technology is widely used in the MBD and MDD. As the increase of the number of states of the system, the model checking method will suffer from the so called state-space exploration problem, which restricts its application a lot.

ISSN: 1975-0080 IJMUE

Compositional verification techniques are used to cope with this problem in concurrent systems. The idea is to apply divide-and-conquer approaches to infer global properties of complex systems from properties of their components, and separate verification of components limits state explosion [8].

Since CPS is a kind of distributed embedded systems, and its components distributed in different physical environments. If we model all system components, the architecture of system will be too large to analyze, therefore, we combine compositional verification techniques with service-oriented architecture and propose a definition of "Component-Service", which means the components of CPS are the service provider and register the components in system according to the its service model, and the "request / response" operation mode is adopted. As a part of component service model, the physical environment of components is used to describe constrains of the service, while for the upper system it used for being shield from the physical environment, and thus can simplify the architecture of the system.

In Section 2, we introduce the related works. In Section 3, we give the concepts that will be used through the rest of this paper and model CPS component service based on hybrid automata. In Section 4, we demonstrate the CPS component service with an intelligent traffic control system. Finally, Section 5 makes a conclusion for the present study, and also gives some suggestions for the further studies.

## 2. Related Work

In recent years, many scholars have paid close attention to the research of the Service-oriented Architecture (SOA) and the component-based architecture. The work in [9] proposes the Service Component Architecture Specification to describe the system's model with Service-Oriented Architecture. The work in [10] proposes leverage existing and emerging standards from both the embedded-device and IT domains within a Service-Oriented Device Architecture (SODA) to eliminate much of the complexity and cost associated with integrating devices into highly distributed enterprise systems. Based on DPWS (Device Profile for Web Service), the work in [11] proposes a concept of Service Gateway, which serves as a middleware used for communication and translation between Web services and embedded systems. The work in [12] proposes a framework to analyze and verify the compositional properties of the Web service. In this framework, the BPEL process is used to describe Web service, transformed to automata model and described with Promela language so it can be verified by SPIN. The work in [13] proposes a CP-nets-based design and verification framework for Web services composition, which is used to create and verify the BPEL process. These researches mainly focus on the creation of integrated framework or the description of Web services, and have no concern about the physical world while the physical world is an important aspect of CPS.

The work in [14] proposes models and their relationship to realizations of CPSs. However, the models they design are not structure models but primarily about dynamics, the evolution of a system's state in time, so they can not represent static information about the construction of a system. The work in [15] proposes a compositional method for the verification of component-based systems described in a subset of the BIP language encompassing multi-party interactions. However, this method doesn't consider the physical environment which is an important part of CPS.

Therefore, based on these researches, we introduce a component service model which can reduce the complexity of CPS model.

## 3. CPS Component Service Model

In this section, we present a basic model of the CPS component service. The CPS combines the communication between discrete and continuous processes, and the hybrid automata includes discrete and continuous state, so we can use it to model CPS component services. Here's the brief introduction about the hybrid automata.

Definition 1 [Hybrid Automata] A hybrid automata HA[16] is defined by $HA = (Q, X, Init, f, Inv, Jump)$ where:

- Q is a set of finite discrete states, which uses to describe the cyber properties;
- X is a set of finite continuous states, which uses to describe the physical properties;
- *Init* is a set of initial state and $Init \subset Q \times X$ ;
- $f : Q \times X \to X$ is a set of continuous dynamic functions;
- $Inv : Q \to 2^X$ is an invariant whose free variables are from $q \in Q$;
- $Jump : Q \times X \to 2^{Q \times X}$ is a set of jump functions.

Since the CPS components are mostly distributed at natural environments, and usually are resource-constrained, so the services offered by CPS components are mostly atomic services. An atomic service is a kind of service which can fulfill a function but the service itself cannot be further divided into two or more services. Every atomic service has the following properties:

1) an unique id to be differentiated from other services;
2) a set of variables which contain discrete and continuous state;
3) a set of functions such as numerical calculation, devices control, etc.;
4) a set of ports which can be used to communicate with other services.

Therefore, the formal definition of the atomic service as shown below:

Definition 2 [Atomic Service] An atomic service can be defined as $S =< Sid, port, HA >$, where $Sid$ is the service id, *port* is a set of service port which is used to communication with other services, $HA = (Q, X, Init, f, Inv, Jump)$ is a hybrid automata which is used to describe properties and behavior of CPS component services.

Example 1. Figure 1 presents a simplified temperature-control system using CPS component services model. There are only 2 services in this model: a temperature-sensing service $S_1$ and a temperature-changing service $S_2$. $S_1$ has two states: low-temperature $l_1$ and high-temperature $l_2$, and two ports: $p_1$ and $q_1$. $S_2$ has two states: open-state $l_3$ and close-state $l_4$, and two ports: $p_2$ and $q_2$. Compared with the classical component-based system model which needs a lot of sensors and actuators, the CPS component service model is much more simplified.

Based on the definition 2, we can define the CPS component. A CPS component is a set of services, and the following are the formal definition.

Definition 3 [connector] Given a set of CPS atomic services $S_1, S_2, \cdots, S_n$, a connector is defined by $Con \subseteq \bigcup_{i=1}^{n} S_i.port$ .where $S_i.port$ is the port of $S_i$, and for $\forall i = 1, \cdots, n$, we have $|Con \cap S_i.port| \leq 1$, i.e., each connector has at most one port per service.

In Figure 1, as an example, the set $\{p_1, p_2\}$ is a connector between $S_1$ and $S_2$. This connector describes a synchronization between different services by ports $p_1$ and $p_2$.

Lemma 1. Let $f(s_i, p_i) = s_j$, $f(s_i, p_j) = s_t$, if $p_i = p_j$, then $s_j = s_t$, i.e., the transfer function is determined by the transfer function f.

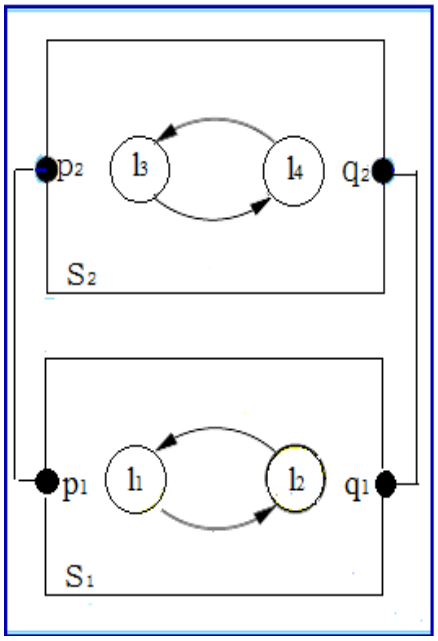



**Figure 1. A Simplified Temperature-control System**

Definition 4 [executable procedure] Given an atomic service $S =< Sid, port, HA >$, and $s_i, i = 1, 2, \cdots, n$ is a set of states of $S$, $p_i, i = 1, 2, \cdots, n$ is a set of ports of $S$, an executable procedure is defined by $\lambda(S) = s_1 p_1 s_2 p_2 \cdots s_n p_n$, where $s_{i-1} \xrightarrow{p_i} s_i, i = 1, 2, \cdots, n$.

Definition 5 [reachable state set] Given an atomic service $S =< Sid, port, HA >$, a reachable state set of $S$ is defined by $rss(S) = \{s_i \mid s_0 \xrightarrow{\lambda} s_i, i = 1, 2, \cdots, n\}$ , where $s_0$ is the initial state of $S$ , and $\lambda = s_1 p_1 s_2 p_2 \cdots s_n p_n$ is an executable procedure of $S$ .

Definition 6 [correctness] Given an atomic service $S =< Sid, port, HA >$, if for every $s_i \in S$ , $s_i \in rss(S)$ , we say the service $S$ is correct.

The correctness of a service means that we can get a state of the service after a list of executable procedures.

Definition 7 [composite service] A composite service is defined by $CS = \gamma(S_1, S_2, \cdots, S_n)$, where

- $CS.port = \bigcup_{i=1}^{n} Connector(S_i)$, i.e., the compositional service 'ports is a union of all services 'ports.
- $CS.Init = \bigcap_{i=1}^{n} S_i.Init$, i.e., the initial state of compositional service is intersection of the initial state of all services.
- $CS.Q = \sum_{i=1}^{n} S_i.Q$ is a set of cyber properties of all services, and $CS.X = \sum_{i=1}^{n} S_i.X$ is a set of physical properties of all services.
- $CS.f = \sum_{i=1}^{n} S_i.f$ is a transfer function of $CS$.

Lemma 2. Let $\gamma = \bigcup_{i=1}^{n} S_i.port$ be a connector of a composite service $CS = \gamma(S_1, S_2, \cdots, S_n)$, for $\forall i \in n, S_i \in CS \wedge S_i.port \notin \gamma$, the states of $S_i$ keep constant, i.e., the communication among some services don't influence others.

Theorem 1. Given a composite service $CS = \gamma(S_1, S_2, \cdots, S_n)$, the transfer function $CS.f$ is unique iff (1) for $\forall p_i \in S_i$, if $CS.f(s_i, p_i) = s_j$, we have $S_i.f(s_i, p_i) = s_j$, and (2) the states which are not in $CS.f$ keep constant.

Proof. ($\rightarrow$) Suppose $\exists p_i \in S_i$ so that $CS.f(s_i, p_i) = s_j$ and $S_i.f(s_i, p_j) \neq s_j$. Just let $S_i.f(s_i, p_i) = s_k$, where $s_i, s_j, s_k$ are states of $S_i$. Depending on the Definition 7, we can know that $S_i.f \subset CS.f$, which is contradictory to that $CS.f$ is unique. So we have $S_i.f(s_i, p_j) = s_j$, and based on Lemma 2, we know that the states which are not in $CS.f$ keep constant.

($\leftarrow$) Suppose there are 2 transfer functions $CS.f_1, CS.f_2$, then for $\forall s_i, s_j, p_i$, if $CS.f_1(s_i, p_i) = s_j$, we have $S_i.f(s_i, p_i) = s_j$, and the states which are not in $CS.f$ keep constant, so we can know that $CS.f_2(s_i, p_i) = s_j$ which is contradictory to the Lemma 1. So the transfer function is unique.

Lemma 3. Given a composite service $CS = \gamma(S_1, S_2, \cdots, S_n)$, if the transfer function $CS.f$ is unique, the service model is correct.

Definition 8 [system] A system can be defined as $Sys =< CS, Init >$, where $CS$ is defined by Definition 7, and $Init$ is an initial state of the system.

As the temperature-control system shown in figure 1, $Sset =< S1, S2 >$ and $Init = l_1 \wedge l_3$ *i.e.*, at initial state, the temperature-control system is at low-temperature state and the temperature-changing devices are also closed.

## 4. Case Study

In this chapter, we will use a smart-traffic system as an example, and specify the application of CPS component service model which is defined previously.

For the purpose of simplicity, we suppose that there are only five different kinds of services: the GPS service can get location information of every components repeatedly over a time period; the traffic lights service can control the vehicles' state; the parking service can show the information of the nearest car park; the information service can send the real-time road's state to system's control center; and the warning service can send unpredictable circumstances to the system.

$GPS\_Service =< GSid, port1, (period, longitude, latitude) >$, where GSid is an id of this service, port1 is the only port of this service, there are three parameters in $GPS\_Service$: $period$ is the period of $GPS\_Service$ and $longitude, latitude$ is the location information of $GPS\_Service$. In the initial state, $port1 = off$, and when $time = period$, we have $port1 = on$.

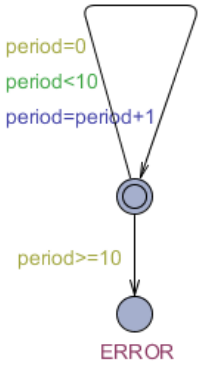



**Figure 2. Hybrid Automata Model for GPS Service**

The traffic lights service $L\_S =< LSid, port1, (per, red, yel, blue, f) >$, where $LSid$ is an id of this service, $port1$ is the only port of this service, there are four parameters in $L\_S$: per is the period of L_S, red, yellow, blue are light information, and $f : red \times period \to blue, blue \times period \to yel, yel \times period \to red$ is a transition function of $L\_S$.

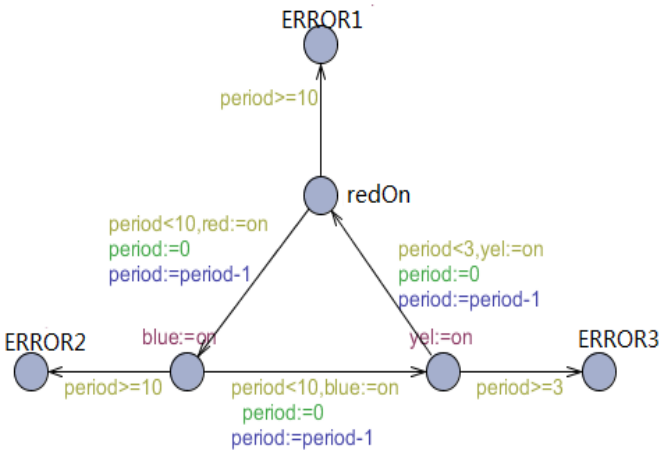



**Figure 3. Hybrid Automata Model for L_S**

Parking service $P\_S =< PSid, port1, (total, rest, f) >$, where $PSid$ is an id of $P\_S$, $port1$ is the port of this service, there are two parameters in $P\_S$: $total$ is the parking spaces and $rest$ is the rest parking spaces of the car parking, $f : rest \to rest+1, rest \to rest-1$ is the transition function.

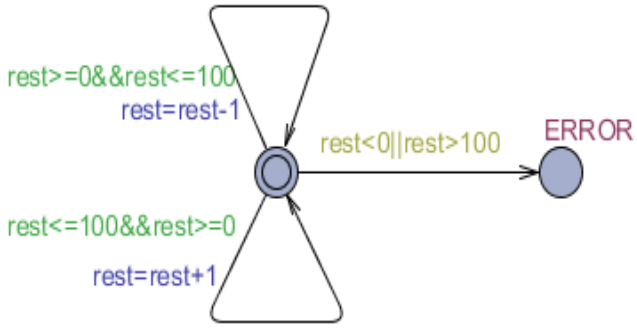



**Figure 4. Hybrid Automata Model for P_S**

Information service $I\_S =< ISid, port, (stat, isbusy, wea) >$, where $ISid$ is the id of this service, $port$ is the port of this service, there are three parameters in $I\_S$: $stat$ presents the real-time state of roads, $isbusy$ is a Boolean value which is used to check whether the road is blocked, $wea$ is used to offer weather information.

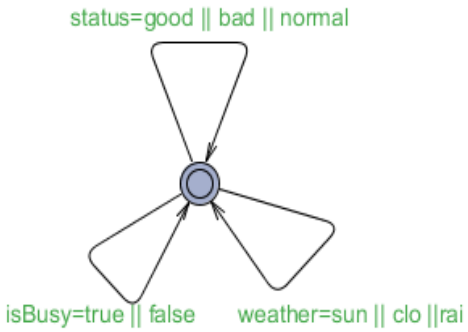



**Figure 5. Hybrid Automata Model for I_S**

The warning service $W\_S = <WSid, port1, (warning, guard)>$, where $WSid$ is an id of this service, $port1$ is the only port of this service, there are two parameters in $W\_S$: warning and guard. In the initial state, $port1 = off, warning = false$ while $guard = true$, $port1 = on, warning = true$.

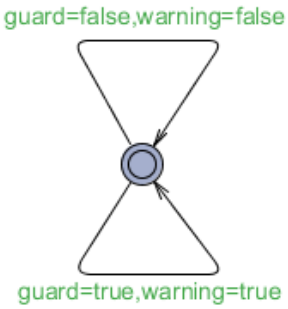



**Figure 6. Hybrid Automata Model for W_S**

## 5. Conclusion and Future Work

In this paper, we propose a model of CPS based on component services, verify the correctness of the model, and ensure the synchronous/asynchronous communication among services by connectors. We also give an example to demonstrate the application of CPS component service model. In the future, we will give a further validation on deadlock free models.

## Acknowledgements

The work of this paper is supported by the Natural Science Foundation of Jiangsu, China (Grant No. BK2011281) and the Applied Fundamental Research Program of Suzhou, China (Grant No. SYG201241).

## References

[1] E. A. Lee, "Cyber physical systems, Design challenges", Proceedings of ISORC, Piscataway, NJ, IEEE, (**2008**), pp. 363-369

[2] J. Hatcliff, A. King, A. MacDonald, A. Fernando, M. Robkin, E. Vasserman, S. Wininger, and J. M. Golderman, "Rationale and Architecture Principles for Medical Application Platforms", Proceedings of the 3rd IEEE/ACM ICCPS,(**2012**) April 17-19, Beijing, China.
[3] P. Park and C. Tomlin, "Investigating Communication Infrastructure of Next Generation Air Traffic Management", Proceedings of the 3rd IEEE/ACM ICCPS, (**2012**) April 17-19, Beijing, China.
[4] J. Kim, K. Lakshman, R. Rajkumar and R. Tasks, "A New Task Model with Continually Varying Periods for Cyber-Physical Systems", Proceeding of the 3rd IEEE/ACM ICCPS, (**2012**) April 17-19, Beijing, China.
[5] C. L. Fok, M. Hanna, S. Gee, T. C. Au, P. Stone, C. Julien and Sriram Wishvanashi, "A Platform for Evaluating Autonomous Intersection Management Policies", Proceedings of the 3rd IEEE/ACM ICCPS, (**2012**) April 17-19, Beijing, China.
[6] P. H. Feiler, B. A. Lewis and S. Vestal, "The SAE architecture analysis and design language (AADL)– A standard for engineering performance critical systems", Proceedings of IEEE Computer Aided Control Systems Design, (**2006**) October 4-6, Munich, Germany.
[7] C. Atkinson and T. Kuhne, "Model-driven development: A metamodeling foundation", IEEE Software, vol. 20, no. 5, (**2003**).
[8] M. Beisiegel, H. Blohm and D. Booz, "Service Component Architecture (SCA), version 1.0. Billerica: Organization for the Advancement of Structured Information Standards (OASIS)", (**2007).**
[9] S. Deugd, K. Kelly, B. Market and J. Ricker, "SODA: Service oriented device architecture", IEEE Pervasive Computing, vol. 5, no. 3, (**2006**).
[10] C. Buckl and S. Sommer, "Generating a Tailored Middleware for Wireless Sensor Network Application", Proceedings of the IEEE SUTC, (**2008**) June 11-13, Taichung, Taiwan.
[11] L. Souza, P. Spiess, D. Guinard, M. Kohler, S. Karnoskov and D. Savio, "SOCRADES: A Web service based shop floor integration infrastructure", Proceedings of the Internet of Things, (**2008**) March 26-28, Zurich, Switzerland.
[12] C. Hein, T.Ritter and M.Wagner, "Mode l-Driven tool integration with Model Bus", In Workshop Future Trends of Model-Driven Development, (**2009**).
[13] J. Huang, "A SOA Model of Cyber Physical Systems", https://utd.edu/~i lyencourse/service/project/jian.pdf.
[14] K. Bae, P. C. Olverzky, T. H. Feng and E. A. Lee, "Verifying hierarchical Ptolemy II discrete-event models using Real-Time Maude", Science of Computer Programming, vol. 77, (**2012**).
[15] S. Bensalem, M. Bozga, T.H. Nguyen and J. Sifakis, "Compositional verification for component-based systems and application", IET Software, vol. 4, no. 3, (**2010**).
[16] H. A. Henzinger, "The Theory of Hybrid Automata", Proceedings of LICS, (**1996**) July 27-30, New Jersey, USA.

2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# A Supervised Neighborhood Preserving Embedding for Face Recognition

Xing Bao
School of Computer Science and Technology& Provincial key Laboratory for Computer information processing technology Soochow University Suzhou 215006, Jiangsu, China
20124227034@suda.edu.cn

Li Zhang
School of Computer Science and Technology& Provincial key Laboratory for Computer information processing technology Soochow University Suzhou 215006, Jiangsu, China
zhangliml@suda.edu.cn

Bangjun Wang
School of Computer Science and Technology& Provincial key Laboratory for Computer information processing technology Soochow University Suzhou 215006, Jiangsu, China
wangbangjun @suda.edu.cn

Jiwen Yang
School of Computer Science and Technology& Provincial key Laboratory for Computer information processing technology Soochow University Suzhou 215006, Jiangsu, China
jwyang @suda.edu.cn

***Abstract*—Neighborhood preserving embedding (NPE) is an approximation to locally linear embedding (LLE), which has an ability to preserve local neighborhood structure on data manifold. As an unsupervised dimensionality reduction method, NPE can be applied to face recognition for preprocessing. However, NPE could not utilize the label information in the classification tasks. To make the data in a reduced subspace separable, this paper proposes a supervised neighborhood preserving embedding which could learn a projection matrix by using both the geometrical manifold structure and the label information of the given data. In addition, the projection matrix could be found by solving a linear set of equations. Experimental results on ORL and Yale face image datasets show that the proposed method has a high recognition rate.**

***Keywords—face recognition; dimension reduction; label information; local preserving embedding***

## I. INTRODUCTION

In the real world, data exists in the form of high-dimensionality, including image data. Dealing with high-dimensional data directly would cause a large computational complexity, the curse of dimensionality and other problems [1]. It is an effective way to overcome the problems caused by high-dimensional data by projecting the high-dimensional data into low-dimensional subspaces. Therefore, dimensionality reduction plays an important role in their specific applications, including data visualization, data compression [2], pattern recognition and classification [3], multimedia information retrieval and others.

The two classical linear embedding methods are linear discriminant analysis (LDA) [4] and principal component analysis (PCA) [5]–[6], which are demonstrated to be computationally efficient and suitable for practical applications. LDA is a supervised dimensionality reduction algorithm. This algorithm aims to find the optimal projection vector on which the data points of different classes are far from each other and the data points of the same class are to be as close to each other as possible. PCA, an unsupervised method, is famous for the low-dimensional representation of high-dimensional data. In other words, LDA tries to find the expected projection direction of the data for classification tasks, while PCA seeks for an effective way to represent data for compressing data.

Manifold learning is a typical nonlinear dimensionality reduction method. Usually, manifold learning is first to construct a data adjacency graph to characterize the data distribution or geometry, and then seek for an optimal mapping or a projection direction to effectively maintain the structure. Most manifold learning algorithms, such as laplacian eigenmaps (LE) [7], locally linear embedding (LLE) [8], locality preserving projection (LPP) [9] and neighborhood preserving embedding (NPE) [10], even including the classic PCA and LDA which can be unified under the framework of adjacency graph construction and dimensionality reduction.LPP is a linear approximation to LE. The goal of LPP is to project the high-dimensional data into a low-dimensional manifold subspace that can better preserve the original data's locality. In other words, the adjacent data points in the original data space can also maintain adjacent relationship respectively in the projection subspace.

NPE is a linear approximation to LLE and has an ability to maintain the local neighborhood information of data manifold. NPE has received extensive attention in face recognition [11-19]. However, in face recognition tasks, NPE is used as an unsupervised dimensionality reduction method, and cannot take advantage of the label information on given data. To utilize the label information, Du et al proposed a new subspace learning method called neighborhood preserving discriminant embedding (NPDE) [20]. NPDE keeps the data information in the local neighborhood manifold structure while emphasizing the discrimination information of high-dimensional data. It can ensure the minimum reconstruction error of local neighborhood and make the projection sample points with minimum within-class scatter and maximum between-scatter. Unlike many existing techniques such as LPP and NPE, in which the local neighborhood information is preserved during the dimension reduction procedure, sparsity preserving projection (SPP) [21] aims to preserve the sparse reconstructive relationship of the data, which is achieved by

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

minimizing a $l_1$ regularization-related objective function. It is well known that maximum margin criterion (MMC) [23] is a method proposed to maximize the trace of the difference of the between-class scatter matrix and within-class scatter matrix. Thus, discriminant sparse neighborhood preserving embedding (DSNPE) [22] was proposed by introducing MMC into the objective function of SPP, which has two advantages: (1) it retains the sparsity characteristic of SPP; (2) it emphasizes the discriminative information by incorporating MMC, which can make the class mean vectors have a wide spread and make every class scatter in a small space. Furthermore, to further increase the discriminative power of DSNPE, it integrates additional discriminant information.

This paper proposes a novel supervised neighborhood preserving embedding (SNPE). Different from NPDE, SNPE utilizes the label information to construct attraction vectors each of which would attract points in the same class. Meanwhile, SNPE requires preserving the local neighborhood structure on data manifold. By doing so, the embedded points in the same class would be close to each other, while the points in the different classes would be far away from each other. In addition, SNPE is cast into a linear set of equations, which is easier to solve. .

The rest of this paper is organized as follows. In Section 2, we briefly review NPE and NPDE. Section 3 proposes SNPE. In Section 4, we compare SNPE with some related works and give experimental results. Conclusions are made in Section 6.

## II. RELATED WORK

### A. Neighborhood Preserving Embedding

Let the training samples matrix be $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_n]$, where $\mathbf{x}_i \in R^m$, $m$ is the dimensionality of the training samples and $n$ is the number of the training samples. NPE is intended to reduce the dimensionality of data and maintain the inherent local neighbor manifold structure at the same time. It seeks for an optimal transformation matrix $\mathbf{A}=[\mathbf{a}_1,\mathbf{a}_2,...,\mathbf{a}_m]$, where $\mathbf{a}_i \in R^d$, which can map the high-dimensional data into a relatively low-dimensional feature subspace.

Similar to LLE, NPE evaluates the affinity weight matrix by using local least squares approximation. The local approximation error in NPE is measured by minimizing the following cost function:

$$\varphi(\mathbf{W})=\sum_{i=1}^{n}\left\|\mathbf{x}_i-\sum_{j=1}^{n}\mathbf{W}_{ij}\mathbf{x}_j\right\|^2 \tag{1}$$

under two constraints: (1) if $\mathbf{x}_j$ is not one of $k$ neighbors of $\mathbf{x}_i$, then $\mathbf{W}_{ij}=0$; otherwise, $\mathbf{W}_{ij}\neq 0$ and $\sum_{j=1}^{n}\mathbf{W}_{ij}=1$, $j=1,2,...,n$. A reasonable criterion for choosing a "good" projection is minimizing the cost function:

$$\begin{aligned}\mathbf{A}&=\arg\min_{\mathbf{A}}\left[\sum_{i=1}^{n}\left\|\mathbf{y}_i-\sum_{j=1}^{n}\mathbf{W}_{ij}\mathbf{y}_j\right\|^2\right] \\ &=\arg\min_{\mathbf{A}} tr(\mathbf{A}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{A})\end{aligned} \tag{2}$$

which subject to $\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A}=\mathbf{I}$, where $tr(\bullet)$ is the trace of matrix $\bullet$, $\mathbf{I}$ is the identity matrix, $\mathbf{Y}=\mathbf{A}\mathbf{X}$ and $\mathbf{M}=(\mathbf{I}-\mathbf{W})^T(\mathbf{I}-\mathbf{W})$ is a sparse, symmetric, and semi-positive definite matrix.

By using Lagrange multiple technique, NPE leads to the following generalized eigenvector problem:

$$\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{A}=\lambda\mathbf{X}\mathbf{X}^T\mathbf{A} \tag{3}$$

### B. Neighborhood Preserving Discriminant Embedding

NPE is an unsupervised learning method, and it could not utilize the label information on given data in classification tasks. Therefore, NPDE was proposed in [20]. NPDE keeps the local neighborhood structure on data manifold and simultaneously emphasizes the discrimination information of data. It can make the local neighborhood reconstruction error minimal, and maintain points with minimum within-class scatter and maximum between-scatter in the subspace. Similarly, both of two methods involve solving the characteristic of the matrix decomposition problem.

Let $\mathbf{X}=[\mathbf{X}_1,\mathbf{X}_2,...,\mathbf{X}_C]$ represent the training samples matrix in original subspace, where $\mathbf{X}_c$ denotes the sample matrix belonging to the $c$th class, and $C$ is the total number of classes. Let $n_c$ be the number of the training samples in the $c$th class. So, the total number of the training samples is $n=\sum_{c=1}^{C}n_c$. Let $\mathbf{Y}=[\mathbf{Y}_1,\mathbf{Y}_2,...,\mathbf{Y}_C]^T$ be the projected training sample matrix in the low dimensional subspace, we have the objective function of NPDE:

$$\min\quad J=\frac{\sum_{c=1}^{C}\sum_{i=1}^{n_c}[\mathbf{y}_i^c-\sum_{j=1}^{n_c}\mathbf{W}_{ij}^c\mathbf{y}_j^c]}{\sum_{c=1}^{C}n_c(\mathbf{u}_c-\mathbf{u})(\mathbf{u}_c-\mathbf{u})^T} \tag{4}$$

where $\mathbf{y}_i^c$ and $\mathbf{y}_j^c$ respectively denote the $i$th and $j$th embedded vectors in the $c$th class, $\mathbf{W}_{ij}^c$ denotes the reconstruction weighting coefficient of training samples in the $c$th class, $\mathbf{u}_c$ denotes the mean of embedded vectors in the $c$th class and $\mathbf{u}$ denotes the mean of all embedded vectors.

The objective function of NPDE can be reduced to (5)

$$\min\quad J=\frac{\boldsymbol{\alpha}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T\mathbf{S}_B\boldsymbol{\alpha}} \tag{5}$$

where $\mathbf{S}_B = \sum_{c=1}^{C} n_c(\mathbf{m}_c - \mathbf{m})(\mathbf{m}_c - \mathbf{m})^T$ denotes the between-class scatter matrix in the original space, $\mathbf{m}_c = \frac{1}{n_c}\sum_{i=1}^{n_c}\mathbf{x}_i^c$, $\mathbf{m} = \frac{1}{n}\sum_{c=1}^{C}\sum_{i=1}^{n_c}\mathbf{x}_i^c$, the matrix $\mathbf{M}$ is

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & 0 & \dots & 0 \\ 0 & \mathbf{M}_2 & \dots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \dots & \mathbf{M}_C \end{bmatrix}$$

with $\mathbf{M}_c = (\mathbf{I} - \mathbf{W}_c)^T(\mathbf{I} - \mathbf{W}_c)$, $\mathbf{W}_c \in R^{n_c \times n}$ whose ith row and jth column is $\mathbf{W}_{ij}^c$.In addition, the rank of $\mathbf{S}_B$ is $C-1$.

The optimal transformation matrix can be obtained by minimizing the objective function of (5). Minimizing the objective function of (5) is equivalent to minimize the numerator term and maximize the denominator term of the objective function simultaneously.

## III. Supervised Neighborhood Preserving Embedding

In order to incorporate the label information on the given data, this paper proposes an alternative supervised NPE method, called SNPE. In our method, we construct attraction vectors by using the label information of training samples and make the samples in the subspace drawn to these attraction points.

Assume that there has a set of training samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$, where $\mathbf{x}_i \in R^m$, $\mathbf{y}_i \in \{1, 2, \cdots, c\}$, $m$ is the dimensionality of the training samples, $n$ is the total number of the training samples, and $C$ is the number of classes. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in R^{m \times n}$ be the training samples matrix. For each training sample, we construct an attraction point $\mathbf{h}_i \in R^C$ by using its label information. If $\mathbf{y}_i = c$, then the $c$th entry of $\mathbf{h}_i$ is 1 and other entries are zero. Thus training samples belonging to the same class share the same attraction point. We hope that each sample in the subspace could be attracted to its attraction point.

To make the tradeoff between the geometric characteristics of low-dimensional coordinate point and label information, we seek for an optimal projection matrix $\mathbf{A} \in R^{m \times C}$, which can project training points into a relatively low-dimensional feature subspace. The sample matrix in the subspace could be represented as $\mathbf{Y} = \mathbf{A}^T\mathbf{X}$.

Based on NPE, we introduce a discriminate information term and obtain the following optimal problem:

$$\min_{\mathbf{A}} \quad \frac{1}{2}\sum_{i=1}^{n}\left\|\mathbf{y}_i - \sum_{j=1}^{k}\mathbf{W}_{ij}\mathbf{y}_j\right\|^2 + \beta\sum_{i=1}^{n}\|\mathbf{y}_i - \mathbf{h}_i\|^2 \qquad (6)$$

where $\beta \in [0, +\infty)$ is a balance parameter which is used to balance the importance of the label information. The reconstruction weighting coefficient $\mathbf{W}$ can be obtained by solving (1).

From (2) we can know that the first term in (6) can be written as

$$\sum_{i=1}^{n}\left\|\mathbf{y}_i - \sum_{j=1}^{k}\mathbf{W}_{ij}\mathbf{y}_j\right\|^2 = \mathbf{YMY} = \mathbf{A}^T\mathbf{XMX}^T\mathbf{A} \qquad (7)$$

and the second term in (6) can be modified as

$$\begin{aligned}\beta\sum_{i=1}^{n}\|\mathbf{y}_i - \mathbf{h}_i\|^2 &= \beta(\mathbf{Y} - \mathbf{H})(\mathbf{Y} - \mathbf{H})^T \\ &= \beta(\mathbf{A}^T\mathbf{X} - \mathbf{H})(\mathbf{A}^T\mathbf{X} - \mathbf{H})^T\end{aligned} \qquad (8)$$

where the attraction matrix $\mathbf{H} = [\mathbf{h}_1, \cdots, \mathbf{h}_n] \in R^{C \times n}$. Obviously, (6) is equivalent to the following optimal problem

$$\min_{\mathbf{A}} \quad tr\left(\mathbf{A}^T\mathbf{XMX}^T\mathbf{A} + \beta(\mathbf{A}^T\mathbf{X} - \mathbf{H})(\mathbf{A}^T\mathbf{X} - \mathbf{H})^T\right) \qquad (9)$$

We show the solution to (9) in the following theorem.

**Theorem 1.** Given the symmetric, and semi-positive definite matrix $\mathbf{M} \in R^{n \times n}$, the real matrix $\mathbf{H} \in R^{C \times n}$, $\beta \in R^+$ and a full rank matrix $\mathbf{X} \in R^{m \times n}$, the object function of (9) has its optimal value when $\mathbf{A} = \beta(\mathbf{XMX}^T + \beta\mathbf{XX}^T)^{-T}\mathbf{XH}^T$.

**Proof.** Let $L(\mathbf{A}) = \mathbf{A}^T\mathbf{XMX}^T\mathbf{A} + \beta(\mathbf{A}^T\mathbf{X} - \mathbf{H})(\mathbf{A}^T\mathbf{X} - \mathbf{H})^T$. Obviously $tr(L(\mathbf{A}))$ will have the minimal value when the derivative of $L(\mathbf{A})$ equals to 0, that is

$$\frac{\partial L(\mathbf{A})}{\partial \mathbf{A}} = 0 \Rightarrow \mathbf{A}^T\mathbf{XMX}^T + \beta(\mathbf{A}^T\mathbf{X} - \mathbf{H})\mathbf{X}^T = 0 \qquad (10)$$

(10) can be modified as

$$(\mathbf{XMX}^T + \beta\mathbf{XX}^T)^T\mathbf{A} = \beta\mathbf{XH}^T \qquad (11)$$

Now we need to certify that $\mathbf{XMX}^T + \beta\mathbf{XX}^T$ is invertible, which could be rewritten as

$$\mathbf{XMX}^T + \beta\mathbf{XX}^T = \mathbf{X}(\mathbf{M} + \beta\mathbf{I})\mathbf{X}^T \qquad (12)$$

where $\mathbf{I} \in R^{n \times n}$ is the identity matrix.

The matrix $\mathbf{X} \in R^{m \times n}$ is a full rank real matrix. If $m < n$, $\mathbf{X}$ is a row full rank matrix, otherwise $\mathbf{X}$ is a column full rank matrix. Assume $\mathbf{X}$ is a row full rank matrix, and the rank of $\mathbf{X}$ is $m$. The case of column full rank is similar to that of row full rank. Thus, we only observe that $\mathbf{X}(\mathbf{M} + \beta\mathbf{I})\mathbf{X}^T$ is invertible under the circumstance of row full rank.

Since $\mathbf{M} + \beta\mathbf{I}$ is real symmetric positive definite matrix, we can apply the square root decomposition on $\mathbf{M} + \beta\mathbf{I}$ as follows:

$$\mathbf{M} + \beta\mathbf{I} = \mathbf{LL}^T$$

where $\mathbf{L} \in \mathbf{R}^{n \times n}$ is a positive definite full rank matrix. So (12) can be represented as

$$\mathbf{X}(\mathbf{M} + \beta\mathbf{I})\mathbf{X}^T = \mathbf{DD}^T$$

where $\mathbf{D} = \mathbf{XL}$ is still a full rank matrix. We only verify $\mathbf{DD}^T$ is invertible as follows.

For an arbitrary vector $\mathbf{s} \in R^m$, where $\mathbf{s}$ is a nonzero vector, we can get $\mathbf{s}^T(\mathbf{D}^T\mathbf{D})\mathbf{s} = (\mathbf{Ds})^T(\mathbf{Ds})$. Let $\mathbf{Ds} = [t_1, t_2, ..., t_m]^T$, thus $\mathbf{s}^T(\mathbf{D}^T\mathbf{D})\mathbf{s} = t_1^2 + t_2^2 + ... + t_m^2 \geq 0$. If $\mathbf{s}^T(\mathbf{D}^T\mathbf{s})\mathbf{D}=0$, then $t_1=t_2=\cdots=t_m=0$. The linear set of equations $\mathbf{Ds}=\mathbf{0}$ has a zero solution, or $\mathbf{s}=\mathbf{0}$ where $\mathbf{0}$ denotes a vector of all zeros. Thus, we have $rank(\mathbf{D}) < m$ which contradicts the known or $\mathbf{s}$ is a nonzero vector. Therefore, we can get $t_1^2 + t_2^2 + ... + t_m^2 > 0$. Finally we can observe that $\mathbf{DD}^T$ is non-singular, that is, $\mathbf{X}(\mathbf{M} + \beta\mathbf{I})\mathbf{X}^T$ is invertible.

This means $L(\mathbf{A})$ would get the minimal value when

$$\mathbf{A} = \beta\left(\mathbf{XMX}^T + \beta\mathbf{XX}^T\right)^{-T}\mathbf{XH}^T \qquad (13)$$

This completes the proof.

From Theorem 1, equation (9) is cast into a linear set of equations, which is very easier to solve. Our approach is a promotion to the unsupervised NPE algorithm. We add discriminative information by constructing an attraction matrix and introduce a parameter $\beta$ to control the importance of the label information. If the label information is not untrustworthy, we can let $\beta = 0$. Equation (9) is equivalent to (2).

The improvement of SNPE algorithm over NPE method benefits mostly from two aspects: one aspect is that SNPE tries to find the subspace that best discriminates different face classes; the other aspect is that SNPE reduces the energy of noise and transformation difference as much as possible.

## IV. EXPERIMENTS

To verify the effectiveness of SNPE, two experiments are carried out here. The first one is performed on a two-dimensional artificial dataset. In the second experiment, two well-known and benchmark face image databases (ORL and Yale [24]) are used to evaluate the performance of SNPE by comparing with PCA, LDA, LPP, NPE, and NPDE. To make the comparison fair, for all the evaluated algorithms, we first apply PCA on the face data to reduce the dimensionality and remove the noise. A nearest neighbor (NN) classifier is employed to classify the projected samples. The experiments are implemented on MATLAB platform.

### *A. Artificial dataset*

In order to compare SNPE with NPE and NPDE and analyze the involved parameter, we generate a two-class synthetic dataset which can better be visualized in the 3-dimensional space. The first class is generated from the Gaussian distribution with a mean $[0,0,0]^T$ and a covariance matrix $\mathbf{I} \in R^{3\times3}$, while the second class is generated from the Gaussian distribution with a mean $[2,2,2]^T$ and a covariance matrix $\mathbf{I} \in R^{3\times3}$. We randomly generate 100 datasets. Each class has 100 training and 100 test points in each trial. We try to project these points into a two-dimensional space. NPE, NPDE and SNPE all have a neighborhood parameter $k$ when constructing the adjacent graph. In addition, SNPE is also involved in a control parameter $\beta$ which has an effect on results of embedding projection theoretically.

We first observe the effect of $\beta$ on the recognition accuracy of SNPE when setting $k = 5$. Let $\beta$ change in the set $\{2^{-4}, 2^{-3}, \cdots, 2^4\}$. We repeat 100 times independently and report the average recognition accuracy as depicted in Fig 1. From this figure, we can see that the curve remains flat with varying the parameter $\beta$ since the standard deviation on the whole parameter set is $1.089\times10^{-6}$, which is very small so that it can be ignored approximately. Therefore, in order to make the tradeoff between the original manifold geometry and label information of training samples, $\beta$=1 is an ideal choice in the experiment.

Since NPE, NPDE and SNPE are related to the adjacency graph neighborhood parameter $k$, which has different effects on projection results. For these three methods, $k$ varies from 1 to 15. The experiments are repeated 100 times and the average accuracy is recorded. Fig. 2 illustrates that recognition accuracy curves of four methods vary with different neighborhood parameters. The curve of "NN" in Fig. 2 denotes the result of the nearest neighbor classifier without dimensionality reduction. Since "NN" is independent of the graph neighborhood parameter, the result of "NN" is a fixed value. Observation on Fig. 2 indicates that SPNE is much better than the other three methods and is relative stable when varying $k$. NPE and NPDE are sensitive to selection for neighbor parameter with different degrees. NPDE has a relatively large fluctuation. These three methods almost have a higher accuracy when $k = 3$. The following experiments, $k$ is taken to 3.

Fig. 3(a) shows the randomly generated data points in one trial. Two-class points are represented by '+' and 'o', respectively. Fig 3 (b), (c) and (d) show the projected points in the two-dimension space obtained by NPE, NPDE and SNPE, respectively. Note that the data points overlap significantly in the original 3-dimensional space, the dimensionality reduction data points would still overlap. We can see that the projected point obtained by NPDE is greatly overlapping, which shows that the method cannot utilize the label information well and perform even worse than NPE. Relatively speaking, SNPE is better than the other two methods, which means it is easy to perform classification tasks. Visualization on points in Fig. 3 also reflects the results of Fig. 2.

### *B. ORL Database*

The ORL face database consists of a total of 400 face images, and of a total of 40 people (10 samples per person). For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glassed/no glassed). All the images were taken against a dark homogeneous background with the subjects in an upright, front position

(with tolerance for some side movement). All images are grayscale and normalized to a resolution of 92×112 pixels.

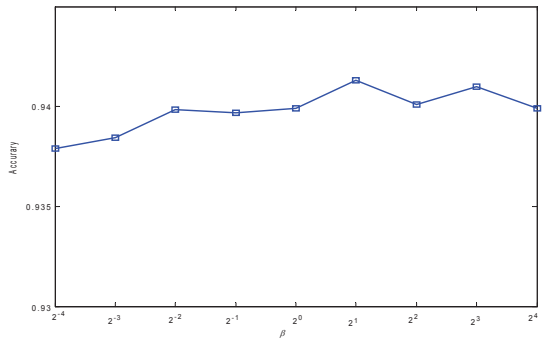



Fig.1 Recognition accuracy under different control parameters

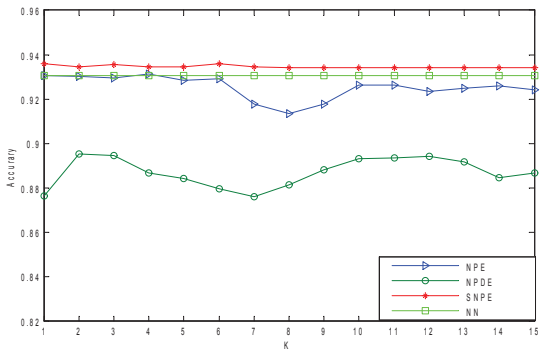
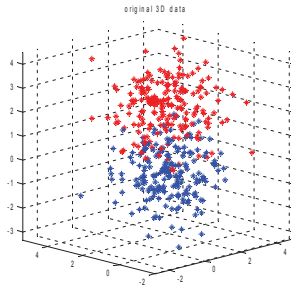
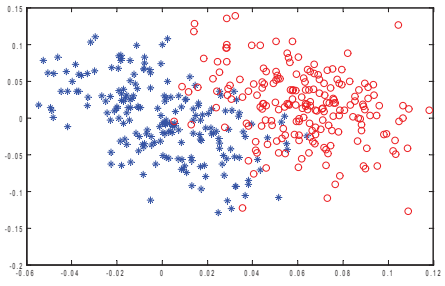


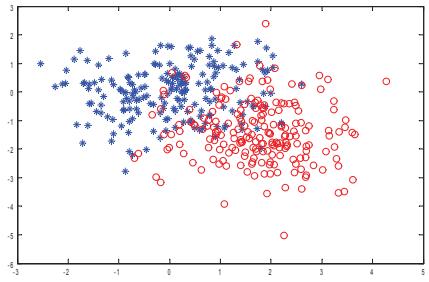

Fig.2 Recognition accuracy under different neighborhood parameter



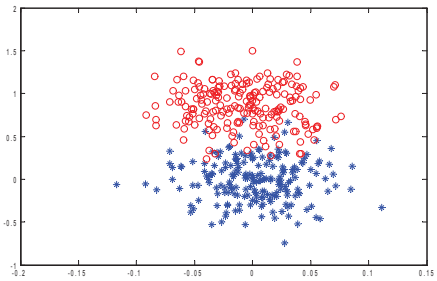

(a)Random data



(b) NPE



(c) NPDE



(d) SNPE

Fig.3 Random data and the results of visualization

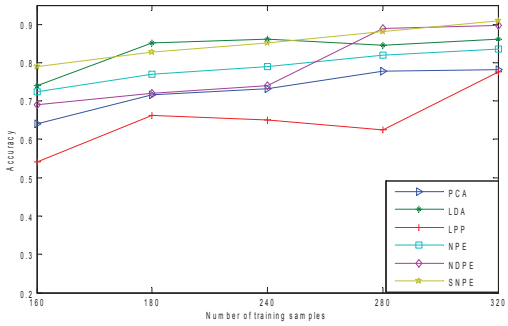



Fig.4. Results on the ORL database

In this experiment, we certify the recognition accuracy of several algorithms with different training samples. We randomly select $n$ samples of each individual for training, and the rest $(10-n)$ for testing. The number of the training sample set varies from 160 to 320. To overcome the complication of singular matrices, we first apply PCA on the face data to reduce dimensionality and remove the noise and remove its null space so that the resulting matrix is non-singular. Then the proposed method is used for feature extraction, that is, other methods are used in this 100-dimensional space, including LDA, LPP, NPE, NPDE and SNPE. For PCA, the dimensionality of subspace is 100. Since the rank of $\mathbf{S}_B$ is $C-1$ in (5), the final dimensionality of LDA is 39, and the other approach is 40. Finally, the nearest neighbor classifier is used for classification.

We perform 100 trials to randomly choose the training set and calculate the average recognition rates. To compare PCA, LDA, LPP, NPE, NPDE and SNPE under the condition of different training samples, we give the average classification rate curves in Fig. 4. We observe that the

recognition accuracy of each method increases when increasing the number of training samples. SNPE is always better than PCA, NPE and LPP. In addition, SNPE performs better than other methods in the case of a few training samples. Compared with NDPE, SNPE is dominant with less training samples. LDA method is only better than SNPE when the number of training samples is from 200 to 240.

### C. Yale Database

The Yale face database contains 165 gray scale images of 15 individuals, and each individual has 11 images. The images demonstrate variations in lighting condition, facial expression (normal, happy, sad, sleepy, surprised, and wink). Fig. 5 shows sample images of one person.

We randomly select some samples of each individual for training, and the rest of the Yale database for testing. The number of the training samples set changes from 45 to 120. The experimental setup is the same as Section 4.2. First, we reduce dimension of the training samples to 100 by using PCA, and then other methods are used for the second dimensionality reduction. The dimension for PCA is 100, for LDA is 14, and for the other approaches are 15. The average results on 100 independent experiments are shown in Fig. 6.

Fig.5 Images of one person in Yale.

Fig. 6 depicts that the average classification rate curves of six methods with different number of training samples. From the experimental results, we can also see that the performance of supervised techniques (or LDA, NPDE, and SNPE) is always better than unsupervised techniques (or PCA, NPE and LPP). In three supervised methods, SNPE is the best one, which indicates that SNPE can make full use of the label information.

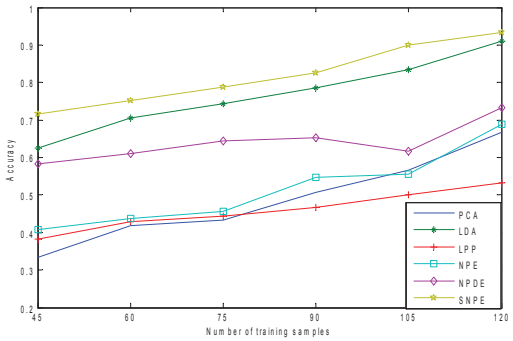



Fig.6 Results on the Yale database

## V. Conclusion

This paper proposes a supervised neighborhood preserving embedding approach which can not only maintain the geometrical manifold structure but also use the label information on the given data. SPNE first builds the neighbor graph in the high-dimensional space, and then find the weight matrix of the adjacency graph. Finally, SPNE learns the projection matrix by using the label information and projects samples from the high-dimensional space into a low-dimensional space. As can be seen in the experiment, SNPE has an advantage in face recognition tasks. SNPE gives better results than other algorithm under different number of training samples. Moreover, SNPE also has a higher recognition rate when the number of training samples is not too much.

## Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61373093 and 61033013, by the Natural Science Foundation of Jiangsu Province of China under Grant Nos. BK2011284 and BK201222725, by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No.13KJA520001, and by the Qing Lan Project.

## References

[1] J.H. Friedman, “On bias, variance, 0/1—loss, and the curse of dimensionality,” Data mining and knowledge discovery, 1997, pp: 55-77.

[2] J. Amador, “Random Projection and Orthonormality for Lossy Image Compression, ”Image and Vision Computing, 2007.

[3] R. Duda, P. Hart, and D. Stork, Pattern Classification, 2nd ed., Wiley, 2000, pp.1-15.

[4] R. A. Fisher, “The Use of Multiple Measurements in Taxonomic Problems, ”Annals of Eugenics, 1936.

[5] I .T .Jolliffe, Principal component analysis,New York: Springer- Verlag, ch. 5.

[6] J. Yang, D. Zhang, A.F. Frangi, J.Y. Yang, “Two-dimensional PCA: a new approach to appearance-based face representation and recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, pp: 131–137.

[7] B. Mikhail, N. Partha, “Laplacian eignmaps for dimensionality reduction and data representation,” Neural Computation, 2003, pp: 1373-1396.

[8] T. R. Sam, K.S. Lawrence, “Nonlinear Dimensionality Reduction by Locally Linear Embedding,” Science，2000,pp: 2323-2326.

[9] X.F. He, N. Partha, “Locality Preserving Projections,” Proceedings of the 17th Annual Conference on Neural Information Processing Systems, Vancouver, 2003, pp: 153-160.

[10] X.F. He, D. Cai, S. C, Yan et al, “Neighborhood preserving embedding,” In: Proceedings of 10th IEEE International Conference on Computer Visions[C], Washington D C, America, 2005, pp: 2208-1213.

[11] H. Murase, S.K. Nayar, “Visual learning and recognition of 3-D objects from appearance,” International Journal of Computer Vision, 1995, pp: 5–24.

[12] P.N. Belhumeur, J.P. Hespanha, D.J. Krieg man, “Eigen faces vs. Fisher faces: recognition using class specific linear projection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997, pp: 711–720.

[13] R. Chellappa, C.L. Wilson, S. Sirohey, “Human and machine recognition of faces: a survey,” Proc. IEEE, 1995, pp: 705–740.

[14] W. Zhao, R. Chellappa, A. Rosenfeld, P.J. Phillips, “Face recognition: a literature survey,” ACM Compute. Surv. 2003, pp: 399–458.

[15] A.M. Martinez, A.C. Kak, “PCA versus LDA,” IEEE Trans. Pattern

Anal. Mach. Intell, 2001, pp: 228–233.

[16] M. Turk, A. Pentland, “Eigenfaces for recognition,” J. Cognitive Neurosci, 1991, pp: 71–86.

[17] J. Yang, D. Zhang, A.F. Frangi, J-Y. Yang, “Two-dimensional PCA: a new approach to appearance-based face representation and recognition,”IEEE Trans. Pattern Anal. Mach. Intell, 2004, pp: 131–137.

[18] Y. Xu, D. Zhang, J. Yang, J.-Y. Yang, “An approach for directly extracting features from matrix data and its application in face recognition, Neurocom- putting,” 2008, pp: 1857–1865.

[19] Y. Xu, D. Zhang, J.-Y. Yang, “A feature extraction method for use with bimodal biometrics,” Pattern Recognition 2010, pp: 1106–1115.

[20] HS.Du,XL.Chai, “Face recognition method using neighborhood preserving discriminant embedding,”2010,pp: 625–629.

[21] L.S. Qiao, S.C. Chen, X.Y. Tan, “Sparsity preserving projections with applications to face recognition,” Pattern Recognition 2010, pp:331–341. January

[22] J. Gui, Z. N. Sun, “Discriminant sparse neighborhood preserving embedding for face recognition,” Pattern Recognition, 2012, pp: 2884–2893.

[23] G.F. Lu, Z. Lin, Z. Jin, “Face recognition using discriminant locality preserving projections based on maximum margin criterion,” Pattern Recognition, 2010, pp :3572–3579.

[24] A. S. Georghiades, P. N. Belhumeur, et al, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” IEEE Transaction on Pattern Analysis and Machine Intelligence, 2001, pp: 643-660.

2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# Similarity-balanced Discriminant Neighborhood Embedding

Chuntao Ding
School of Computer Science and Technology & Provincial Key Laboratory for Computer Information Processing
Suzhou, China
Email: 20124227036@suda.edu.cn

Li Zhang
School of Computer Science and Technology Provincial Key Laboratory for Computer Information Processing
Suzhou, China
Email: zhangliml@suda.edu.cn

Yaping Lu
School of Computer Science and Technology Provincial Key Laboratory for Computer Information Processing
Suzhou, China
Email: 20134227010@stu.suda.edu.cn

Shuping He
School of Computer Science and Technology Provincial Key Laboratory for Computer Information Processing
Suzhou, China
Email: sphe@suda.edu.cn

***Abstract*—The idea that with the help of proper dimensionality reduction, trying to make the samples with the same label be compact and the ones with the different labels be separate after projection, is introduced into classification problems with high-dimensional data. Based on the analysis of the drawbacks of Discriminant Neighborhood Embedding (DNE) and Locality-Based Discriminant Neighborhood Embedding (LDNE), being the two relatively successful Locally Discriminant Analysis methods proposed in recent years, this paper proposes a method called Similarity-balanced Discriminant Neighborhood Embedding (SBDNE). When constructing the adjacent graph, SBDNE fully takes into account the geometric construction of manifold and the problem of imbalance between the intra-class points and the inter-class points. By endowing these two kinds of samples with different similarities and selecting the near neighbors according to the similarity matrix, not only the structure in the original space can be preserved more efficiently, but also the choice of discriminative information increases. The method proposed here has a better recognition with comparisons to some classical methods, which fully shows that SBDNE method has the capacity to efficiently solve the classification problem.**

***Keywords—discriminant neighborhood embedding; adjacent graph; intra-class; inter-class***

## I. INTRODUCTION

There are more and more study and application fields that need to deal with high-dimensional data. As a result, to achieve its analysis and visualization, we have to reduce the dimensionality so as to make the high-dimensional data embed into relatively low-dimensional feature subspace with the inner structure of data preserved. This skill is widely used in the fields such as computer vision, machine learning and pattern recognition and so on.

As classical methods, both Principal Component Analysis (PCA) [1-2] and Linear Discriminant Analysis (LDA) [2-3] assume that the data processed are from the Euclidean space. However, the manifold learning [4-6], rising after 2000, shows that many complex objects are situated in some manifold subspace and their non-linear inner structure cannot be learned via traditional methods. Nevertheless, manifold learning algorithms only consider the training samples and cannot get an explicit mapping, so they cannot perform incremental learning for new data, namely called the out-of-sample problem, because of which manifold learning methods are under restrictions for classification. To cover this shortage, He et al. proposed Locality Preserving Projection (LPP) [7] and Neighborhood Preserving Embedding (NPE) [8], both of which can directly map the new sample into a low-dimensional subspace via the projection matrix obtained by the training procedure.

Dimensionality reduction methods are composed of unsupervised ones and supervised ones. The former focuses on the better representations of high-dimensional data without considering the labels, and the latter tries to achieve the classification efficiently with the labels employed. They are also called represented dimensionality reduction and discriminative dimensionality reduction. As a classical linear discriminative dimensionality reduction method, LDA tries to find a projection direction, being conducive to discriminate, by minimizing the divergence of samples with the same class and maximizing the divergence of samples with the different classes. Although LDA has been widely used in pattern recognition field, it still has the problem of the small sample size and requires the data to obey a Gaussian distribution. However, the practical data often dissatisfy the hypothesis, so, to overcome this drawback, maximum margin criterion (MMC) [18] and margin Fisher analysis (MFA) [9-10] methods have been proposed. MMC is mainly to maximize the difference between inter-class and intra-class scatters and MFA as an extension of LDA, MFA is able to efficiently solve the problems discussed above. For MFA, the locally structure of samples is preserved by constructing the homogeneous and heterogeneous neighbor adjacency graphs, and the optimal projection direction is found by minimizing the ratio of the sum of distance between the samples with the same class and the sum of distance between the samples with the different classes.

Dimensionality reduction methods can also be divided into the non-graph-structure-based ones and the graph-structure-based ones. The former directly reduces the dimensionality without taking into account the structure information of data in the original space, and the latter makes the geometric structure

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

of data in high-dimensional space still be preserved in low-dimensional space by constructing the preserved structure graph. As a relatively classical graph-structure-based method, LPP achieves to preserve the local structure of original data by making the samples being close to each other in original space and still be close to each other in low-dimensional space. However, LPP is an unsupervised method that does not efficiently utilize the label information. , which might degrade their performance in pattern recognition. Based on the idea of LPP, many methods have been proposed, such as Supervised Locality Preserving Projections (SLPP) [11] and Neighborhood Discriminant Projection for Face Recognition (NDP) [12] and so on, we can easily see that these supervised methods mainly make use of class label information to well guide the procedure of dimensionality reduction. Among which the Discriminant Neighborhood Embedding (DNE) [13] proposed by Zhang et al. is a much efficient method. For DNE method, first, by constructing an adjacent graph, the relationship between the samples in original space and their neighbors, including the same class and the different classes, is preserved, then make the samples have the same structure in the low-dimensional space, and finally by employing the spectral analysis the dimensionality of discriminative subspace is calculated. However, DNE cannot preserve the detailed position relationship between the samples and their neighbors, including the same class and the different classes. As a consequence, the recognition rate in low-dimensional space would decrease when the data are unbalanced. By constructing a different adjacent graph with DNE method and endowing different weights, Locality-Based Discriminant Neighborhood Embedding (LDNE) proposed in [14] makes the optimization problem change to optimize the difference between the distance of samples with the same class and the distance of samples with the different classes.

Based on the analysis of the drawbacks of DNE and LDNE, this paper proposes a new supervised dimensionality reduction method called Similarity-balanced Discriminant Neighborhood Embedding (SBDNE). By introducing a new similarity function, SBDNE endows the data with the same class and the different class with different similarity functions. The similar neighbors are selected based on the matrix composed of the similarity functions. By constructing the structure graphs of the samples with the same class and the samples with the different classes and utilizing the geometric structure of manifold, the unbalanced problem between the same class and the different classes is solved. As a result, not only the structure in the original space can be preserved more efficiently, but also the choice of discriminative information increases. Finally, experimental results on the artificial dataset, ORL face dataset , Yale face dataset and FERET face dataset show the effectiveness of SBDNE.

## II. RELATED WORK

In this section, we review both DNE and LDNE, which are supervised dimensionality reduction methods. Suppose we have the training samples $\{\mathbf{x}_i, y_i\}_{i=1}^N, \mathbf{x}_i \in R^d, y_i = \{1,2,...,c\}$ , where $y_i$ is the label of $\mathbf{x}_i$. $c$ , $N$ and $d$ respectively denotes the number of classes, the number of samples and the dimensionality of samples. The purpose of DNE and LDNE is to find a linear projection that maps the data in the $d$ -dimensional space into the $r$ -dimensional subspace, such as $\mathbf{v}_i = \mathbf{P}^T\mathbf{x}_i$ where $\mathbf{v}_i$ represents the low-dimensional data after projection and $\mathbf{P} \in R^{d\times r}$ is the projection matrix.

### A. Discriminant Neighborhood Embedding

DNE aims to make the samples with the same label form a compact sub-manifold and the distance between the samples with the different labels are as far as possible in the low-dimensional subspace after projection. The process of DNE method is as follows:

(1) Define an adjacent graph $\mathbf{F}$ , of which the element $F_{ij}$ is given by

$$F_{ij} = \begin{cases} +1\ , & \mathbf{x}_i \in \aleph_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \aleph_k(\mathbf{x}_i) \text{ and } (y_i = y_j) \\ -1\ , & \mathbf{x}_i \in \aleph_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \aleph_k(\mathbf{x}_i) \text{ and } (y_i \neq y_j) \\ 0\ , & \text{otherwise} \end{cases} \quad (1)$$

where $\aleph_k(\mathbf{x}_j)$ denotes the set of $k$ nearest neighbors of $\mathbf{x}_j$ , $y_i$ and $y_j$ respectively represent the labels of $\mathbf{x}_i$ and $\mathbf{x}_j$ .

(2) Feature mapping: Optimize the following objective function:

$$\begin{cases} \min \sum_{ij} \| \mathbf{P}^T\mathbf{x}_i - \mathbf{P}^T\mathbf{x}_j \|^2 F_{ij} \\ s.t. \quad \mathbf{P}^T\mathbf{P} = \mathbf{I}. \end{cases} \quad (2)$$

where $\mathbf{I}$ is the identity matrix, and $\mathbf{P}$ is the projection matrix. Through a simple derivation, the optimization problem changes to be as follows:

$$\begin{cases} \min\limits_{\mathbf{P}} tr(\mathbf{P}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{P}) \\ s.t. \qquad \mathbf{P}^T\mathbf{P} = \mathbf{I} \end{cases} \quad (3)$$

where $\mathbf{L} = \mathbf{D} - \mathbf{F}$ , $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j F_{ij}$ and $tr(\cdot)$ is the trace of matrix. Finally, the projection matrix $\mathbf{P}$ can be obtained by the decomposition of the proper value of a matrix according to the following objective function:

$$\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{P} = \lambda\mathbf{P} \quad (4)$$

where the optimal projection matrix $\mathbf{P}$ is composed of the $r$ eigenvectors corresponding to the $r$ minimum eigenvalues.

### B. Locality-Based Discriminant Neighborhood Embedding

Based on DNE and by endowing the adjacent graph with different weights, LDNE is able to preserve the nearest neighbors. Moreover, it also tries to find an optimal projection matrix by maximizing the difference between the aggregation of samples with the same class and the divergence of samples with the different classes. The process of LDNE method is as follows:

(1) According to the $k$ nearest neighbors rule, construct a similarity matrix $\mathbf{S}$ by

$$S_{ij}=\begin{cases} -\exp\left(\dfrac{-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{\beta}\right), & if \quad (y_i=y_j) \text{ and} \\ & (\mathbf{x}_i\in\aleph_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j\in\aleph_k(\mathbf{x}_i)) \\ +\exp\left(\dfrac{-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{\beta}\right), & if \quad (y_i\neq y_j) \text{ and} \\ & (\mathbf{x}_i\in\aleph_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j\in\aleph_k(\mathbf{x}_i)) \\ 0, & otherwise \end{cases} \tag{5}$$

where $\beta>0$ is the parameter selected by users.

(2) Feature mapping: Optimize the following objective function:

$$\begin{cases} \max \sum_{ij} \| \mathbf{P}^T\mathbf{x}_i-\mathbf{P}^T\mathbf{x}_j \|^2 S_{ij} \\ s.t. \quad \mathbf{P}^T\mathbf{P}=\mathbf{I}. \end{cases} \tag{6}$$

Through a simple derivation, the optimization problem changes to be as follows:

$$\begin{cases} \max\limits_{\mathbf{P}} tr(\mathbf{P}^T\mathbf{X}\mathbf{H}\mathbf{X}^T\mathbf{P}) \\ s.t. \qquad \mathbf{P}^T\mathbf{P}=\mathbf{I} \end{cases} \tag{7}$$

where $\mathbf{H}=\mathbf{D}-\mathbf{S}$, $\mathbf{D}$ is a diagonal matrix, of which the diagonal elements are composed of the sum of $\mathbf{S}$ by row or by column, such as $D_{ii}=\sum_j S_{ij}$.

Being similar to the DNE method, the projection matrix $\mathbf{P}$ of LDNE can also be obtained by the decomposition of the proper value of a matrix according to the following objective function:

$$\mathbf{X}\mathbf{H}\mathbf{X}^T\mathbf{P}=\lambda\mathbf{P} \tag{8}$$

where $\mathbf{P}$ is composed of the $r$ eigenvectors corresponding to the $r$ maximum eigenvalues.

### C. The Drawbacks of DNE and LDNE

According to the discussion above, we know that when constructing an adjacent graph, DNE only endows the samples with the same label with +1 and the ones with the different labels with -1, which would lead to three drawbacks. First, the locally structure information of data cannot be preserved. Second, sometime, it cannot be efficiently act on the samples with the same label and the ones with the different labels at the same time. Third, when the data are unbalanced, all the nearest neighbors of an example may completely belong to the same class or the different classes so that when constructing an adjacent graph it cannot find the association between the samples with the same label or the different labels. As a consequence, DNE may not find the most efficient sub-manifold.

For LDNE method, it achieves to preserve the locally structure information of data by calculating the similarity between the example and its neighbors. But it has two drawbacks. On the one hand, it is not obvious to distinguish the relationship between the same class and the different classes since they are endowed with the same similarity function. On the other hand, being similar to DNE, when the data are unbalanced, it may not find the most efficient sub-manifold as well.

## III. SIMILARITY-BALANCED DISCRIMINANT NEIGHBORHOOD EMBEDDING

To overcome the drawbacks of DNE and LDEN, this paper proposes a new supervised sub-manifold learning algorithm called Similarity-balanced Discriminant Neighborhood Embedding (SBDNE). In detail, it is able to efficiently make the samples with the same label be aggregated and the ones with the different classes be separated in the low-dimensional subspace so as to get a better classification performance.

### A. Similarity function

Suppose we have the training samples $\{\mathbf{x}_i,y_i\}_{i=1}^N$. Then, we define a new similarity function between $\mathbf{x}_i$ and $\mathbf{x}_j$ as follows:

$$G(\mathbf{x}_i,\mathbf{x}_j)=\begin{cases} \exp\left(\dfrac{-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{\beta}\right)\exp\left(\exp\left(\dfrac{-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{\beta}\right)+1\right), & \text{if } y_i=y_j \\ \exp\left(\dfrac{-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{\beta}\right)\exp\left(1-\exp\left(\dfrac{-\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{\beta}\right)\right), & \text{if } y_i\neq y_j \end{cases} \tag{9}$$

From (9), we know that the similarity functions for the samples with the same label and the ones with the different labels are different. Specifically, the previous ones are endowed with larger weights and the latter ones are endowed with smaller weights. Fig. 1 shows the curves of similarity function $G(\mathbf{x}_i,\mathbf{x}_j)$ vs the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_j$. When these two samples belong to the same class, the similarity rapidly decreases with the increase of their distance. If they belong to different classes, the similarity slowly decreases with the increase of their distance. Note that the curve for $y_i=y_j$ always lies above that of $y_i\neq y_j$. Moreover, the similarity degree in the same class situates between 0 and $e^2$, but in the different classes the interval changes to be between 0 and 1 so that the similarity degrees for the different classes can be inhibited.

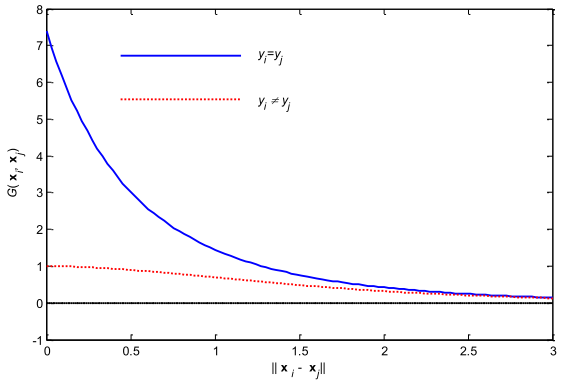
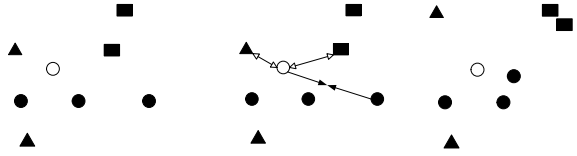



Fig. 1 The similarity between intra-class and inter-class

### B. Construction of adjacent graphs

Now, we consider the construction of adjacent graphs according to the new similarity function (9). Our scheme is to select the farthest homogeneous neighbors for a sample to construct an intra-class graph $\mathbf{F}^w$, and it's nearest heterogeneous neighbors to build an inter-class graph $\mathbf{F}^b$. The

reason can be illustrated by Fig. 2. In Fig. 2(a), there are three classes denoted by solid square, circle and solid triangle. For the hollow circle point, we select the farthest neighbor in the solid circle points, and the nearest neighbors in the solid square and triangle points as shown in Fig. 2(b). Fig. 2(c) ideally gives their images in the subspace. We expect that the farthest homogeneous could be attracted to around the sample and the nearest heterogeneous neighbors could be pushed way from the sample.



(a) (b) (c)

Fig.2 An illustration. (a) The hollow circle point has seven neighbors. (b) The interactions by attraction and repulsion for the points. (c) Projected points in the subspace.

For $F_{ij}^{w}$, we select the $k$ homogeneous samples with the smallest similarity for $\mathbf{x}_i$ and preserve their structural relationships. Namely,

$$F_{ij}^{w}=\begin{cases} G(\mathbf{x}_i,\mathbf{x}_j), & \text{if } \mathbf{x}_i\in N_k^{+}(\mathbf{x}_j) \text{ or } \mathbf{x}_j\in N_k^{+}(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $N_k^{+}(\mathbf{x}_i)$ and $N_k^{+}(\mathbf{x}_j)$ respectively denote the set of farthest homogeneous neighbors of $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathbf{x}_i$ has the same label with $\mathbf{x}_j$. The intra-class compactness has the form:

$$\Phi(\mathbf{P})=\sum_{i,j}\| \mathbf{P}^T\mathbf{x}_i-\mathbf{P}^T\mathbf{x}_j \|^2 F_{ij}^{w} \tag{11}$$

On the contrary, for $F_{ij}^{b}$, $k$ heterogeneous nearest neighbors with the highest similarity are selected for $\mathbf{x}_i$. Then,

$$F_{ij}^{b}=\begin{cases} G(\mathbf{x}_i,\mathbf{x}_j), & \text{if } \mathbf{x}_i\in N_k^{-}(\mathbf{x}_j) \text{ or } \mathbf{x}_j\in N_k^{-}(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

where $N_k^{-}(\mathbf{x}_i)$ and $N_k^{-}(\mathbf{x}_j)$ respectively denote the set of nearest heterogeneous neighbors of $\mathbf{x}_i$ and $\mathbf{x}_j$, and $\mathbf{x}_i$ has the different label with $\mathbf{x}_j$. Thus, we have the inter-class divergence as

$$\Omega(\mathbf{P})=\sum_{i,j}\| \mathbf{P}^T\mathbf{x}_i-\mathbf{P}^T\mathbf{x}_j \|^2 F_{ij}^{b} \quad . \tag{13}$$

By respectively building intra-class structure graph and inter-class structure graph, each example is able to get the associations with the samples with the same or different classes. In other words, for an example, we can get at least two associations, namely the association with the same class and the association with the different classes.

We try to maximize the difference between the nearest inter-class distance and the farthest intra-class distance so as to make the distance between the same classes is nearer and the distance between the different classes is farther in the projection sub-space. That's to say, we need to maximize

$$\Psi(\mathbf{P})=\Phi(\mathbf{P})-\Omega(\mathbf{P}) \quad . \tag{14}$$

Through a simple derivation (see Appendix A), the optimization problem changes to be:

$$\begin{cases} \max\limits_{\mathbf{P}} \; tr(\mathbf{P}^T\mathbf{X}\mathbf{U}\mathbf{X}^T\mathbf{P}) \\ s.t. \qquad \mathbf{P}^T\mathbf{P}=\mathbf{I} \end{cases} \tag{15}$$

where $\mathbf{U}=\mathbf{D}^b-\mathbf{F}^b-\mathbf{D}^w+\mathbf{F}^w$, $\mathbf{D}^b$ and $\mathbf{D}^w$ are diagonal matrixes with $D_{ii}^{b}=\sum_j F_{ij}^{b}$ and $D_{ii}^{w}=\sum_j F_{ij}^{w}$, respectively. The detail of this algorithm is shown in Algorithm 1.

**Algorithm 1 Similarity-balanced Discriminant Neighborhood Embedding (SBDNE)**

Input: Training set $\{(\mathbf{x}_i,y_i)\}_{i=1}^{N}$; sample matrix $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N]\in R^{d\times N}$

Output: Projection matrix $\mathbf{P}$

1). Build the intra-class adjacent graph $\mathbf{F}^w$ and inter-class adjacent graph $\mathbf{F}^b$ according to (10) and (12), respectively.

2). Perform eigendecomposition on the matrix $\mathbf{X}\mathbf{U}\mathbf{X}^T$. Suppose we obtain the eigenvalue $\lambda_i$ and the corresponding eigenvector $\mathbf{p}_i$, and eigenvalues are organized by descending order, $\lambda_1\geq\lambda_2\geq\cdots\geq\lambda_d$.

3). Get the $r$ eigenvectors corresponding to the first $r$ eigenvalues, and then we have the projection matrix as $\mathbf{P}=[\mathbf{p}_1,\mathbf{p}_2,\ldots,\mathbf{p}_r]$.

## IV. EXPERIMENTS

In this section, we will discuss applications of SBDNE, and its comparisons with MFA, DNE and LDNE. Both the number of neighbors $k$ for the four methods and $\beta$ for the SBDNE, and LDNE are the tunable parameters. In our experiments, we select nearest neighbor classifier to classify our data after dimensionality reduction.

### *A. Synthetic Dataset*

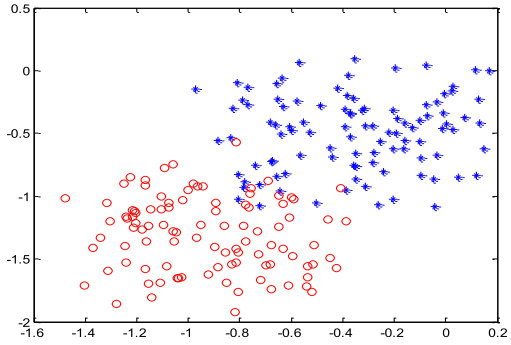
We generate two class samples obeying uniform distribution, ones of which are the random numbers drawn from the interval $[0,1]^5$, and the other ones are drawn from $[0.7,1.7]^5$. There are 200 training and 200 test samples. The projection matrix is learned by DNE, LDNE and SBDNE respectively.

In this experiment, $k$ is selected to be 1 and $\beta$ is got via 10-fold cross-validation for LDNE and SBDNE. The range of $\beta$ is from 1 to 50. Fig. 3 shows the projected data obtained by DNE, LDNE and SBDNE, respectively.

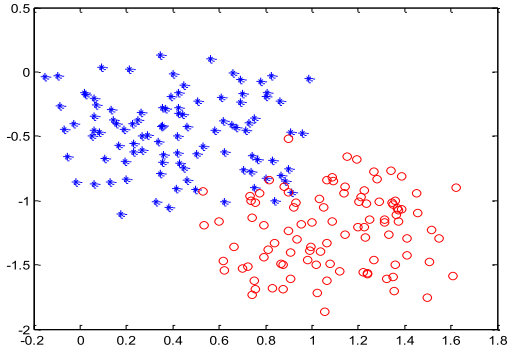
From Fig. 3, we can know that compared with DNE and LDNE, SBDNE works better for classification since it achieves to make the intra-class samples be aggregated and the inter-class ones be separated. From another point of view, this also indicates that the projection matrix learned by SBDNE is more satisfied to classify the samples. For DNE method, it builds the adjacent graph by exploiting the relationships between the samples and their neighbors without considering the locally position information of the samples. For LDNE method, on the one hand, it is not obvious to distinguish the samples when the intra-class samples and the inter-class ones are endowed with the same similarity function, and on the other hand, being similar to DNE method, it may not find the most efficient sub-manifold. On the contrary, SBDNE method fully takes into

account not only the position information of data but also the balanced relationships between the intra-class data and the inter-class data so that it has the better recognition effect.

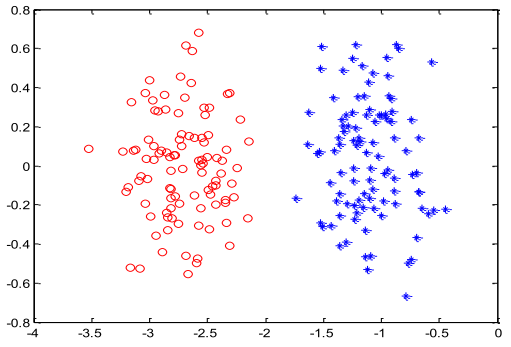
To verify this, Table I provides the quantitative analysis. Table I presents intra-class scatter, inter-class scatter and the ratio between them, where the intra-class scatter is the sum of distances of all two samples in the same class, the inter-class scatter is the sum of all two samples in the different class. Of course, we expect that the inter-class scatter is large, the intra-class scatter is small, and the ratio of inter-class scatter to intra-class scatter is large. The larger the ratio is, the better the separability is. For the raw data, the ratio is 2.072. The projected data obtained by the three methods has a higher ratio. The inter-class scatters in four cases are almost the same. But SBDNE generates a rather smaller intra-class scatter. Thus, the separability on the projected data obtained by SBDNE is the best.



(a) DNE



(b) LDNE



(c) SBDNE

Fig.3 Projected data obtained by DNE (a), LDNE (b) and SBDNE (c).

TABLE I. SEPARABILITY ON SYNTHETIC DATASET

| | **Intra-class scatter** | **Inter-class scatter** | **Ratio of inter-class scatter to Intra-class scatter** |
|---|---|---|---|
| Raw Data | $1.7395 \times 104$ | $3.6044 \times 104$ | 2.0720 |
| DNE | $1.0279 \times 104$ | $2.1449 \times 104$ | 2.0867 |
| LDNE | $1.0252 \times 104$ | $2.1901 \times 104$ | 2.1362 |
| SBDNE | $9.8788 \times 103$ | $3.2958 \times 104$ | 3.3362 |

### *B. Experiments on Face Recognition*

This experiment is based on the three famous datasets, ORL dataset [16], Yale dataset [17] and FERET subset dataset. For SBDNE, MFA, DNE and LDNE, their performance is measured by recognition rate. In the experiment, the parameters $k$ and $\beta$ of SBDNE are selected to be several different sets of values so as to observe their effect on the recognition rate. The whole training set is divided into 60% training set and 40% validation set, and the value of parameter $\beta$ is selected based on the training result on the validation set. Finally, all the methods employ the Nearest Neighbors as their classifier.

### *C. ORL Face DataSet*

The ORL face dataset [16] consists of 400 face images of 40 persons, with 10 images for each person. Some images are taken at different times so that the person's face expression and face detail may have the different degrees of variation such as open eyes or closed eyes, simile or not simile and with glasses or without glasses. Additionally, face posture changes with deep or plane rotation to 20 degree and face size also has the 10% variation. Each image has the grayscale from 0 to 255 with digitization and normalization and is scaled to be 32x32 (this means an image has 1024 features) for the efficient computation. Fig. 4 shows the images of one person from the ORL dataset.

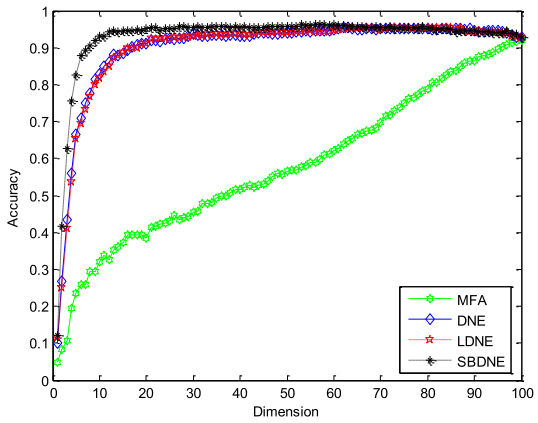
In this experiment, owing to the high dimensionality of ORL dataset, we would reduce dimensionality with two times so as to get a high running speed. Additionally, PCA is employed firstly to reduce the data to be 100 features since it can eliminate the majority of noises. 4 samples of each person in the ORL dataset are selected to be training ones and the rest 6 ones are test ones.

The neighborhood parameter $k$ in MFA, DNE, LDNE and SBDNE is set to be 1 and 3, respectively. We repeat 10 data division for training and test and report the average experimental results in Fig. 5. Although the number of neighbors is different when constructing the adjacent graph, the recognition rates for each method have the consistent tendency. Compared with MFA, DNE and LDNE, SBDNE has a better recognition rate and its optimal discriminative subspace has a relatively low dimensionality so as to reduce the complexity of calculation.

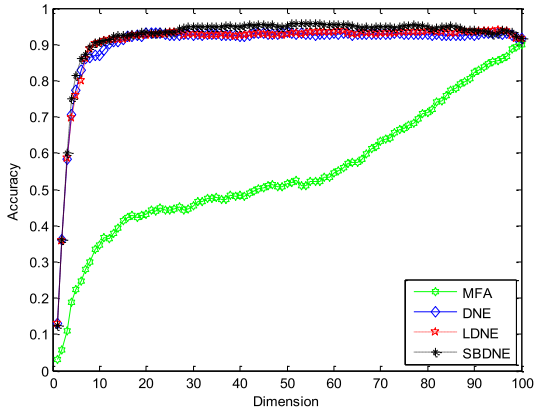
Table II presents the optimal recognition rate and the dimensionality of discriminative sub-space with different number of nearest neighbors. Compared with other methods, SBDNE has not only a better recognition rate but also a lower dimensionality of discriminative sub-space.

When building adjacent graphs, SBDNE not only confirms the locally structure information and the positions of data but also solves the unbalanced problem. By endowing the intra-class data and the inter-class data with different similarities, SBDNE has a more powerful discrimination than MFA, DNE and LDNE so as to make the learned projection matrix be able to more efficiently achieve aggregation among intra-class data and separation among inter-class data.

Fig.4 Samples of face image for the ORL face database




(a) $k=1$




(b) $k=3$

Fig.5 Recognition performance for the ORL database with different neighbor parameter

TABLE II. PERFORMANCE COMPARISON ON THE ORL DATABASE

| | k=1 | | k =3 | |
|---|---|---|---|---|
| | ***Sub-dimension*** | ***Recognition rate (%)*** | ***Sub-dimension*** | ***Recognition rate (%)*** |
| MFA | 100 | $92.08 \pm 0.32$ | 100 | $89.86 \pm 0.33$ |
| DNE | 62 | $95.42 \pm 1.21$ | 50 | $93.33 \pm 2.19$ |
| LDNE | 73 | $95.56 \pm 0.17$ | 84 | $93.89 \pm 0.64$ |
| SBDNE | 53 | $96.25 \pm 0.38$ | 52 | $95.83 \pm 0.43$ |

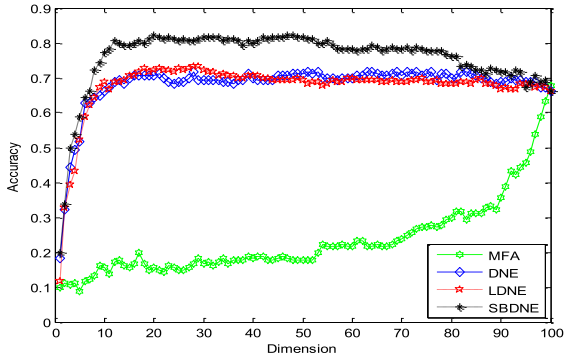
## D. *Yale Dataset*

Yale dataset [17] contains 165 face images of 15 persons, with 11 images for each person. The face expression and light condition for each image are as follows: centered light, with glasses or without glasses, happy, normality, left side light, right side light, sad, sleep, surprise and blink. The size of each image is 32x32 with grayscale from 0 to 255. Fig. 6 shows some face images with different conditions from the Yale dataset.

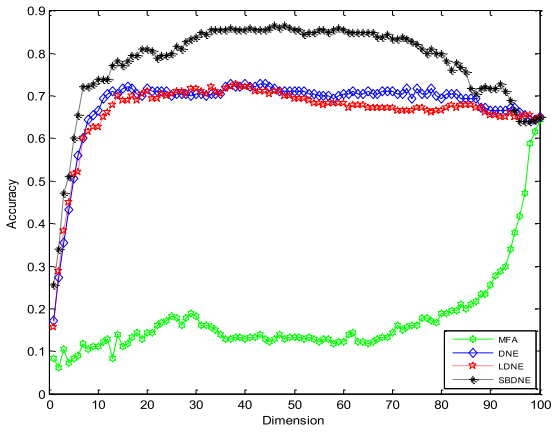
Being similar to ORL face dataset, Yale face samples are also reduced to 100 features via PCA, after which we would respectively employ the methods of MFA, DNE, LDNE and SBDNE to achieve the second dimensionality reduction. Here, we mainly focus on the effect of the number of samples on the recognition rate and all the recognition rates are the average values of 100 experiments with $k=1$. In the experiment, we randomly select 5 (or 7) samples of one person as our training set and the rest ones as test set.

Fig. 7 shows the recognition rates of each discriminative subspace with the different number of samples. From Fig. 7(a) and 7(b), we can see that although the number of samples is different, the general trends of recognition rate are the same. Compared with MFA, DNE and LDNE, SBDNE always presents a better recognition rate and tends to the high recognition rate in a relatively fast speed. Table III provides the optimal recognition rates for the four methods with the different number of samples.

Fig.6 Samples of face image for the Yale face database




(a) 5 training samples




(b) 7 training samples

Fig.7 Recognition performance for the Yale database with different training samples.

TABLE III. PERFORMANCE COMPARISON ON THE YALE DATABASE

| | Five training samples | | Seven training samples | |
|---|---|---|---|---|
| | *Sub-dimension* | *Recognition rate (%)* | *Sub-dimension* | *Recognition rate (%)* |
| MFA | 100 | $67.78 \pm 0.34$ | 100 | $64.78 \pm 0.34$ |
| DNE | 51 | $71.67 \pm 1.12$ | 40 | $72.18 \pm 1.12$ |
| LDNE | 28 | $73.33 \pm 2.63$ | 38 | $72.78 \pm 2.63$ |
| SBDNE | 20 | $82.22 \pm 0.64$ | 46 | $86.67 \pm 0.64$ |

### *E. Yale Dataset*

The FERET database is a standard database for evaluating state-of-art face recognition algorithms. In this experiment, a subset, this contains 1400 face images of 200 individuals with 7 images per individual. Fig. 8 shows Sample images for one individual of the FERET subset

Fig.8 Sample images for one individual of the FERET subset

Being similar to ORL and Yale face dataset, FERET face samples are also reduced to 100 features via PCA, after which we would respectively employ the methods of MFA, DNE, LDNE and SBDNE to achieve the second dimensionality reduction. And also we mainly focus on the effect of the number of samples on the recognition rate and all the recognition rates are the average values of 100 experiments with $k$ =1. In the experiment, we randomly select 3(or 4) samples of one person as our training set and the rest ones as test set.

Table IV provides the optimal recognition rates for the four methods with the different number of samples.

TABLE IV. PERFORMANCE COMPARISON ON THE FERET DATABASE

| | Three training samples | | Four training samples | |
|---|---|---|---|---|
| | *Sub-dimension* | *Recognition rate (%)* | *Sub-dimension* | *Recognition rate (%)* |
| MFA | 100 | $35.75 \pm 3.34$ | 100 | $47.75 \pm 2.34$ |
| DNE | 21 | $50.12 \pm 3.15$ | 38 | $65.50 \pm 1.25$ |
| LDNE | 25 | $52.88 \pm 2.48$ | 38 | $66.75 \pm 0.95$ |
| SBDNE | 20 | $79.37 \pm 0.28$ | 22 | $83.75 \pm 0.80$ |

## V. CONCLUSION

This paper proposes a new linear dimensionality reduction method, called Similarity-balanced Discriminant Neighborhood Embedding (SBDNE). Based on the MFA and LDNE, SBDNE gives some improvements in that the information of samples' positions and the balanced relationship between the intra-class data and the inter-class data are taken into account. As a result, we are able to give an overall consideration on the preservation of manifold's original geometric structure and utility of classification information.

Through numerical experiments, the advantages of SBDNE are verified on the synthetic dataset and the two face image datasets. By directly using the constructed low-dimensional model, it is able to quickly get the low-dimensional information of new test example, at the same time, with a rising recognition rate. However, SBDNE is still a linear method, so to improve on the classification performance, in the future, we would try to extend it to be non-linear.

## APPENDIX A

First, SBDNE computes the similarity function $G(\mathbf{x}_i, \mathbf{x}_j)$ according to the labels, from which we can know that the similarity of the samples with the same class being farthest from one example is the minimal among the samples with the same class, and the similarity of the samples with the different classes being nearest to one example is the maximal among the samples with the different classes.

According to $G(\mathbf{x}_i, \mathbf{x}_j)$, the intra-class structure graph $\mathbf{F}^w$ and the inter-class structure graph $\mathbf{F}^b$ are constructed by (10) and (12), respectively. The intra-class compactness is given by

$$\Phi(\mathbf{P})=\sum_{i,j} \| \mathbf{P}^T\mathbf{x}_i - \mathbf{P}^T\mathbf{x}_j \|^2 \ F_{ij}^w$$

and the inter-class divergence is given by

$$\Omega(\mathbf{P})=\sum_{i,j} \| \mathbf{P}^T\mathbf{x}_i - \mathbf{P}^T\mathbf{x}_j \|^2 \ F_{ij}^b \quad .$$

For SBDNE method, the difference $\Psi(\mathbf{P})$ between the sum of the distances between the nearest samples with the different classes and the sum of the distances between the farthest samples with the same class is maximal and its derivation process is as follows:

$$\begin{aligned}\Psi(\mathbf{P}) &= \Phi(\mathbf{P}) - \Omega(\mathbf{P}) \\ &= 2tr\{\mathbf{P}^T\mathbf{X}(\mathbf{D}^b - \mathbf{F}^b)\mathbf{X}^T\mathbf{P} - 2\mathbf{P}^T\mathbf{X}(\mathbf{D}^w - \mathbf{F}^w)\mathbf{X}^T\mathbf{P}\} \\ &= 2tr\{\mathbf{P}^T\mathbf{X}(\mathbf{D}^b - \mathbf{F}^b - \mathbf{D}^w + \mathbf{F}^w)\mathbf{X}^T\mathbf{P}\} \\ &= 2tr\{\mathbf{P}^T\mathbf{X}\mathbf{U}\mathbf{X}^T\mathbf{P}\} \\ &= 2\sum_{i=1}^{d}\mathbf{P}_i^T\mathbf{X}\mathbf{U}\mathbf{X}^T\mathbf{P}\end{aligned}$$

where $\mathbf{U} = \mathbf{D}^b - \mathbf{F}^b - \mathbf{D}^w + \mathbf{F}^w$, according to the method for DNE algorithm, then we have the optimization problem as follows:

$$\begin{cases}\max\limits_{\mathbf{P}} \ tr(\mathbf{P}^T\mathbf{X}\mathbf{U}\mathbf{X}^T\mathbf{P}) \\ s.t. \qquad \mathbf{P}^T\mathbf{P} = \mathbf{I}\end{cases}$$

whose corresponding problem is

$$\max \ \sum_{i=1}^{d}\mathbf{p}^T{}_i\mathbf{X}\mathbf{U}\mathbf{X}^T\mathbf{p}_i = \sum_{i=1}^{d}\lambda_i \ .$$

Suppose the eigenvalues of $\mathbf{X}\mathbf{U}\mathbf{X}^T$ are $\lambda_1 \geq \cdots \geq \lambda_d$, we select the $r$ eigenvectors corresponding to the first $r$ eigenvalues to form the transformation matrix, or $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_r]$.

## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61373093 and 61033013, by the Natural Science Foundation of Jiangsu Province of China under Grant Nos. BK2011284 and BK201222725, by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No.13KJA520001, and by the Qing Lan Project.

## REFERENCES

[1] I. Joliffe. Principal Component Analysis. Springer, New York, 1986

[2] Martinez and A. Kak. "PCA versus LDA", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, 2001, pp.228-233.

[3] K. Fukunnaga, Introduction to Statistical Pattern Recognition, Academic Press, second edition, 1991.

[4] J. B. Tenenbaum, V. D. Silva, J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction", Science, 2000, vol. 290, pp. 2319-2323

[5] S. Roweis, L. Saul. "Nonlinear dimensionality reduction by locally linear embedding", Science, 2000, vol. 290, pp. 2323-2326

[6] M. Belkin, P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation", Neural Computation, 2003, vol. 15 pp. 1373-1396

[7] X. F. He, S. C. Yan, Y. X. Hu, P. Niyogi, H. J. Zhang, "Face recognition using Laplacian faces", IEEE Trans. Pattern Analysis and Machine Intelligence, 2005, Volume. 27, pp. 328-340.

[8] X. F. He, D. Cai, S. C. Yan and H. J. Zhang, "Neighborhood preserving embedding", In: Proceedings of IEEE International Conference on Computer Vision, 2005, vol. 2, pp.1208-1213

[9] S. C. Yan, D. Xu, B. Y. Zhang, H. J. Zhang, "Graph Embedding: A General Framework for Dimensionality Reduction", Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[10] S. C. Yan, D. Xu, B. Y. Zhang, H. J. Zhang, S. Lin, Q. Yang, "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction". IEEE Trans. Pattern Analysis and Machine Intelligence29, 2007, pp. 40-51

[11] Y. Q. Lu, C. Lu, M. Qi, S. Y. Wang, "A Supervised Locality Preserving Projections Based Local Matching Algorithm for Face Recognition", Advances in Computer Science and Information Technology, 2010, Volume. 6059, pp. 28-37

[12] Q. B. You, N. N. Zheng, S. Y. Du, Y. Wu, "Neighborhood discriminant projection for face recognition", Pattern Recognition, 2007, Volume. 28, pp. 1156-1163

[13] W. Zhang, X. Y. Xue, H. Lu, Y. F. Guo, "Discriminant neighborhood embedding for classification", Pattern Recognition, 2006, Volume. 39, pp. 2240-2243.

[14] J. P. Gou, Z. Yi, "Locality-Based Discriminant Neighborhood Embedding", The Computer Journal, 2013, Volume. 56, pp. 1063-1082

[15] W. Zhang, X. Y. Xue, Study on Feature Transformation Algorithm based on K-Nearest-neighbor Classification Rule, Fudan University, China,2007, pp.37-40.

[16] The Database of Faces, Available: http://www.cl.cam.ac.uk/research /dtg/attarchive/facedatabase.html,(accessed December 21, 2013)

[17] UCSD Computer Vision, Available: http://vision.ucsd.edu/content/yale-face-database, (accessed December 21, 2013)

[18] X. R. Li, T. Jiang, K. S. Zhang, "Efficient and robust feature extraction by maximum margin criterion", Neural Networks, 2006, Volume. 17, pp. 157-165

2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# Hidden Space Discriminant Neighborhood Embedding

Chuntao Ding
School of Computer Science and Technology & Provincial Key laboratory for Computer Information Processing, Soochow University, Suzhou 215006, Jiangsu, China
Email: 20124227036@suda.edu.cn

Li Zhang
School of Computer Science and Technology & Provincial Key laboratory for Computer Information Processing, Soochow University, Suzhou 215006, Jiangsu, China
Email: zhangliml@suda.edu.cn

Bangjun Wang
School of Computer Science and Technology & Provincial Key laboratory for Computer Information Processing, Soochow University, Suzhou 215006, Jiangsu, China
Email: wangbangjun@suda.edu.cn

***Abstract*—Discriminant neighborhood embedding (DNE) algorithm is one of supervised linear dimensionality reduction methods. Its nonlinear version kernel discriminant neighborhood embedding (KDNE) is expected to behave well on classification tasks. However, since KDNE constructs an adjacent graph in the original space, the adjacency graph could not represent the adjacent information in the kernel mapping space. By introducing hidden space, this paper proposes a novel nonlinear method for DNE, called hidden space discriminant neighborhood embedding (HDNE). This algorithm first maps the data in the original space into a high dimensional hidden space by a set of nonlinear hidden functions, and then builds an adjacent graph incorporating neighborhood information of the dataset in the hidden space. Finally, DNE is used to find a transformation matrix which would map the data in the hidden space to a low-dimensional subspace. The proposed method is applied to ORL face and MNIST handwritten digit databases. Experimental results show that the proposed method is efficiency for classification tasks.**

## I. Introduction

Dimensionality reduction methods have been attracted a lot of attention in machine learning, pattern recognition and computer vision etc. As one of important preprocessing steps in the analysis of high dimensional data, dimensionality reduction usually makes the data in a high dimensional space embed in a relatively low dimensional space, meanwhile, with most of the original data information preserved [1] [2]. Usually, dimensionality reduction methods can be divided into two groups, or linear and non-linear ones [4][5][6][7][8][9][10][11][12][13][14][15][16].

The most classic linear dimensionality reduction method is principal components analysis (PCA), of which the variance of data is used to measure useful information [20]. Typically, the larger the variance of data in some direction is, the more information this direction has; otherwise, the less information and value it has.

Since S. Roweis et al. proposed locally linear embedding (LLE) algorithm [3], manifold learning representing nonlinear dimensionality reduction methods quickly attracted attention of so many researchers. For the advantage of both linear dimensionality reduction and manifold learning, locality preserving projection (LPP), regarded as an upgrade version of PCA, was proposed [4]. As an unsupervised dimensionality reduction method, it could maximally keep the neighborhood structure of a high dimensional dataset. If points are close to each other in both the original space, they remain a relatively close distance after reducing dimensionality so as to preserve the local structure. Therefore, LPP is able to find a better projection direction for the data belonging to different classes with a far distance between each other. Some manifold learning methods, such as LLE, cannot yield a projection matrix, so they cannot perform incremental learning for new data. To cover this shortage, neighborhood preserving embedding (NPE) was proposed, which is a linear approximation of LLE and able to learn a projection matrix [5].

Usually, classification is a supervised learning with prior knowledge of class information. However, manifold dimensionality reduction methods discussed above are all unsupervised so that they cannot make full use of the prior knowledge. To remedy this, Zhang et al. proposed a supervised linear dimensionality reduction method, called discriminant neighborhood embedding (DNE) [16]. In DNE, if the points belonging to the same class are close to each other in original space, they would still remain a relatively close distance after reducing dimensionality. While if the points belonging to the different classes are close to each other in original space, they would remain a relatively far distance after reducing dimensionality. By introducing kernel tricks into DNE, a non-linear version called kernel DNE (KDNE) was proposed [25], where kernel function must satisfy Mercer's condition [17] and [18]. Nevertheless, being similar to DNE, KDNE only constructs the adjacent graph of original space without taking into account one of mapping space so that local geometric structure cannot be preserved efficiently when learning dimensionality reduction of the transition matrix.

Considering that DNE cannot get a better projection with linearly non-separable samples and KDNE cannot employ neighborhood relationships efficiently in a high dimensional space, we introduce the conception of hidden space. By using a nonlinear hidden function, the data in the original space are mapped into a high dimensional space. As a consequence, some linearly non-separable samples in a low dimensional space are now separable [19]. The novel method

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

is called hidden space discriminant neighborhood embedding (HDNE), which is also a nonlinear extension of DNE.

HDNE first maps the data in the original space into the the high dimensional hidden space in which the data would be linearly separable, and then builds its adjacent graph so that the local relationships for samples can be preserved in the hidden space. Reducing the dimensionality of samples in the hidden space by applying DNE can make the samples be linearly separable not only in the hidden space but also in the discriminant subspace. As a result, the recognition rate could be significantly improved. Experimental results on artificial and real-world datasets show that HDNE has higher recognition rates.

The remainder of the paper is organized as follows. In Section 2, we briefly review the DNE and KDNE. Section 3 presents HDNE. Simulation experiments are given in Section 4 and conclusions are provided in Section 5.

## II. Related Works

In this section, DNE method and KDNE method will be reviewed briefly.

### A. Discriminant Neighborhood Embedding

To exploit the class information efficiently, Zhang et al. proposed DNE which requires to build an adjacent graph between the samples in an original space and meanwhile tries to preserve the adjacent relationships in a low dimensional space. If the points belonging to the same class are close to each other in the high dimensional space, they would remain a relatively close distance in a discriminant subspace. If the points belonging to the different classes are close to each other in the high dimensional space, they would be separated in a discriminant subspace. Next, we will give a brief introduction of DNE algorithm.

Given a set of training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in R^m$, $y_i \in \{1, 2, ..., c\}$ is the label of $\mathbf{x}_i$, and $c$, $N$ and $m$ denote the number of classes, the number of samples and dimensionality, respectively. The goal of DNE is to find a projection matrix $\mathbf{A}$. If any two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ belonging to the same class are close, $\mathbf{v}_i = \mathbf{A}^T\mathbf{x}_i$ and $\mathbf{v}_j = \mathbf{A}^T\mathbf{x}_j$ are close, too. Of course, if they belong to different classes, the distance between them would become far after projection. The projection matrix is represented as $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_d] \in R^{m\times d}$, where $d < m$ and the vectors $\mathbf{a}_i \in R^m$ are independent of each other. The detail procedure of DNE is listed in Algorithm 1.

### B. Kernel Discriminant Neighborhood Embedding

Given $\mathbf{x}, \mathbf{z} \in X \subseteq R^m$ and nonlinear function $\Phi$, we can map $\mathbf{x}$ and $\mathbf{z}$ in the input space $X$ into a feature space $F$, where $F \subseteq R^M$ and $m \ll M$. According to the Mercer theorem, we have

$$k(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x})^T\Phi(\mathbf{z})$$

where $k(\mathbf{x}, \mathbf{z})$ denotes a Mercer kernel function which makes $M$-dimensional inner product operation in a high dimensional space change to be $m$-dimensional calculus of function in a low-dimensional space.

**Algorithm 1** DNE

Input: Training sample matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in R^{m\times N}$, and the dimensionality of discriminant subspace $d$;
Output: Projection matrix $\mathbf{A}$;

1). Construct the adjacent graph $\mathbf{F}$, which is defined as:

$$F_{ij} = \begin{cases} +1, & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors and } y_i = y_j \\ -1, & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors and } y_i \neq y_j \\ 0, & \text{otherwise} \end{cases}$$

2). Solve the following optimization problem:

$$\min_{\mathbf{A}} trace(\mathbf{A}^T\mathbf{X}(\mathbf{S}-\mathbf{F})\mathbf{X}^T\mathbf{A})$$
$$\text{s.t.} \quad \mathbf{a}_i{}^T\mathbf{a}_i = 1, \mathbf{a}_i{}^T\mathbf{a}_j = 0, i \neq j, i, j = 1, \ldots, d$$

where $\mathbf{S}$ is a diagonal matrix and its entries are $\mathbf{S}_{ii} = \sum_j \mathbf{F}_{ji}$.
The projection matrix $\mathbf{A}$ can be obtained by computing the eigenvalue problem of $\mathbf{X}(\mathbf{S}-\mathbf{F})\mathbf{X}^T\mathbf{A} = \lambda\mathbf{A}$.
Let eigenvalues be $\lambda_i$ and their corresponding eigenvectors be $\mathbf{a}_i$. Assume that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ .

3). Return $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_d]$

As a result, the problems such as curse of dimensionality are solved skillfully. Next, three common kernels are presented below [22]. Polynomial kernel is

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i{}^T\mathbf{x}_j + c)^p \tag{1}$$

where $p$ is parameter of this kernel. Gaussian kernel is

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/s^2) \tag{2}$$

where $s > 0$ is parameter of this kernel. Linear kernel is

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i{}^T\mathbf{x}_j \tag{3}$$

By introducing kernel tricks, DNE could be generalized to its nonlinear version, or KDNE. The goal of KDNE is also to find a projection matrix $\mathbf{A} \in R^{N\times d}$ which cannot be obtained explicitly. Fortunately, we could get samples in the discriminant subspace space by using an auxiliary matrix $\mathbf{B} \in R^{N\times d}$. Namely, $\mathbf{v}_j = \mathbf{B}^T\mathbf{K}_{\cdot j}$, where $\mathbf{K}$ is a kernel Gram matrix with $K_{ij} = k\left(\mathbf{x}_i, \mathbf{x}_j\right)$. The detail for KDNE is given in Algorithm 2.

**Algorithm 2** KDNE

Input: A training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, and the dimension of discriminant subspace $d$
Output: Auxiliary matrix $\mathbf{B}$;

1). Construct the adjacent graph $\mathbf{F}$ which is defined as:

$$F_{ij} = \begin{cases} +1, & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors and } y_i = y_j \\ -1, & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors and } y_i \neq y_j \\ 0, & \text{otherwise} \end{cases}$$

2). Solve the following optimization problem:

$$\min \quad trace(\mathbf{A}^T(\mathbf{S}-\mathbf{F})\mathbf{K}^T\mathbf{A})$$
$$\text{s.t.} \quad \mathbf{a}_i{}^T\mathbf{a}_i = 1, \mathbf{a}_i{}^T\mathbf{a}_j = 0, i \neq j, i, j = 1, \ldots, d$$

where $\mathbf{S}$ is a diagonal matrix and its entries are $\mathbf{S}_{ii} = \sum_j \mathbf{F}_{ji}$.
The auxiliary matrix $\mathbf{B}$ can be obtained by computing the eigenvalue problem of $(\mathbf{S}-\mathbf{F})\mathbf{K}^T\mathbf{B} = \lambda\mathbf{B}$.
Let eigenvalues be $\lambda_i$ and their corresponding eigenvectors be $\mathbf{b}_i$. Assume that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ .

3). Return $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_d]$

## III. Hidden Space Discriminant Neighborhood Embedding

DNE is a linear feature transform so that it does not work well for the linearly non-separable data. Although the nonlinear version of DNE, or KDNE, has been proposed, its adjacent graph is still construed in the original space. Usually, the local relationship between samples in the original

space cannot be guaranteed in the high-dimensional space obtained by nonlinear mapping. Taking into account these shortcomings in DNE and KDNE, we propose a hidden space discriminant neighborhood embedding method.

### A. Hidden Space

Hidden space is derived from neural networks, and is introduced to support vector machines (SVMs) in [19]. Generally, SVMs require the Mercer kernel functions. However, Nonlinear hidden functions could be any kernel ones. Some learning algorithms have been extended into the hidden space such as PCA [23] and LDA [24].

With the help of some nonlinear hidden function, data being linearly non-separable in the original space can be mapped into a high-dimensional space in which data are now linearly separable. Given $N$ samples $\{\mathbf{x}_i\}_{i=1}^N$, we map them into a hidden space by using a hidden function $\varphi(\mathbf{x})$. Let $\mathbf{z} = \varphi(\mathbf{x})$, where $\mathbf{z}$ is the image of $\mathbf{x}$. We take kernel functions as hidden functions, and we have

$$\mathbf{z} = [k(\mathbf{x}, \mathbf{x}_1), ..., k(\mathbf{x}, \mathbf{x}_N)]^T \tag{4}$$

In hidden functions, we require only the symmetry for kernel functions instead of Mercer's condition. In addition, we can obtain the mapped samples $\mathbf{z}$. So, it is very convenient to calculate statistics for samples.

### B. HDNE

In a classification task, assume that the set of labeled training samples is $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i \in R^m$, and $y_i \in \{1, 2, ..., c\}$. By employing the hidden function (4), the images of $\mathbf{x}_i$ in the hidden space are

$$\mathbf{z}_i = [k(\mathbf{x}_i, \mathbf{x}_1), ..., k(\mathbf{x}_i, \mathbf{x}_N)]^T$$

In the hidden space, the training set can be represented as $\{\mathbf{z}_i, y_i\}_{i=1}^N$ where $\mathbf{z}_i \in R^N$.

Since the concrete form of samples has been known in the hidden space, so we are able to directly build the adjacent graph $\mathbf{F}$ in this space. The entries of $i$th row and $j$th column in $\mathbf{F}$ is

$$F_{ij} = \begin{cases} +1, & \mathbf{z}_i \textit{ and } \mathbf{z}_j \textit{ are neighbors and } y_i = y_j \\ -1, & \mathbf{z}_i \textit{ and } \mathbf{z}_j \textit{ are neighbors and } y_i \neq y_j \\ 0, & \textit{otherwise} \end{cases} \tag{5}$$

Recall that the dimensionality of hidden space may be very high, so it needs to reduce dimensionality for efficient computation. Let the transformation matrix be $\mathbf{P} \in R^{N \times d}$, where $d$ denotes the dimensionality of discriminant subspace. In the discriminant subspace, the sample $\mathbf{z}_i \in R^m$ is transformed to be $\mathbf{P}^T\mathbf{z}_i \in R^d$.

Let $\phi(\mathbf{P})$ and $\varphi(\mathbf{P})$ be the within-class and the between-class neighborhood scatters, respectively. The within-class neighborhood scatter $\phi(\mathbf{P})$ is defined as

$$\phi(\mathbf{P}) = \sum_{i,j,y_i=y_j} ||\mathbf{P}^T\mathbf{z}_i - \mathbf{P}^T\mathbf{z}_j||^2 \tag{6}$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ are neighbors and belong to the same class. The between-class neighborhood scatter $\varphi(\mathbf{P})$ is defined as

$$\varphi(\mathbf{P}) = \sum_{i,j,y_i \neq y_j} ||\mathbf{P}^T\mathbf{z}_i - \mathbf{P}^T\mathbf{z}_j||^2 \tag{7}$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ are still neighbors but belong to the different classes. We hope that if neighbors belong to the same class they should be close to each other in the discriminant subspace; otherwise, be far away from each other. We can implement our demand by minimizing the $\phi(\mathbf{P})$ and maximizing $\varphi(\mathbf{P})$ at the same time, which could be described as

$$\min_{\mathbf{P}} \quad \Delta(\mathbf{P}) = \phi(\mathbf{P}) - \varphi(\mathbf{P}) \tag{8}$$

Substituting (5) into the expression of $\Delta(\mathbf{P})$, we can rewritten it as follows:

$$\begin{aligned} \Delta(\mathbf{P}) &= \sum_{i,j=1}^{N} ||\mathbf{P}^T\mathbf{z}_i - \mathbf{P}^T\mathbf{z}_j||^2 F_{ij} \\ &= 2\sum_{i,j=1}^{N} (\mathbf{z}_i{}^T\mathbf{P}\mathbf{P}^T\mathbf{z}_i - \mathbf{z}_i{}^T\mathbf{P}\mathbf{P}^T\mathbf{x}_j)F_{ij} \\ &= 2\sum_{i,j=1}^{N} tr((\mathbf{P}^T\mathbf{z}_i\mathbf{z}_i{}^T\mathbf{P} - \mathbf{P}^T\mathbf{z}_j\mathbf{z}_i{}^T\mathbf{P})F_{ij}) \\ &= 2tr(\sum_{i,j=1}^{N} (\mathbf{P}^T\mathbf{z}_iF_{ij}\mathbf{z}_i{}^T\mathbf{P} - \mathbf{P}^T\mathbf{z}_jF_{ij}\mathbf{z}_i{}^T\mathbf{P})) \\ &= 2tr(\mathbf{P}^T\mathbf{Z}\mathbf{S}\mathbf{Z}\mathbf{P} - \mathbf{P}^T\mathbf{Z}\mathbf{F}\mathbf{Z}\mathbf{P}) \\ &= 2tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P}) \end{aligned} \tag{9}$$

where $tr(\cdot)$ denotes the trace of a matrix, and $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N]$, and $\mathbf{S}$ is a diagonal matrix with $S_{ii} = \sum_{j=1}^N F_{ij}$. As a result, the problem (8) can be rewritten as

$$\begin{aligned} \min_{\mathbf{P}} \quad & tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P}) \\ \text{s.t.} \quad & \mathbf{P}^T\mathbf{P} = \mathbf{I} \end{aligned} \tag{10}$$

and where $\mathbf{I}$ is the identity matrix.

In the following, we introduce a lemma in [25] and give a theorem which describes the solution to (10).

**Lemma 1**: Suppose $\mathbf{A} \in R^{N \times N}$ is a real symmetric matrix and its minimum eigenvalue is $\lambda_1$. The solution to the minimization problem of $\boldsymbol{\eta}^T\mathbf{A}\boldsymbol{\eta}$ which subjects to $\boldsymbol{\eta}^T\boldsymbol{\eta} = 1$ and $\boldsymbol{\eta} \in R^N$ is the eigenvector corresponding to the eigenvalue $\lambda_1$.

Theorem 2: Assume that the eigenvalues of the matrix $\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T$ are $\lambda_1 \leq ... \leq \lambda_{i-1} \leq \lambda_i \leq ... \leq \lambda_N$, and $\boldsymbol{\xi}_i$ is the corresponding eigenvector of eigenvalue $\lambda_i$. Then optimal $\mathbf{P}$ to the minimization problem $tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P})$ is the corresponding eigenvectors of the first $d$ eigenvalues. Namely, $\mathbf{P} = [\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_d]$.

**Proof**: Since $(\mathbf{S} - \mathbf{F})$ is a real symmetric matrix, $\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T$ is also a real symmetric matrix. According to Lemma 1, if $d = 1$ , only when $\mathbf{P}$ is the eigenvector corresponding to the minimum eigenvalue $\lambda_1$ of matrix $\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T$ (namely $\mathbf{P} = \boldsymbol{\xi}_1$) is $\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P}$ minimum. Right now, $\lambda_1$ is the optimal value of $tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P})$.

Similarly, if let $\mathbf{P}$ represent eigenvectors corresponding to the first $d$ minimum eigenvalues (namely $\mathbf{P} = [\xi_1, \cdots, \xi_d]$), then we have $tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P}) = \sum_{i=1}^{d} \lambda_i$. Right now,

$tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S}-\mathbf{F})\mathbf{Z}^T\mathbf{P})$ achieves its optimal value. This completes the proof.

Theorem 2 shows that minimizing $tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S}-\mathbf{F})\mathbf{Z}^T\mathbf{P})$ is equivalent to eigendecompose on the matrix $\mathbf{Z}(\mathbf{S}-\mathbf{F})\mathbf{Z}^T$. If the projection matrix is composed of the eigenvectors corresponding to the first $d$ eigenvalues, the value of object function with respect to optimization problem (11) is minimum.

### C. Comparison of DNE, KDNE and HDNE

By constructing the adjacent graph for samples in the original space, DNE makes their local structure be preserved in a discriminant subspace. In addition, the problem of finding the projection matrix is equivalent eigendecompose $\mathbf{X}(\mathbf{S}-\mathbf{F})\mathbf{X}^T$.

Nevertheless, DNE is linear so that it cannot be applied to the linearly non-separable data in the original space. As a remedy for this drawback, KDNE makes DNE extend to be nonlinear, but it still utilizes the adjacent graph constructed in the original space. For KDNE, the matrix that needs eigendecomposition is $(\mathbf{S}-\mathbf{F})\mathbf{K}^T$ .

The method proposed here is first to map the data in the original space into into the hidden space and then to construct the adjacent graph in this space. The local structure of samples in the hidden space can be preserved when performing dimensionality reduction. Thus, HDNE remedies the drawbacks that DNE is not fit for nonlinear problems and KDNE cannot preserve the local structure of high-dimensional space. HDNE tries to eigendecompose $\mathbf{Z}(\mathbf{S}-\mathbf{F})\mathbf{Z}^T$ .

From the above, the three methods are all based on eigendecomposition of some matrix, and then obtain the projection matrix composed of the eigenvectors corresponding to the first $d$ minimum eigenvalues.

## IV. Simulation Experiments

In this section, to validate the efficiency of HDNE, we compare it with other methods, including PCA, LDA, LPP, NPE, DNE, KFDA and KDNE on image classification problems. Here, we consider two kinds of images: face and handwritten digit.

For KDNE, KFDA and HDNE, Gaussian kernel $k(\mathbf{x},\mathbf{x}') = \exp\{-p\|\mathbf{x}-\mathbf{x}'\|\}$ with the kernel parameter $p>0$ is used. This parameter $p$ is selected by using 5-fold cross validation.

### A. Face Recognition

Consider the widely studied ORL Face dataset (from the website: http://archive.ics.uci.edu/ml/datasets.html),which is created by the University of Cambridge and has a total of 400 face images with different illumination intensity and facial expression etc., each 112 x 92, from 40 persons and 10 for each one. It also gives considerations to race, gender and facial expression and is a frequently-used face dataset.

In the face recognition experiment, we mainly focus on the effect of the dimensionality of discriminant subspace on recognition rates under different choices for $K$, where $K$ is the parameter of the nearest neighbor (NN) classifier. Thus, without prior knowledge, $K$ is set to be 1, 3 and 5 respectively. In the experiment, we randomly select 5 samples from the same person for training and the rest are for test. There are 200 training and test samples, respectively. All samples are divided by 255 to implement normalization.

For the high-dimensional original images, we first utilize PCA to reduce dimensionality from 10,304 to 100. In doing so, there are two benefits. On the one hand, computations are greatly reduced. On the other hand, the majority of noises are diminished. We repeat our experiment 100 trials and report the average result on test sets.

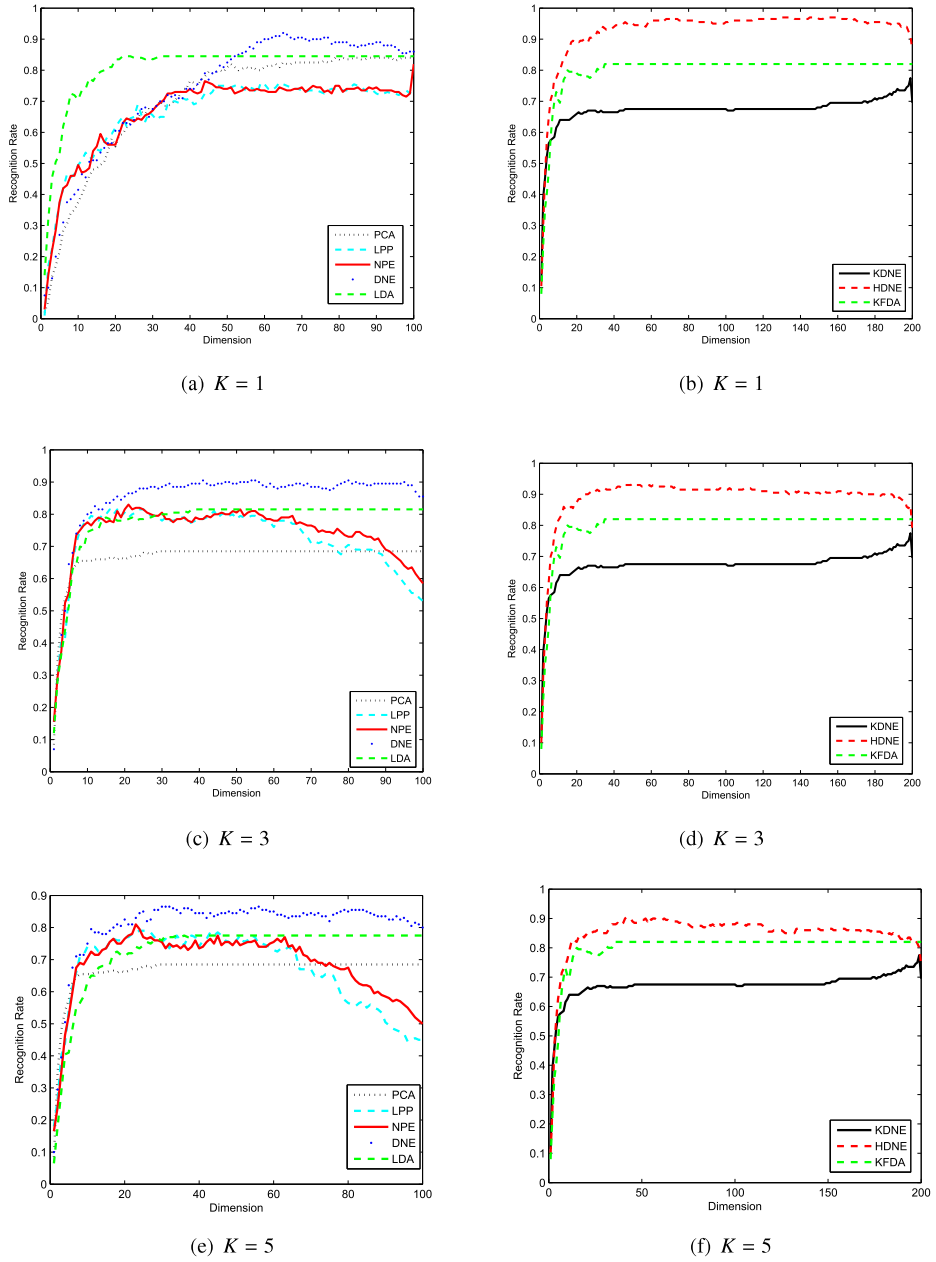
For PCA, LDA, LPP, NPE and DNE, their dimensionalities of discriminant subspace and recognition rates are plotted in one figure because their maximal dimensionalities are all 100 after dimensionality reduction. While for KDNE, KFDA and HDNE, their results are plotted in another one figure because they have the same maximal dimensionality of $N$.

When different $K$ value is selected, Fig. 1 presents the corresponding performance along with the change of dimensionality for these methods. From Figs. 1, we can know that, for all the methods, at the beginning the performance improves all the time along with increasing dimensionality, and then it tends to be invariable or decreasing. From Figs. 1(a), 1(c) and 1(e), we can know that with different K values, DNE method is always able to reach a maximum, being better than PCA, LDA, LPP and NPE, in different discriminant subspace. As the nonlinear version of DNE, from Fig. 1(b), 1(d) and 1(f), KDNE,KFDA and HDNE have great change in recognition rate along with the change of dimensionality. Obviously, HDNE works better than KDNE and KFDA no matter which $K$ is selected in our experiment.

Table I. Performance comparisons on ORL dataset ($K$ = 1)

| Method | Dimensional of subspace | Recognition rate |
|---|---|---|
| PCA | 79 | 84.00 |
| LDA | 22 | 84.50 |
| LPP | 59 | 76.00 |
| NPE | 100 | 82.00 |
| DNE | 65 | 92.00 |
| KDNE | 194 | 79.50 |
| KFDA | 35 | 82.00 |
| HDNE | 71 | 96.50 |

From Fig. 1, we can see that all methods have better performance when $K$ = 1. The larger $K$ does not mean better since ORL face data are insufficient. Table 1 provides the best recognition rates obtained by all methods and the corresponding dimensionality of discriminant subspace for $K$ = 1. Compared with KDNE, HDNE always has a better recognition rate and a lower dimensionality of discriminant subspace. As a result, our view that although being also a nonlinear extension of DNE, KDNE does not employ the adjacent graph to preserve the local structure so that lead to an unsatisfied recognition rate is verified. By contrast, HDNE method achieves this, which makes the following view be persuasive: compared with the practice that use kernel as nonlinear extension, this method that let the data map into high-dimensional space, and then construct adjacent graph to preserve the relationships between neighbors can work better and get a higher recognition rate in discriminant subspace.




(a) $K = 1$ (b) $K = 1$

(c) $K = 3$ (d) $K = 3$

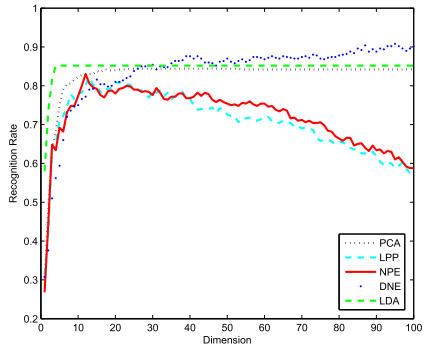
(e) $K = 5$ (f) $K = 5$

Figure 1. Recognition vs. dimensionality on ORL face dataset

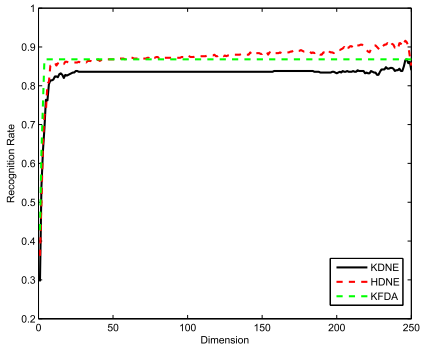
### B. Handwritten Digit Recognition

Consider the MNIST dataset *(from website http://archive.ics.uci.edu/ml/datasets.html)*, which has a total of 60,000 training and 10,000 test images with total 10 classes. Five classes, including digits 1, 3, 7, 8 and 9, are selected. For each class, we randomly select 50, 100 and 150 samples from the original training set as our training set, 100 samples from the original test set as our test set. In this experiment, PCA is stilled utilized to preprocess in order to obtain the *100-dimensional* data, and nearest neighbor is selected as the classifier.

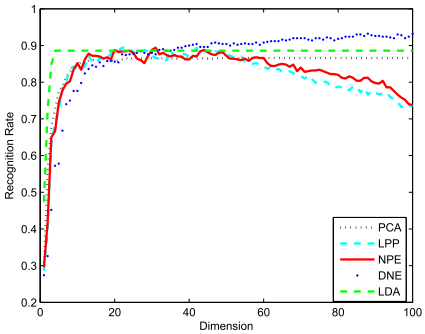
In this experiment, we will mainly perform analysis on the effect of the number of samples on the dimensionality of discriminant subspace and recognition rates. Fig. 2 clearly shows that the larger the number of samples is, the higher recognition rate is under condition of the same dimensionality for discriminant subspace. For Figs. 2(a), 2(c) and 2(e), the statuses of recognition rates for four methods are clearly presented along with the change of dimensionality of discriminant subspace. With the increase of dimensionality, the recognition rate of each method improves on the whole. Figs. 2(b), 2(d) and 2(f) respectively describe the recognition rates of KDNE and HDNE with respect to the dimensionality of discriminant subspace under condition of the same number of samples. We have the conclusion that with the same number of samples and the same dimensionality of discriminant subspace, as to recognition rate, HDNE obviously works better than KDNE and KFDA.



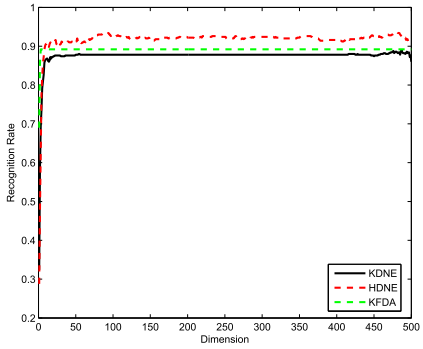

(a) 50 training samples



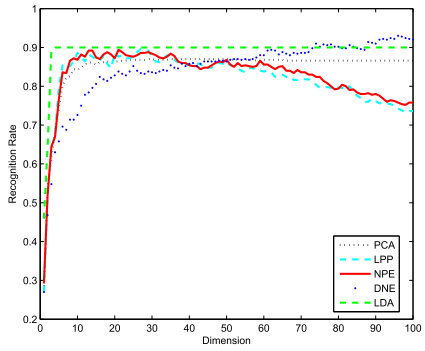

(b) 50 training samples



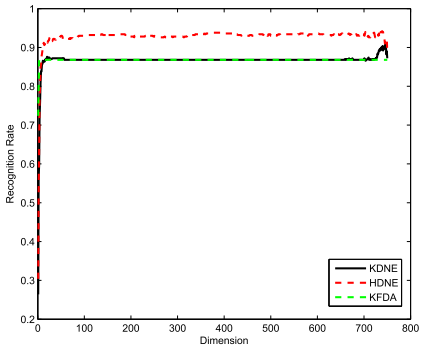

(c) 100 training samples




(d) 100 training samples




(e) 150 training samples




(f) 150 training samples

Figure 2. Recognition vs. dimensionality on MNIST dataset

Table II. PERFORMANCE COMPARISONS ON THE MNIST DATASET (50 *trainingsamples*)

| Method | Dimensional of subspace | Recognition rate |
|---|---|---|
| PCA | 24 | 84.40 |
| LPP | 14 | 82.20 |
| NPE | 12 | 83.00 |
| LDA | 4 | 86.40 |
| DNE | 95 | 90.10 |
| KDNE | 247 | 86.80 |
| KFDA | 3 | 86.86 |
| HDNE | 244 | 91.80 |

Table III. PERFORMANCE COMPARISONS ON THE MNIST DATASET (100 *trainingsamples*)

| Method | Dimensional of subspace | Recognition rate |
|---|---|---|
| PCA | 26 | 86.80 |
| LPP | 22 | 89.20 |
| NPE | 31 | 89.40 |
| LDA | 4 | 87.80 |
| DNE | 88 | 91.60 |
| KDNE | 487 | 89.00 |
| KFDA | 3 | 89.20 |
| HDNE | 480 | 93.80 |

Tables 2-4 show the optimal recognition rates for each method in the whole discriminant subspace with the certain number of samples, from which the conclusion is that recognition rate of HDNE is higher than the other methods. As a nonlinear extension of DNE as well, HDNE also has a higher recognition rate than KFDA and KDNE.

Table IV. PERFORMANCE COMPARISONS ON THE MNIST DATASET (150 *trainingsamples*)

| Method | Dimensional of subspace | Recognition rate |
|---|---|---|
| PCA | 29 | 87.00 |
| LPP | 27 | 89.40 |
| NPE | 21 | 89.40 |
| LDA | 4 | 90.00 |
| DNE | 96 | 91.80 |
| KDNE | 746 | 90.60 |
| KFDA | 3 | 86.98 |
| HDNE | 739 | 94.40 |

## V. CONCLUSIONS

HDNE is proposed by introducing hidden functions, which is a nonlinear extension of DNE. Specifically, it performs analysis on the preserved local structure in hidden space. The data being linearly non-separable in original space are linearly separable in hidden space and at the same time the adjacent graph is constructed to preserve the local structure of data. From experimental results on ORL dataset with different K values in K-nearest neighbor and on MNIST dataset with the different number of samples, we have the conclusion that HDNE has a better performance than the other methods.

## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61373093 and 61033013, by the Natural Science Foundation of Jiangsu Province of China under Grant Nos. BK2011284 and BK201222725, by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No.13KJA520001,and by the Qing Lan Project.

## REFERENCES

[1] J. B. Tenenbaum, V. de Silva and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290, 2319-2323, 2000.

[2] S. Roweis and L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, 290, pp. 2323-2326, 2000.

[3] S. Roweis and L. K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, vol. 290, 22 December, 2000.

[4] X. F. He and P. Niyogi, Locality Preserving Projections, *Advances in Neural Information Processing Systems 16* Vancouver, British Columbia, Canada, 2003.

[5] X. F. He, D. Cai, S. C. Yan, eta, Neighborhood Preserving Embedding, *Proceeding of the IEEE International Conference on Computer Vision.* Beijing, China, pp, 1208-1213, 2005

[6] H. T. Chen, H. W. Chang and T. L. Liu, Local discriminant embedding and its variants, *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2005.

[7] M. Belkin and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation*, vol. 15, pp. 1373-1396, 2003.

[8] X. He, S. Yan, Y. Hu and H. J. Zhang, Learning a Locality Preserving Subspace for Visual Recognition, *Proc. 9th International Conference on Computer Vision*, France, 2003.

[9] K. C. Lee, J. Ho, M. H. Yang and D. Kriegman. Video-Based Face Recognition Using Probabilistic Appearance Manifolds, *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, 2003.

[10] B. Scholkopf, A. Smola and K. R. Muller, Nonlinear Component Analysis as a Kernel Eigenvalue Problem, *Neural Computation*, *10* (5)*:* 1299-1319, 1998.

[11] M. Sugiyama, Local Fisher Discriminant Analysis for Supervised Dimensionality Reduction, *Proc of the 23rd International Conference on Machine Learning*,Pittsburgh, USA, pp, 905-912. 2006.

[12] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, PAMI, vol. 19, no. 7, pp. 711-720, July. 1997.

[13] M. Belkin and P. Niyogi, Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, NIPS 14, 2001.

[14] C. Hu, Chang Y, R. Feris and M. Turk, Manifold Based Analysis of Facial Expression, *IEEE Workshop on Face Processing in Video*,2004.

[15] M. H. Yang, Kernel Eigenfaces vs. Kernel Fisherfaces : Face Recognition Using Kernel Methods, AFGR, pp. 205-211, 2002.

[16] W. Zhang, X. Y. Xue, H. Lu and Y. F. Guo, Discriminant Neighborhood Embedding for Classification, *Pattern Recognition*, *39* (11)*:* 2240-2243, Nov. 2006.

[17] S. Saitoh, Theory of Reproducing Kernels and Its Applications, *UK* : *Longman Scienti fic and Technical*, 2004.

[18] J. Shawe−Taylor and N. Cristianini. Kernel Methods for Pattern Analysis, *Cambridge University Press*, 2004.

[19] L. Zhang, W. D. Zhou and L. C. Jiao, Hidden Space Support Vector Machines IEEE Trans, vol. 15, no. 6, 2004.

[20] K. Fukunaga, *Introduction to Statistical Pattern Recognition,* 2nd ed. New York : Academic, 1991.

[21] K. Mtiller, S. Mika, G. Riitsch, K. Tsuda and B. Sch61kopf, An Introduction to kernel-based learning algorithms, *IEEE Transactions on Neural Networks*, Vol. 12, pp. 181−201, 2001.

[22] D. M. J. Tax and R. P. W. Duin, Support vector data description, Mach. Learn. 54 (2004) 45-66

[23] W. D. Zhou, L. Zhang and L. C. Zhang. Hidden space principal component analysis 10*th* Pacific, *Asia Conference on Knowledge Discovery and Data Mining*, LNAI 3981, 801 − 805, 2006.

[24] L. Zhang, W. D. Zhou and P. C. Chang. Generalized Nonlinear Discriminant Analysis and Its Small Sample Size Problems Neurocomputing, 74, 568 − 574, 2011.

[25] W. Zhang and X. Y. Xue. *Study on Feature Transformation Algorithm based on K − Nearest − Neighbor Classification Rule,* Fudan University, China, 37-40, 2007.

2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# Hidden Space Discriminant Neighborhood Embedding

Chuntao Ding
School of Computer Science and Technology & Provincial Key laboratory for Computer Information Processing, Soochow University, Suzhou 215006, Jiangsu, China
Email: 20124227036@suda.edu.cn

Li Zhang
School of Computer Science and Technology & Provincial Key laboratory for Computer Information Processing, Soochow University, Suzhou 215006, Jiangsu, China
Email: zhangliml@suda.edu.cn

Bangjun Wang
School of Computer Science and Technology & Provincial Key laboratory for Computer Information Processing, Soochow University, Suzhou 215006, Jiangsu, China
Email: wangbangjun@suda.edu.cn

***Abstract*—Discriminant neighborhood embedding (DNE) algorithm is one of supervised linear dimensionality reduction methods. Its nonlinear version kernel discriminant neighborhood embedding (KDNE) is expected to behave well on classification tasks. However, since KDNE constructs an adjacent graph in the original space, the adjacency graph could not represent the adjacent information in the kernel mapping space. By introducing hidden space, this paper proposes a novel nonlinear method for DNE, called hidden space discriminant neighborhood embedding (HDNE). This algorithm first maps the data in the original space into a high dimensional hidden space by a set of nonlinear hidden functions, and then builds an adjacent graph incorporating neighborhood information of the dataset in the hidden space. Finally, DNE is used to find a transformation matrix which would map the data in the hidden space to a low-dimensional subspace. The proposed method is applied to ORL face and MNIST handwritten digit databases. Experimental results show that the proposed method is efficiency for classification tasks.**

## I. Introduction

Dimensionality reduction methods have been attracted a lot of attention in machine learning, pattern recognition and computer vision etc. As one of important preprocessing steps in the analysis of high dimensional data, dimensionality reduction usually makes the data in a high dimensional space embed in a relatively low dimensional space, meanwhile, with most of the original data information preserved [1] [2]. Usually, dimensionality reduction methods can be divided into two groups, or linear and non-linear ones [4][5][6][7][8][9][10][11][12][13][14][15][16].

The most classic linear dimensionality reduction method is principal components analysis (PCA), of which the variance of data is used to measure useful information [20]. Typically, the larger the variance of data in some direction is, the more information this direction has; otherwise, the less information and value it has.

Since S. Roweis et al. proposed locally linear embedding (LLE) algorithm [3], manifold learning representing nonlinear dimensionality reduction methods quickly attracted attention of so many researchers. For the advantage of both linear dimensionality reduction and manifold learning, locality preserving projection (LPP), regarded as an upgrade version of PCA, was proposed [4]. As an unsupervised dimensionality reduction method, it could maximally keep the neighborhood structure of a high dimensional dataset. If points are close to each other in both the original space, they remain a relatively close distance after reducing dimensionality so as to preserve the local structure. Therefore, LPP is able to find a better projection direction for the data belonging to different classes with a far distance between each other. Some manifold learning methods, such as LLE, cannot yield a projection matrix, so they cannot perform incremental learning for new data. To cover this shortage, neighborhood preserving embedding (NPE) was proposed, which is a linear approximation of LLE and able to learn a projection matrix [5].

Usually, classification is a supervised learning with prior knowledge of class information. However, manifold dimensionality reduction methods discussed above are all unsupervised so that they cannot make full use of the prior knowledge. To remedy this, Zhang et al. proposed a supervised linear dimensionality reduction method, called discriminant neighborhood embedding (DNE) [16]. In DNE, if the points belonging to the same class are close to each other in original space, they would still remain a relatively close distance after reducing dimensionality. While if the points belonging to the different classes are close to each other in original space, they would remain a relatively far distance after reducing dimensionality. By introducing kernel tricks into DNE, a non-linear version called kernel DNE (KDNE) was proposed [25], where kernel function must satisfy Mercer's condition [17] and [18]. Nevertheless, being similar to DNE, KDNE only constructs the adjacent graph of original space without taking into account one of mapping space so that local geometric structure cannot be preserved efficiently when learning dimensionality reduction of the transition matrix.

Considering that DNE cannot get a better projection with linearly non-separable samples and KDNE cannot employ neighborhood relationships efficiently in a high dimensional space, we introduce the conception of hidden space. By using a nonlinear hidden function, the data in the original space are mapped into a high dimensional space. As a consequence, some linearly non-separable samples in a low dimensional space are now separable [19]. The novel method

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

is called hidden space discriminant neighborhood embedding (HDNE), which is also a nonlinear extension of DNE.

HDNE first maps the data in the original space into the the high dimensional hidden space in which the data would be linearly separable, and then builds its adjacent graph so that the local relationships for samples can be preserved in the hidden space. Reducing the dimensionality of samples in the hidden space by applying DNE can make the samples be linearly separable not only in the hidden space but also in the discriminant subspace. As a result, the recognition rate could be significantly improved. Experimental results on artificial and real-world datasets show that HDNE has higher recognition rates.

The remainder of the paper is organized as follows. In Section 2, we briefly review the DNE and KDNE. Section 3 presents HDNE. Simulation experiments are given in Section 4 and conclusions are provided in Section 5.

## II. Related Works

In this section, DNE method and KDNE method will be reviewed briefly.

### A. Discriminant Neighborhood Embedding

To exploit the class information efficiently, Zhang et al. proposed DNE which requires to build an adjacent graph between the samples in an original space and meanwhile tries to preserve the adjacent relationships in a low dimensional space. If the points belonging to the same class are close to each other in the high dimensional space, they would remain a relatively close distance in a discriminant subspace. If the points belonging to the different classes are close to each other in the high dimensional space, they would be separated in a discriminant subspace. Next, we will give a brief introduction of DNE algorithm.

Given a set of training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in R^m$, $y_i \in \{1, 2, ..., c\}$ is the label of $\mathbf{x}_i$, and $c$, $N$ and $m$ denote the number of classes, the number of samples and dimensionality, respectively. The goal of DNE is to find a projection matrix $\mathbf{A}$. If any two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ belonging to the same class are close, $\mathbf{v}_i = \mathbf{A}^T\mathbf{x}_i$ and $\mathbf{v}_j = \mathbf{A}^T\mathbf{x}_j$ are close, too. Of course, if they belong to different classes, the distance between them would become far after projection. The projection matrix is represented as $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_d] \in R^{m\times d}$, where $d < m$ and the vectors $\mathbf{a}_i \in R^m$ are independent of each other. The detail procedure of DNE is listed in Algorithm 1.

### B. Kernel Discriminant Neighborhood Embedding

Given $\mathbf{x}, \mathbf{z} \in X \subseteq R^m$ and nonlinear function $\Phi$, we can map $\mathbf{x}$ and $\mathbf{z}$ in the input space $X$ into a feature space $F$, where $F \subseteq R^M$ and $m \ll M$. According to the Mercer theorem, we have

$$k(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x})^T\Phi(\mathbf{z})$$

where $k(\mathbf{x}, \mathbf{z})$ denotes a Mercer kernel function which makes $M$-dimensional inner product operation in a high dimensional space change to be $m$-dimensional calculus of function in a low-dimensional space.

**Algorithm 1** DNE

Input: Training sample matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in R^{m\times N}$, and the dimensionality of discriminant subspace $d$;
Output: Projection matrix $\mathbf{A}$;

1). Construct the adjacent graph $\mathbf{F}$, which is defined as:

$$F_{ij} = \begin{cases} +1, & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors and } y_i = y_j \\ -1, & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors and } y_i \neq y_j \\ 0, & \text{otherwise} \end{cases}$$

2). Solve the following optimization problem:

$$\min_{\mathbf{A}} trace(\mathbf{A}^T\mathbf{X}(\mathbf{S}-\mathbf{F})\mathbf{X}^T\mathbf{A})$$
$$\text{s.t.} \quad \mathbf{a}_i{}^T\mathbf{a}_i = 1, \mathbf{a}_i{}^T\mathbf{a}_j = 0, i \neq j, i, j = 1, \ldots, d$$

where $\mathbf{S}$ is a diagonal matrix and its entries are $\mathbf{S}_{ii} = \sum_j \mathbf{F}_{ji}$.
The projection matrix $\mathbf{A}$ can be obtained by computing the eigenvalue problem of $\mathbf{X}(\mathbf{S}-\mathbf{F})\mathbf{X}^T\mathbf{A} = \lambda\mathbf{A}$.
Let eigenvalues be $\lambda_i$ and their corresponding eigenvectors be $\mathbf{a}_i$. Assume that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ .

3). Return $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_d]$

As a result, the problems such as curse of dimensionality are solved skillfully. Next, three common kernels are presented below [22]. Polynomial kernel is

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i{}^T\mathbf{x}_j + c)^p \tag{1}$$

where $p$ is parameter of this kernel. Gaussian kernel is

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/s^2) \tag{2}$$

where $s > 0$ is parameter of this kernel. Linear kernel is

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i{}^T\mathbf{x}_j \tag{3}$$

By introducing kernel tricks, DNE could be generalized to its nonlinear version, or KDNE. The goal of KDNE is also to find a projection matrix $\mathbf{A} \in R^{N\times d}$ which cannot be obtained explicitly. Fortunately, we could get samples in the discriminant subspace space by using an auxiliary matrix $\mathbf{B} \in R^{N\times d}$. Namely, $\mathbf{v}_j = \mathbf{B}^T\mathbf{K}_{\cdot j}$, where $\mathbf{K}$ is a kernel Gram matrix with $K_{ij} = k\left(\mathbf{x}_i, \mathbf{x}_j\right)$. The detail for KDNE is given in Algorithm 2.

**Algorithm 2** KDNE

Input: A training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, and the dimension of discriminant subspace $d$
Output: Auxiliary matrix $\mathbf{B}$;

1). Construct the adjacent graph $\mathbf{F}$ which is defined as:

$$F_{ij} = \begin{cases} +1, & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors and } y_i = y_j \\ -1, & \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are neighbors and } y_i \neq y_j \\ 0, & \text{otherwise} \end{cases}$$

2). Solve the following optimization problem:

$$\min \quad trace(\mathbf{A}^T(\mathbf{S}-\mathbf{F})\mathbf{K}^T\mathbf{A})$$
$$\text{s.t.} \quad \mathbf{a}_i{}^T\mathbf{a}_i = 1, \mathbf{a}_i{}^T\mathbf{a}_j = 0, i \neq j, i, j = 1, \ldots, d$$

where $\mathbf{S}$ is a diagonal matrix and its entries are $\mathbf{S}_{ii} = \sum_j \mathbf{F}_{ji}$.
The auxiliary matrix $\mathbf{B}$ can be obtained by computing the eigenvalue problem of $(\mathbf{S}-\mathbf{F})\mathbf{K}^T\mathbf{B} = \lambda\mathbf{B}$.
Let eigenvalues be $\lambda_i$ and their corresponding eigenvectors be $\mathbf{b}_i$. Assume that $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m$ .

3). Return $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_d]$

## III. Hidden Space Discriminant Neighborhood Embedding

DNE is a linear feature transform so that it does not work well for the linearly non-separable data. Although the nonlinear version of DNE, or KDNE, has been proposed, its adjacent graph is still construed in the original space. Usually, the local relationship between samples in the original

space cannot be guaranteed in the high-dimensional space obtained by nonlinear mapping. Taking into account these shortcomings in DNE and KDNE, we propose a hidden space discriminant neighborhood embedding method.

### A. Hidden Space

Hidden space is derived from neural networks, and is introduced to support vector machines (SVMs) in [19]. Generally, SVMs require the Mercer kernel functions. However, Nonlinear hidden functions could be any kernel ones. Some learning algorithms have been extended into the hidden space such as PCA [23] and LDA [24].

With the help of some nonlinear hidden function, data being linearly non-separable in the original space can be mapped into a high-dimensional space in which data are now linearly separable. Given $N$ samples $\{\mathbf{x}_i\}_{i=1}^N$, we map them into a hidden space by using a hidden function $\varphi(\mathbf{x})$. Let $\mathbf{z} = \varphi(\mathbf{x})$, where $\mathbf{z}$ is the image of $\mathbf{x}$. We take kernel functions as hidden functions, and we have

$$\mathbf{z} = [k(\mathbf{x}, \mathbf{x}_1), ..., k(\mathbf{x}, \mathbf{x}_N)]^T \tag{4}$$

In hidden functions, we require only the symmetry for kernel functions instead of Mercer's condition. In addition, we can obtain the mapped samples $\mathbf{z}$. So, it is very convenient to calculate statistics for samples.

### B. HDNE

In a classification task, assume that the set of labeled training samples is $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i \in R^m$, and $y_i \in \{1, 2, ..., c\}$. By employing the hidden function (4), the images of $\mathbf{x}_i$ in the hidden space are

$$\mathbf{z}_i = [k(\mathbf{x}_i, \mathbf{x}_1), ..., k(\mathbf{x}_i, \mathbf{x}_N)]^T$$

In the hidden space, the training set can be represented as $\{\mathbf{z}_i, y_i\}_{i=1}^N$ where $\mathbf{z}_i \in R^N$.

Since the concrete form of samples has been known in the hidden space, so we are able to directly build the adjacent graph $\mathbf{F}$ in this space. The entries of $i$th row and $j$th column in $\mathbf{F}$ is

$$F_{ij} = \begin{cases} +1, & \mathbf{z}_i \textit{ and } \mathbf{z}_j \textit{ are neighbors and } y_i = y_j \\ -1, & \mathbf{z}_i \textit{ and } \mathbf{z}_j \textit{ are neighbors and } y_i \neq y_j \\ 0, & \textit{otherwise} \end{cases} \tag{5}$$

Recall that the dimensionality of hidden space may be very high, so it needs to reduce dimensionality for efficient computation. Let the transformation matrix be $\mathbf{P} \in R^{N \times d}$, where $d$ denotes the dimensionality of discriminant subspace. In the discriminant subspace, the sample $\mathbf{z}_i \in R^m$ is transformed to be $\mathbf{P}^T\mathbf{z}_i \in R^d$.

Let $\phi(\mathbf{P})$ and $\varphi(\mathbf{P})$ be the within-class and the between-class neighborhood scatters, respectively. The within-class neighborhood scatter $\phi(\mathbf{P})$ is defined as

$$\phi(\mathbf{P}) = \sum_{i,j,y_i=y_j} ||\mathbf{P}^T\mathbf{z}_i - \mathbf{P}^T\mathbf{z}_j||^2 \tag{6}$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ are neighbors and belong to the same class. The between-class neighborhood scatter $\varphi(\mathbf{P})$ is defined as

$$\varphi(\mathbf{P}) = \sum_{i,j,y_i \neq y_j} ||\mathbf{P}^T\mathbf{z}_i - \mathbf{P}^T\mathbf{z}_j||^2 \tag{7}$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ are still neighbors but belong to the different classes. We hope that if neighbors belong to the same class they should be close to each other in the discriminant subspace; otherwise, be far away from each other. We can implement our demand by minimizing the $\phi(\mathbf{P})$ and maximizing $\varphi(\mathbf{P})$ at the same time, which could be described as

$$\min_{\mathbf{P}} \quad \Delta(\mathbf{P}) = \phi(\mathbf{P}) - \varphi(\mathbf{P}) \tag{8}$$

Substituting (5) into the expression of $\Delta(\mathbf{P})$, we can rewritten it as follows:

$$\begin{aligned} \Delta(\mathbf{P}) &= \sum_{i,j=1}^{N} ||\mathbf{P}^T\mathbf{z}_i - \mathbf{P}^T\mathbf{z}_j||^2 F_{ij} \\ &= 2\sum_{i,j=1}^{N} (\mathbf{z}_i{}^T\mathbf{P}\mathbf{P}^T\mathbf{z}_i - \mathbf{z}_i{}^T\mathbf{P}\mathbf{P}^T\mathbf{x}_j)F_{ij} \\ &= 2\sum_{i,j=1}^{N} tr((\mathbf{P}^T\mathbf{z}_i\mathbf{z}_i{}^T\mathbf{P} - \mathbf{P}^T\mathbf{z}_j\mathbf{z}_i{}^T\mathbf{P})F_{ij}) \\ &= 2tr(\sum_{i,j=1}^{N} (\mathbf{P}^T\mathbf{z}_iF_{ij}\mathbf{z}_i{}^T\mathbf{P} - \mathbf{P}^T\mathbf{z}_jF_{ij}\mathbf{z}_i{}^T\mathbf{P})) \\ &= 2tr(\mathbf{P}^T\mathbf{Z}\mathbf{S}\mathbf{Z}\mathbf{P} - \mathbf{P}^T\mathbf{Z}\mathbf{F}\mathbf{Z}\mathbf{P}) \\ &= 2tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P}) \end{aligned} \tag{9}$$

where $tr(\cdot)$ denotes the trace of a matrix, and $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N]$, and $\mathbf{S}$ is a diagonal matrix with $S_{ii} = \sum_{j=1}^N F_{ij}$. As a result, the problem (8) can be rewritten as

$$\begin{aligned} \min_{\mathbf{P}} \quad & tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P}) \\ \text{s.t.} \quad & \mathbf{P}^T\mathbf{P} = \mathbf{I} \end{aligned} \tag{10}$$

and where $\mathbf{I}$ is the identity matrix.

In the following, we introduce a lemma in [25] and give a theorem which describes the solution to (10).

**Lemma 1**: Suppose $\mathbf{A} \in R^{N \times N}$ is a real symmetric matrix and its minimum eigenvalue is $\lambda_1$. The solution to the minimization problem of $\boldsymbol{\eta}^T\mathbf{A}\boldsymbol{\eta}$ which subjects to $\boldsymbol{\eta}^T\boldsymbol{\eta} = 1$ and $\boldsymbol{\eta} \in R^N$ is the eigenvector corresponding to the eigenvalue $\lambda_1$.

Theorem 2: Assume that the eigenvalues of the matrix $\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T$ are $\lambda_1 \leq ... \leq \lambda_{i-1} \leq \lambda_i \leq ... \leq \lambda_N$, and $\boldsymbol{\xi}_i$ is the corresponding eigenvector of eigenvalue $\lambda_i$. Then optimal $\mathbf{P}$ to the minimization problem $tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P})$ is the corresponding eigenvectors of the first $d$ eigenvalues. Namely, $\mathbf{P} = [\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_d]$.

**Proof**: Since $(\mathbf{S} - \mathbf{F})$ is a real symmetric matrix, $\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T$ is also a real symmetric matrix. According to Lemma 1, if $d = 1$ , only when $\mathbf{P}$ is the eigenvector corresponding to the minimum eigenvalue $\lambda_1$ of matrix $\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T$ (namely $\mathbf{P} = \boldsymbol{\xi}_1$) is $\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P}$ minimum. Right now, $\lambda_1$ is the optimal value of $tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P})$.

Similarly, if let $\mathbf{P}$ represent eigenvectors corresponding to the first $d$ minimum eigenvalues (namely $\mathbf{P} = [\xi_1, \cdots, \xi_d]$), then we have $tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S} - \mathbf{F})\mathbf{Z}^T\mathbf{P}) = \sum_{i=1}^{d} \lambda_i$. Right now,

$tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S}-\mathbf{F})\mathbf{Z}^T\mathbf{P})$ achieves its optimal value. This completes the proof.

Theorem 2 shows that minimizing $tr(\mathbf{P}^T\mathbf{Z}(\mathbf{S}-\mathbf{F})\mathbf{Z}^T\mathbf{P})$ is equivalent to eigendecompose on the matrix $\mathbf{Z}(\mathbf{S}-\mathbf{F})\mathbf{Z}^T$. If the projection matrix is composed of the eigenvectors corresponding to the first $d$ eigenvalues, the value of object function with respect to optimization problem (11) is minimum.

### C. Comparison of DNE, KDNE and HDNE

By constructing the adjacent graph for samples in the original space, DNE makes their local structure be preserved in a discriminant subspace. In addition, the problem of finding the projection matrix is equivalent eigendecompose $\mathbf{X}(\mathbf{S}-\mathbf{F})\mathbf{X}^T$.

Nevertheless, DNE is linear so that it cannot be applied to the linearly non-separable data in the original space. As a remedy for this drawback, KDNE makes DNE extend to be nonlinear, but it still utilizes the adjacent graph constructed in the original space. For KDNE, the matrix that needs eigendecomposition is $(\mathbf{S}-\mathbf{F})\mathbf{K}^T$ .

The method proposed here is first to map the data in the original space into into the hidden space and then to construct the adjacent graph in this space. The local structure of samples in the hidden space can be preserved when performing dimensionality reduction. Thus, HDNE remedies the drawbacks that DNE is not fit for nonlinear problems and KDNE cannot preserve the local structure of high-dimensional space. HDNE tries to eigendecompose $\mathbf{Z}(\mathbf{S}-\mathbf{F})\mathbf{Z}^T$ .

From the above, the three methods are all based on eigendecomposition of some matrix, and then obtain the projection matrix composed of the eigenvectors corresponding to the first $d$ minimum eigenvalues.

## IV. Simulation Experiments

In this section, to validate the efficiency of HDNE, we compare it with other methods, including PCA, LDA, LPP, NPE, DNE, KFDA and KDNE on image classification problems. Here, we consider two kinds of images: face and handwritten digit.

For KDNE, KFDA and HDNE, Gaussian kernel $k(\mathbf{x},\mathbf{x}') = \exp\{-p\|\mathbf{x}-\mathbf{x}'\|\}$ with the kernel parameter $p>0$ is used. This parameter $p$ is selected by using 5-fold cross validation.

### A. Face Recognition

Consider the widely studied ORL Face dataset (from the website: http://archive.ics.uci.edu/ml/datasets.html),which is created by the University of Cambridge and has a total of 400 face images with different illumination intensity and facial expression etc., each 112 x 92, from 40 persons and 10 for each one. It also gives considerations to race, gender and facial expression and is a frequently-used face dataset.

In the face recognition experiment, we mainly focus on the effect of the dimensionality of discriminant subspace on recognition rates under different choices for $K$, where $K$ is the parameter of the nearest neighbor (NN) classifier. Thus, without prior knowledge, $K$ is set to be 1, 3 and 5 respectively. In the experiment, we randomly select 5 samples from the same person for training and the rest are for test. There are 200 training and test samples, respectively. All samples are divided by 255 to implement normalization.

For the high-dimensional original images, we first utilize PCA to reduce dimensionality from 10,304 to 100. In doing so, there are two benefits. On the one hand, computations are greatly reduced. On the other hand, the majority of noises are diminished. We repeat our experiment 100 trials and report the average result on test sets.

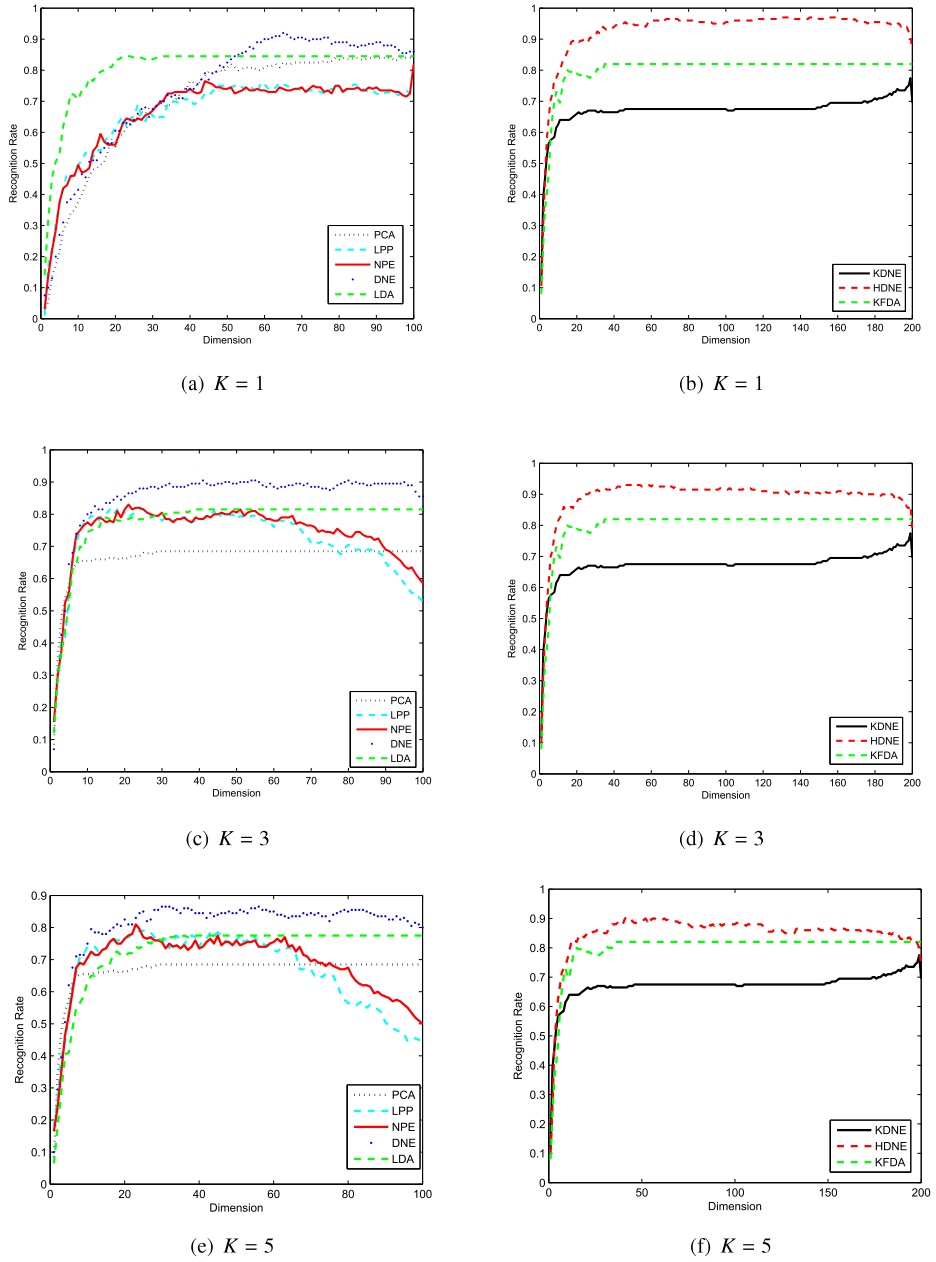
For PCA, LDA, LPP, NPE and DNE, their dimensionalities of discriminant subspace and recognition rates are plotted in one figure because their maximal dimensionalities are all 100 after dimensionality reduction. While for KDNE, KFDA and HDNE, their results are plotted in another one figure because they have the same maximal dimensionality of $N$.

When different $K$ value is selected, Fig. 1 presents the corresponding performance along with the change of dimensionality for these methods. From Figs. 1, we can know that, for all the methods, at the beginning the performance improves all the time along with increasing dimensionality, and then it tends to be invariable or decreasing. From Figs. 1(a), 1(c) and 1(e), we can know that with different K values, DNE method is always able to reach a maximum, being better than PCA, LDA, LPP and NPE, in different discriminant subspace. As the nonlinear version of DNE, from Fig. 1(b), 1(d) and 1(f), KDNE,KFDA and HDNE have great change in recognition rate along with the change of dimensionality. Obviously, HDNE works better than KDNE and KFDA no matter which $K$ is selected in our experiment.

Table I. Performance comparisons on ORL dataset ($K$ = 1)

| Method | Dimensional of subspace | Recognition rate |
|---|---|---|
| PCA | 79 | 84.00 |
| LDA | 22 | 84.50 |
| LPP | 59 | 76.00 |
| NPE | 100 | 82.00 |
| DNE | 65 | 92.00 |
| KDNE | 194 | 79.50 |
| KFDA | 35 | 82.00 |
| HDNE | 71 | 96.50 |

From Fig. 1, we can see that all methods have better performance when $K$ = 1. The larger $K$ does not mean better since ORL face data are insufficient. Table 1 provides the best recognition rates obtained by all methods and the corresponding dimensionality of discriminant subspace for $K$ = 1. Compared with KDNE, HDNE always has a better recognition rate and a lower dimensionality of discriminant subspace. As a result, our view that although being also a nonlinear extension of DNE, KDNE does not employ the adjacent graph to preserve the local structure so that lead to an unsatisfied recognition rate is verified. By contrast, HDNE method achieves this, which makes the following view be persuasive: compared with the practice that use kernel as nonlinear extension, this method that let the data map into high-dimensional space, and then construct adjacent graph to preserve the relationships between neighbors can work better and get a higher recognition rate in discriminant subspace.




(a) $K = 1$ (b) $K = 1$

(c) $K = 3$ (d) $K = 3$

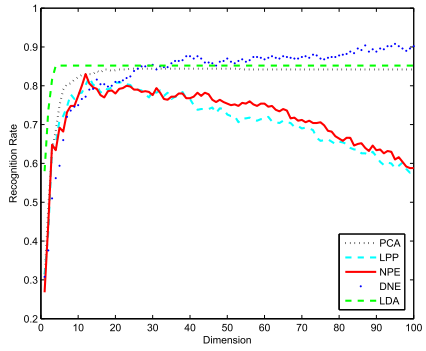
(e) $K = 5$ (f) $K = 5$

Figure 1. Recognition vs. dimensionality on ORL face dataset

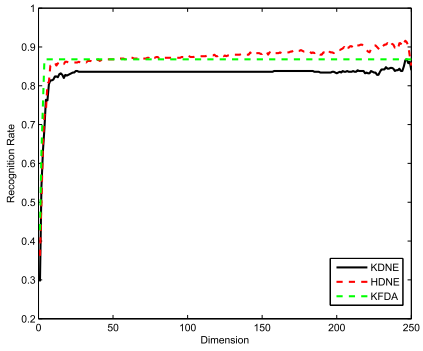
### B. Handwritten Digit Recognition

Consider the MNIST dataset *(from website http://archive.ics.uci.edu/ml/datasets.html)*, which has a total of 60,000 training and 10,000 test images with total 10 classes. Five classes, including digits 1, 3, 7, 8 and 9, are selected. For each class, we randomly select 50, 100 and 150 samples from the original training set as our training set, 100 samples from the original test set as our test set. In this experiment, PCA is stilled utilized to preprocess in order to obtain the *100-dimensional* data, and nearest neighbor is selected as the classifier.

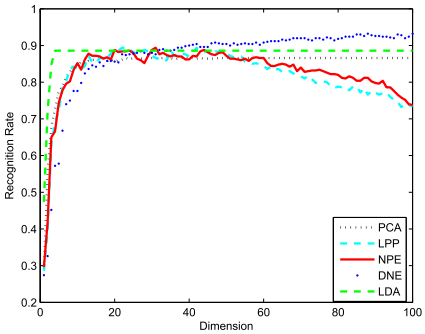
In this experiment, we will mainly perform analysis on the effect of the number of samples on the dimensionality of discriminant subspace and recognition rates. Fig. 2 clearly shows that the larger the number of samples is, the higher recognition rate is under condition of the same dimensionality for discriminant subspace. For Figs. 2(a), 2(c) and 2(e), the statuses of recognition rates for four methods are clearly presented along with the change of dimensionality of discriminant subspace. With the increase of dimensionality, the recognition rate of each method improves on the whole. Figs. 2(b), 2(d) and 2(f) respectively describe the recognition rates of KDNE and HDNE with respect to the dimensionality of discriminant subspace under condition of the same number of samples. We have the conclusion that with the same number of samples and the same dimensionality of discriminant subspace, as to recognition rate, HDNE obviously works better than KDNE and KFDA.



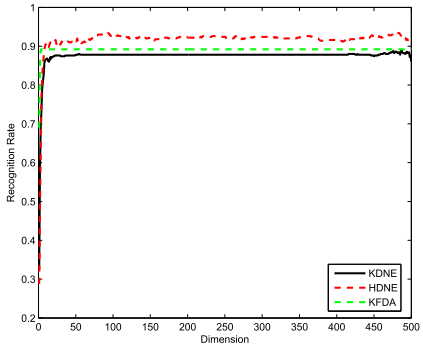

(a) 50 training samples



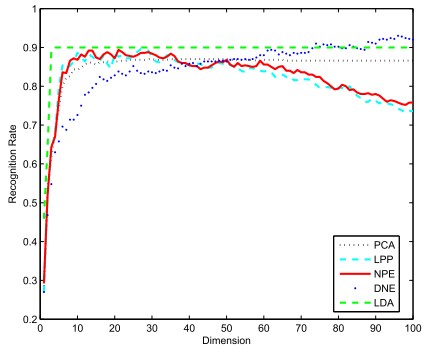

(b) 50 training samples



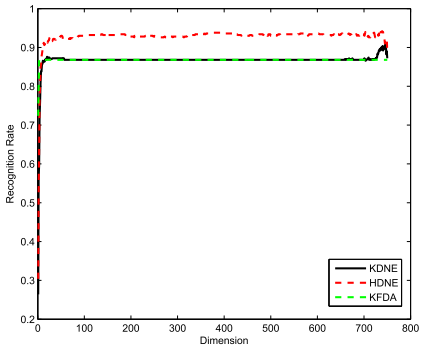

(c) 100 training samples




(d) 100 training samples




(e) 150 training samples




(f) 150 training samples

Figure 2. Recognition vs. dimensionality on MNIST dataset

Table II. PERFORMANCE COMPARISONS ON THE MNIST DATASET (50 *trainingsamples*)

| Method | Dimensional of subspace | Recognition rate |
|---|---|---|
| PCA | 24 | 84.40 |
| LPP | 14 | 82.20 |
| NPE | 12 | 83.00 |
| LDA | 4 | 86.40 |
| DNE | 95 | 90.10 |
| KDNE | 247 | 86.80 |
| KFDA | 3 | 86.86 |
| HDNE | 244 | 91.80 |

Table III. PERFORMANCE COMPARISONS ON THE MNIST DATASET (100 *trainingsamples*)

| Method | Dimensional of subspace | Recognition rate |
|---|---|---|
| PCA | 26 | 86.80 |
| LPP | 22 | 89.20 |
| NPE | 31 | 89.40 |
| LDA | 4 | 87.80 |
| DNE | 88 | 91.60 |
| KDNE | 487 | 89.00 |
| KFDA | 3 | 89.20 |
| HDNE | 480 | 93.80 |

Tables 2-4 show the optimal recognition rates for each method in the whole discriminant subspace with the certain number of samples, from which the conclusion is that recognition rate of HDNE is higher than the other methods. As a nonlinear extension of DNE as well, HDNE also has a higher recognition rate than KFDA and KDNE.

Table IV. PERFORMANCE COMPARISONS ON THE MNIST DATASET (150 *trainingsamples*)

| Method | Dimensional of subspace | Recognition rate |
|---|---|---|
| PCA | 29 | 87.00 |
| LPP | 27 | 89.40 |
| NPE | 21 | 89.40 |
| LDA | 4 | 90.00 |
| DNE | 96 | 91.80 |
| KDNE | 746 | 90.60 |
| KFDA | 3 | 86.98 |
| HDNE | 739 | 94.40 |

## V. CONCLUSIONS

HDNE is proposed by introducing hidden functions, which is a nonlinear extension of DNE. Specifically, it performs analysis on the preserved local structure in hidden space. The data being linearly non-separable in original space are linearly separable in hidden space and at the same time the adjacent graph is constructed to preserve the local structure of data. From experimental results on ORL dataset with different K values in K-nearest neighbor and on MNIST dataset with the different number of samples, we have the conclusion that HDNE has a better performance than the other methods.

## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61373093 and 61033013, by the Natural Science Foundation of Jiangsu Province of China under Grant Nos. BK2011284 and BK201222725, by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No.13KJA520001,and by the Qing Lan Project.

## REFERENCES

[1] J. B. Tenenbaum, V. de Silva and J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290, 2319-2323, 2000.

[2] S. Roweis and L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, 290, pp. 2323-2326, 2000.

[3] S. Roweis and L. K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, *Science*, vol. 290, 22 December, 2000.

[4] X. F. He and P. Niyogi, Locality Preserving Projections, *Advances in Neural Information Processing Systems 16* Vancouver, British Columbia, Canada, 2003.

[5] X. F. He, D. Cai, S. C. Yan, eta, Neighborhood Preserving Embedding, *Proceeding of the IEEE International Conference on Computer Vision.* Beijing, China, pp, 1208-1213, 2005

[6] H. T. Chen, H. W. Chang and T. L. Liu, Local discriminant embedding and its variants, *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 2005.

[7] M. Belkin and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Computation*, vol. 15, pp. 1373-1396, 2003.

[8] X. He, S. Yan, Y. Hu and H. J. Zhang, Learning a Locality Preserving Subspace for Visual Recognition, *Proc. 9th International Conference on Computer Vision*, France, 2003.

[9] K. C. Lee, J. Ho, M. H. Yang and D. Kriegman. Video-Based Face Recognition Using Probabilistic Appearance Manifolds, *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, Wisconsin, 2003.

[10] B. Scholkopf, A. Smola and K. R. Muller, Nonlinear Component Analysis as a Kernel Eigenvalue Problem, *Neural Computation*, *10* (5)*:* 1299-1319, 1998.

[11] M. Sugiyama, Local Fisher Discriminant Analysis for Supervised Dimensionality Reduction, *Proc of the 23rd International Conference on Machine Learning*,Pittsburgh, USA, pp, 905-912. 2006.

[12] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, PAMI, vol. 19, no. 7, pp. 711-720, July. 1997.

[13] M. Belkin and P. Niyogi, Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, NIPS 14, 2001.

[14] C. Hu, Chang Y, R. Feris and M. Turk, Manifold Based Analysis of Facial Expression, *IEEE Workshop on Face Processing in Video*,2004.

[15] M. H. Yang, Kernel Eigenfaces vs. Kernel Fisherfaces : Face Recognition Using Kernel Methods, AFGR, pp. 205-211, 2002.

[16] W. Zhang, X. Y. Xue, H. Lu and Y. F. Guo, Discriminant Neighborhood Embedding for Classification, *Pattern Recognition*, *39* (11)*:* 2240-2243, Nov. 2006.

[17] S. Saitoh, Theory of Reproducing Kernels and Its Applications, *UK* : *Longman Scienti fic and Technical*, 2004.

[18] J. Shawe−Taylor and N. Cristianini. Kernel Methods for Pattern Analysis, *Cambridge University Press*, 2004.

[19] L. Zhang, W. D. Zhou and L. C. Jiao, Hidden Space Support Vector Machines IEEE Trans, vol. 15, no. 6, 2004.

[20] K. Fukunaga, *Introduction to Statistical Pattern Recognition,* 2nd ed. New York : Academic, 1991.

[21] K. Mtiller, S. Mika, G. Riitsch, K. Tsuda and B. Sch61kopf, An Introduction to kernel-based learning algorithms, *IEEE Transactions on Neural Networks*, Vol. 12, pp. 181−201, 2001.

[22] D. M. J. Tax and R. P. W. Duin, Support vector data description, Mach. Learn. 54 (2004) 45-66

[23] W. D. Zhou, L. Zhang and L. C. Zhang. Hidden space principal component analysis 10*th* Pacific, *Asia Conference on Knowledge Discovery and Data Mining*, LNAI 3981, 801 − 805, 2006.

[24] L. Zhang, W. D. Zhou and P. C. Chang. Generalized Nonlinear Discriminant Analysis and Its Small Sample Size Problems Neurocomputing, 74, 568 − 574, 2011.

[25] W. Zhang and X. Y. Xue. *Study on Feature Transformation Algorithm based on K − Nearest − Neighbor Classification Rule,* Fudan University, China, 37-40, 2007.

2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# Locally Linear Embedding Algorithm Based on OMP for Incremental Learning

Yiqin Leng
School of Computer Science and Technology & Provincial Key Laboratory for Computer Information Processing Technology
Soochow University
Email: 20124227054@suda.edu.cn

Li Zhang
School of Computer Science and Technology & Provincial Key Laboratory for Computer Information Processing Technology
Soochow University
Email: zhangliml@suda.edu.cn

Jiwen Yang
School of Computer Science and Technology & Provincial Key Laboratory for Computer Information Processing Technology
Soochow University
Email: jwyang@suda.edu.cn@suda.edu.cn

***Abstract*—Locally Linear Embedding (LLE) is a sort of powerful nonlinear dimensionality reduction algorithms.The basic idea behind the LLE method is that each data point and its neighbors lie on or close to a locally linear patch of the manifold if there is sufficient data. Then the local geometry of these patches is described by using linear coefficients which can reconstruct each data point from its neighbors. However, LLE operates in a batch way and its dimension reduction cannot be generalized to unseen samples. If a test sample arrives, LLE must run repeatedly and the former computational results are discarded. Thus, some incremental methods have been proposed for LLE to solve this problem. In these incremental methods, the neighbor number is globally fixed, which may result in selecting points from another linear space as neighbors. This paper presents LLE based on orthogonal matching pursuit (OMP) and applies it to classification tasks. In the classification tasks, dimensionality reduction on test samples is implemented by applying dimension reduction on training samples.The new LLE method could select a more appropriate neighbors from the selected neighbors. OMP is applied to not only LLE for training samples, but also the incremental learning of LLE for test samples. Compared with other linear incremental methods, experimental results show that the proposed method is promising.**

## I. Introduction

The manifold learning algorithm has become the most extensive study for nonlinear dimensionality reduction since 2000 [1][2]. The most representative nonlinear manifold methods include locally linear embedding (LLE) [1], isometric feature mapping (ISOMAP) [2], Laplacian eigenmap [3], Hessian eigenmaps [4] and local tangent space alignment(LTSA) [5]. We focus on LLE here. LLE operates in a batch or offline mode, which means that LLE just works on the training set. Thus, the whole algorithm must run repeatedly and all the former computational results are discarded when performing dimensionality reduction on a test sample. It is worth considering to make LLE with incremental processing capabilities, and get the projection vector for all the test samples. The incremental versions for ISOMAP have been proposed [6][7], for LLE have been proposed in [8]–[10].

In [8], Kouropteva et al. proposed an incremental LLE based on a linear generalization, which assumes that a manifold is locally linear, called ILLE-LG here. ILLE-LG looks for a linear transformation matrix $K$ among nearest neighbors in both high and low dimensional spaces. Kouropteva et al. also developed an incremental version for LLE in [9] by solving an optimization problem. Although the optimization problem is feasible, it is not convenient to deal with, since optimization problem is transformed into solving the problem of feature vectors in dealing with LLE method. Saul et al. presented an alternative incremental method [10]. First, this method is to search $K$ nearest neighbors of a new sample $\mathbf{x}$. Then this new sample $\mathbf{x}$ is represented by using the linear combination of $K$ nearest neighbors. Finally, the linear weight coefficients got in previous step are applied to the corresponding embedded $K$ points in the low-dimensional space, which results in the embedded point for $\mathbf{x}$. To find linear weight coefficients, the least square (LS) method is adopted in this incremental scheme. Hereafter, we call this method ILLE-LS.

In LLE and its incremental methods mentioned above, the number of neighbors $K$ is a free parameter. It directly influences the mapping result. However, the neighbor number $K$ is globally fixed in LLE. In addition, Euclidean distance is used to select the neighbor points, and a larger neighbor number would result in selecting points from another linear space as neighbors(another linear space means another person face when dealing with face dataset ). LS is adopted to get the linear weight coefficients, and during processing it involves solving inverse matrix. If a matrix is singular, there is no solution or infinitely many . Even if we can employ the matrix perturbation to avoid the singularity of the matrix, we are not sure it is the best way to get the weighted coefficients.

Zhang et al. proposed an incremental LLE based on orthogonal matching pursuit (OMP) (ILLE-OMP) in [11]. ILLE-OMP is similar to ILLE-LS. The difference is that ILLE-OMP uses OMP instead of LS in the incremental scheme. However, ILLE-OMP uses the computational results obtained from LLE. If LLE generates a bad solution, then ILLE-OMP would work badly. To avoid the problem, this paper applies OMP to both LLE and incremental LLE. OMP could select a few proper neighbors from the given neighbors, and OMP method is not affected by a singular matrix. We call this method ILLE-OMP-OMP, which is applied to classification tasks.

This remainder of the paper is organized as follows: Section 2 gives a brief description of LLE and two linear incremental methods for it. Section 3 presents LLE based on OMP for classification, ILLE-OMP-OMP. Section 4 simulates

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

and evaluates the performance of our method. Finally, some conclusions are made about our work in Section 5.

## II. Related work

### A. Locally linear embedding

In LLE, there are two optimization problems. One is to find linear represented coefficients of each point in original space, and the other is to find low dimensional coordinates when the linear represented coefficients of all points are generated. We give a brief description of LLE in the following.

Consider a set of unlabeled samples $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, where $\mathbf{x}_i \in R^D$, $D$ is the dimensionality of samples in a high-dimensional space, and $n$ is the number of samples. The goal of LLE is to transform data in the high-dimensional space into a low-dimensional embedding space, or $\mathbf{x}_i \rightarrow \mathbf{y}_i,\ i = 1, \cdots, n$, where $\mathbf{y}_i \in R^d$, $d$ is the dimensionality of the embedding space and $D \gg d$.

The procedure of finding the corresponding embedded points $\mathbf{y}_i$ of $\mathbf{x}_i$ can be described as follows. First, we find $K$ nearest neighbors for each $\mathbf{x}_i$ and put them into a set $X_i^K = \{\mathbf{x}_{i1}, ..., \mathbf{x}_{iK}\}$, where $\mathbf{x}_{im}$ is the $m$-neighbors of $\mathbf{x}_i$. Second, we represent $\mathbf{x}_i$ by the linear combination of its neighbors, or

$$\mathbf{x}_i = \sum_{j=1}^{n} w_{ij}\mathbf{x}_j \tag{1}$$

where $w_{ij}$ is the weight coefficient of $\mathbf{x}_j$ when representing $\mathbf{x}_i$, and

$$\begin{cases} w_{ij} \neq 0, & if \quad \mathbf{x}_j \in X_i^K \\ w_{ij} = 0, & if \quad \mathbf{x}_j \notin X_i^K \end{cases} \tag{2}$$

which means that it requires to find only $K$ weight coefficients instead of $n$ weight coefficients where $n > K$. The LS algorithm is used to solve the following $n$ optimization problems:

$$\varepsilon(\mathbf{W}) = \sum_i \left| \mathbf{x}_i - \sum\nolimits_j w_{ij}\mathbf{x}_i \right|^2 \tag{3}$$

which subject to (2) and $\sum_{j=1}^{n} w_{ij} = 1$. Finally, we fix the weight coefficients and compute the low-dimensional embedded points by optimizing the following problem:

$$\begin{aligned} \min_{\mathbf{Y}} \quad & \sum_{i=1}^{n} \left\| \mathbf{y}_i - \sum_{j=1}^{n} w_{ij}\mathbf{y}_j \right\|^2 \\ s.t. \quad & \tfrac{1}{n}\sum_{i=1}^{n} \mathbf{y}_i\mathbf{y}_i^T = \mathbf{I},\ \sum_{i=1}^{n} \mathbf{y}_i = 0 \end{aligned} \tag{4}$$

where $\mathbf{I}$ is the identify matrix and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n] \in R^{d \times n}$. Define a new matrix $\mathbf{M}$ by

$$\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W}) \tag{5}$$

where $\mathbf{M}$ is a sparse and symmetric matrix, $\mathbf{W}$ is the weight matrix, whose entry in the $i$-th row and the $j$-th column is $w_{ij}$. The solution to (4) is cast into finding the eigenvectors corresponding to $(d+1)$ smallest eigenvalues of $\mathbf{M}$. Note that the first eigenvector which consists of 1's is excluded.

LLE has merits with easy implementation and small computational complexity. While the performance of LLE mainly depends on the selection of parameter of neighbors. The selection of neighbors is a problem. If $K$ is too small, LLE would lose the global significance. If $K$ is too large, LLE would lose the non-linear characteristics. In addition, the LS algorithm is used to find the weight coefficients, which may make the point of another linear space with a greater weight. As a result, it would lose the significance of linear representation.

### B. Linear incremental version for locally linear embedding

It is well known that LLE operates in a batch or offline mode. In order to obtain the embedded coordinates of test set, incremental version for LLE is considered for classification. In linear incremental version for LLE, assume that any nonlinear manifold can be considered as locally linear, which consists with the LLE's assumption. There are two possibilities of linear incremental methods [8]–[10]. The linear incremental method is to derive and use a transformation between the original and projected data. In the following, we give a brief description of the two linear incremental methods, or ILLE-LG and ILLE-LS.

*1) ILLE-LG:* Let the original samples be $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ and the corresponding embedded points be $\{\mathbf{y}_1, \cdots, \mathbf{y}_n\}$. Given a test sample $\mathbf{x}_{n+1}$, we select $K$ nearest neighbors of $\mathbf{x}_{n+1}$ from the original samples. Put the $K$ nearest neighbors of $\mathbf{x}_{n+1}$ and the corresponding embedded points into matrixes: $\mathbf{X}_{n+1}^K = [\mathbf{x}_{(n+1)1}, ..., \mathbf{x}_{(n+1)K}]$ and $\mathbf{Y}_{n+1}^K = [\mathbf{y}_{(n+1)1}, ..., \mathbf{y}_{(n+1)K}]$. By using the assumption that the manifold is locally linear, the following equation is approximately true:

$$\mathbf{Y}_{n+1}^K = \mathbf{Z}\mathbf{X}_{n+1}^K \tag{6}$$

where $\mathbf{Z}$ is a linear transformation matrix of size $d \times D$ and can be determined by

$$\mathbf{Z} = \mathbf{Y}_{n+1}^K (\mathbf{X}_{n+1}^K)^T \mathbf{C}^{-1} \tag{7}$$

where $\mathbf{C} = \mathbf{X}_{n+1}^K \left(\mathbf{X}_{n+1}^K\right)^T \in R^{D \times D}$. Since $\mathbf{X}_{n+1}^K$ is the neighborhood of $\mathbf{x}_{n+1}$ and LLE preserves local structures, the new projection can be found as

$$\mathbf{y}_{n+1} = \mathbf{Z}\mathbf{x}_{n+1} \tag{8}$$

*2) ILLE-LS:* To find the embedded point $\mathbf{y}_{n+1}$ of the test sample $\mathbf{x}_{n+1}$. First, the $K$ nearest neighbors of $\mathbf{x}_{n+1}$ are detected among the original points in the $D$-dimensional space. Then the linear weights $\mathbf{w}_{n+1}$, which will reconstruct $\mathbf{y}_{n+1}$ from its neighbors are computed by using LS. Finally, the new embedded point $\mathbf{y}_{n+1}$ is found as: $\mathbf{y}_{n+1} = \sum_{j=1}^{n} w_{(n+1)j}\mathbf{y}_j$.

## III. Locally Linear Embedding Based on OMP for Incremental learning

This paper presents a new approach based on OMP [12]–[14] which can select a few good neighbors from the selected neighbors. OMP is applied to both LLE and its incremental algorithm for classification tasks. Compared with LS, OMP would result in a stable solution. In the following, OMP is first introduced and then LLE based on OMP is proposed for classification tasks.

### A. Orthogonal Matching Pursuit (OMP)

In recent years, compressed sensing has attracted substantial attentions in signal processing, machine learning, computer vision, etc. There has the similar problem of linear representation in compressed sensing, or sparse signal reconstruction. So far, lots of methods for sparse signal reconstruction have been proposed. Generally, three main techniques or their combinations are available to obtain a sparse representation, including zero-trapped loss functions, sparse regularization and matching pursuit [15]. Here, we focus on matching pursuit methods, which are greedy and to solve the 0-norm problem. In fact, the 0-norm regularization is the desirable one to obtain sparseness, but the 0-norm regularization is so discontinuous that it is very difficult to optimize the objective function containing it. Therefore, the greedy matching pursuit methods are the best ones for finding the 0-norm problem.

OMP is one of signal reconstruction methods, which is famous for its faster optimization speed and a sparse solution. The signal reconstruction problem can be formulated as:

$$\begin{aligned} &\min \quad \|\mathbf{v}\|_0 \\ &s.t. \quad \mathbf{u} = \mathbf{A}\mathbf{v} \end{aligned} \tag{9}$$

where $\mathbf{v}$ is the sparse vector, $\mathbf{u}$ is the measurement vector, $\mathbf{A}$ is the sparse representation matrix and $\|\cdot\|_0$ denotes the 0-norm of $\cdot$. OMP is a greedy method that identifies the location of one nonzero entry of $\mathbf{v}$ at a time when solving the optimization problem (6). The OMP algorithm is given in Algorithm 1.

TABLE I. ALGORITHM FOR ORTHOGONAL MATCHING PURSUIT

**Algorithm1** OMP method

**Input:**
-$\mathbf{u}$: the measurement vector
-$\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_n] \in R^{m\times n}$: the sparse representation matrix
-$S$: the sparse degree
-$\delta > 0$: the threshold
**Output:**
- $\mathbf{v}$: Sparser vector

Step 1 Initialize the iteration time $t = 0$ , the solution $\mathbf{v}^t = [0, \cdots, 0]^T$ ,the residual $\mathbf{r}^t = \mathbf{u} - \mathbf{A}\mathbf{v}^t = \mathbf{u}$, and the index set $I^t = \emptyset$.
Step 2 While $t \le S$ or $\|\mathbf{r}^t\|_2 < \delta$ do
$t = t + 1$;
Compute $\lambda_t = \arg\max_{j\in\{1,\cdots,n\}\setminus I^t} \mathbf{a}_j^T \mathbf{r}^{t-1}$ and let $I^t = I^t \cup \{\lambda_t\}$ ;
Use LS to solve the optimization problem: $\mathbf{v}^t = \min \|\mathbf{u} - \sum_{j\in I^t} \mathbf{a}_j v_j\|_2$
Update the residual $\mathbf{r}^t = \mathbf{u} - \mathbf{A}\mathbf{v}^t$;
End while

### B. LLE based on OMP for incremental learning

This section deals with LLE based on OMP for incremental tasks. Our scheme divided into three parts. LLE-OMP is first used for the given training samples to generate corresponding embedded points, and then ILLE-OMP is applied to test samples to obtain their embedded points. Finally, a K nearest neighbor classifier is adapted to classify (KNN) test samples in the embedding space.

*1) Dimensionality reduction for training set:* Generally, Euclidean distance is used to select the neighbor points, which may results in selecting points from another linear space as the neighbor points. Here, LS in the second step of LLE is replaced by OMP when dealing with a given training set. OMP method can select a few proper points from a large number of points and would result in a sparse solution. LS is unable to obtain a sparse solution.

Similar to LLE, LLE-OMP also has three steps. The second step of LLE-OMP is different from LLE. First, we look for the $K$ nearest neighbors for each $\mathbf{x}_i$ and put them into a matrix $\mathbf{X}_i^K = [\mathbf{x}_{i1}, ..., \mathbf{x}_{iK}]$. Second, we represent $\mathbf{x}_i$ by the linear combination of its neighbors. OMP is used to obtain the weight coefficients. The optimization problem is described as follows:

$$\begin{aligned} &\min \quad \left\|\mathbf{w}_i^K\right\|_0 \\ &s.t. \quad \mathbf{x}_i = \mathbf{X}_i^K \mathbf{w}_i^K \end{aligned} \tag{10}$$

where $\mathbf{w}_i^K = [w_{i1}, ..., w_{iK}]^T$. The solution to (10) is sparse since (10) is the 0-norm problem. The third step in LLE-OMP is the same as that in LLE. The LLE-OMP algorithm is given in Algorithm 2.

TABLE II. ALGORITHM FOR LLE BASED ON OMP

**Algorithm2** LLE-OMP method

**Input:**
-$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in R^{D\times n}$: the high-dimensional sample matrix
-$K$: the number of nearest neighbors
-$d$: the dimensionality of embedding space
-$\delta > 0$: the threshold
-$S$: the sparse degree
**Output:**
-$\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_n] \in R^{d\times n}$: The test embedded sample matrix

Step 1 Search the $K$ nearest neighbors for each $\mathbf{x}_i$ $(i = 1, ..., n)$ by using the Euclidean distance as a similarity measure. Let $\mathbf{X}_i^K \in R^{D\times K}$ be the matrix consisting of nearest neighbors of $\mathbf{x}_i$.
Step 2 Use OMP to solve (10) to obtain reconstruction weights $\mathbf{w}_i$ for each $\mathbf{x}_i$.
Step 3 Let $\mathbf{M} = (\mathbf{I} - \mathbf{W})^T (\mathbf{I} - \mathbf{W})$ where $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_n]$. Eigendecompose $\mathbf{M}$ and obtain the first $(d + 1)$ smallest eigenvalues and the corresponding eigenvectors $\mathbf{u}_i, i = 1, \cdots, (d + 1)$.
Step 4 Let$\mathbf{Y} = [\mathbf{u}_2, ..., \mathbf{u}_{d+1}]^T$.

*2) Dimensionality reduction for test set:* We have given a brief description of the two linear incremental methods of LLE to get the embedding coordinates for a test set in Section 2. Now, we come up with another linear incremental method of LLE. This incremental method is based on OMP, which is developed in [11].

As ILLE-LG and ILLE-LS, we take advantage of the existing embedded representations of the original samples, i.e. training samples. First, we receive corresponding embedded representations $\mathbf{Y}_{train} = [\mathbf{y}_1, ..., \mathbf{y}_n]$ of $\mathbf{X}_{test} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ by LLE-OMP. Then, we try to find a test embedded point $\mathbf{y}_{n+1}$ corresponding to the test point $\mathbf{x}_{n+1}$ which is sampled from the same manifold as $\mathbf{X}_{train}$. Like tanning samples, test samples can be linear reconstructed by its neighbors. The procedure of finding the test embedded point $\mathbf{y}_{n+1}$ is similar to ILLE-LS method, find the $K$ nearest neighbors of $\mathbf{x}_{n+1}$ among $\mathbf{X}_{train}$ and $\mathbf{x}_{n+1}$ is linear reconstructed by the neighbors. The linear weights $\mathbf{w}_{n+1}^K$ is solved by OMP method and finally get the

embedded point $\mathbf{y}_{n+1} = \mathbf{Y}_{n+1}^{K}\mathbf{w}_{n+1}^{K}$ . The ILLE-OMP-OMP algorithm is given in Algorithm 3.

TABLE III. LLE BASED ON OMP FOR INCREMENTAL LEARNING

**Algorithm3** ILLE-OMP-OMP method

**Input:**
-$\mathbf{X}_{train} = [\mathbf{x}_1, ..., \mathbf{x}_n] \in R^{D\times n}$: the high-dimensional sample matrix
-$K$: the number of nearest neighbors
-$d$: the dimensionality of embedding space
-$\mathbf{X}_{test} = \{\mathbf{x}_i\}_{i=n+1}^{N} \in R^{D}$:the new high-dimensional sample set
-$S$: the sparse degree
-$\delta > 0$: the threshold
**Output:**
-$\mathbf{Y}_{test} = \{\mathbf{y}_i\}_{i=n+1}^{N} \in R^{d}$: The test embedded sample matrix

Step 1 Get embedded points $\mathbf{Y}_{train} = [\mathbf{y}_1, ..., \mathbf{y}_n]$ of $\mathbf{X}_{trian}$ by LLE-OMP.
Step 2 Incremental learning:
1) Search nearest neighbors of each $\mathbf{x}_i \in \mathbf{X}_{test}, (n+1 \le i \le N)$ among $\mathbf{X}_{train}$ and put them into a matrix $\mathbf{X}_i^K$, and find the embedded points $\mathbf{Y}_i^K$ of $\mathbf{X}_i^K$.
2) Linearly reconstruct $\mathbf{x}_i$ from $\mathbf{X}_i^K$. The weight coefficient $\mathbf{w}_i^K$ is solved by OMP method.
3) The new embedded point $\mathbf{y}_i$ of $\mathbf{x}_i$ is $\mathbf{y}_i = \mathbf{Y}_i^K\mathbf{w}_i^K$.
4) $\mathbf{Y}_{test} = [\mathbf{y}_{n+1}, ...., \mathbf{y}_N]$.

*3) LLE based on OMP for Classification :* After the dimensionality reduction, we usually perform the classification on the test set. The most common used classifier, KNN is used here.

## IV. SIMULATION

In this section, we evaluate the performance on classification tasks of our method (ILLE-OMP-OMP) with respect to other three linear incremental methods ILLE-LS, ILLE-LG and ILLE-OMP. Both ILLE-OMP-OMP and ILLE-OMP have three parameters, the number of nearest neighbors $K$, the dimensionality of embedding space $d$ and the sparse degree $S$. ILLE-LS and ILLE-LG have two parameters, $K$ and $d$. The performance index is recognition rate on the test embedded sets. Extensive experiments have been carried out on two UCI datasets [16] as well as some real world datasets [17]–[19].

### A. UCI database

Two common data sets are from UCI database [16], or the Wine and Vehicle datasets. The Wine data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The Wine data set consists of 178 data from three types, whose dimension is 13. The purpose of using Vehicle data set is to classify a given silhouette as one of four types of vehicle by using a set of features extracted from the silhouette. The vehicle may be viewed from one of many different angles. The Vehicle data set consists of 846 data, each of which has 18 features.

For visualization, let $d = 2$. We repeat 10 trials. In each trial, each dataset is divided into two halves, half is testing samples and half is training samples. The partition of the data set is performed randomly. Generating the coordinates of the test samples in $R^d$, a KNN classifier is carried out in $R^d$ to classify them. The parameter $K$ in the KNN classifier is exactly identical to the number of nearest neighbors in LLE.

The curves of average recognition rates on the test embedded data against $K$ from 4 to 13 for two datasets are shown in Figures 1 and 2, respectively. In Figures 1 and 2, $S = 3$. From Figure 1, we can see that when $K > 6$ , ILLE-OMP-OMP is the best one among the four incremental methods in the view of classification recognition rate on the Wine dataset. Observation on Figure 2 also indicates this fact on the Vehicle dataset. The ILLE-OMP-OMP has the best performance compared to ILLE-LS, ILLE-LG and ILLE-OMP.

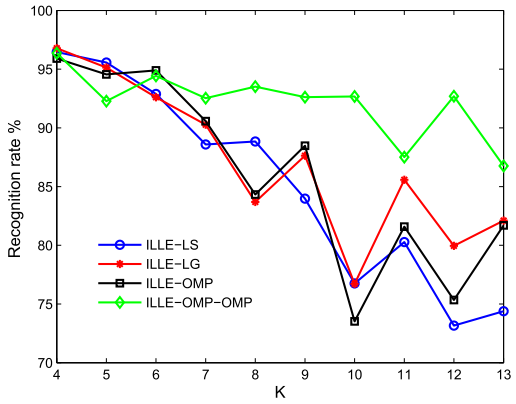



Fig. 1. Average recognition rates on the Wine dataset

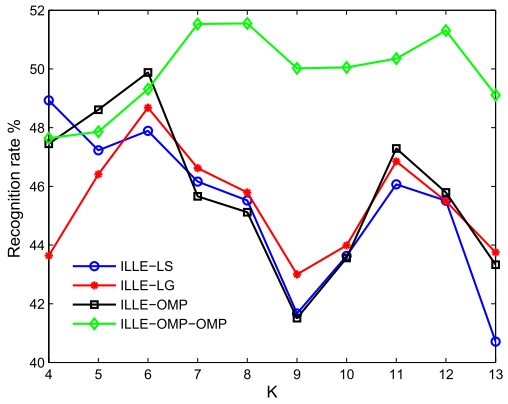



Fig. 2. Average recognition rates on the Vehicle dataset

Figures 3 and 4 visualize the embedded points of the two test data sets obtained with the proposed ILLE-OMP-OMP, ILLE-OMP, ILLE-LS and ILLE-LG under $K = 8$, $d = 2$ and $S = 3$. Table IV and Table V show the recognition rates in $R^D$ on Wine and Vehicle, respectively. As can be seen in Figure 3 and Figure 4, the test data obtained with the proposed ILLE-OMP-OMP are best presented in the two-dimensional space. The visualization on the Vehicle data is not ideal, which is associated with the low recognition rate. Moreover, from Tables IV and V, we can observe that the recognition rate decreases after dimensionality reduction .

### B. MNIST dataset

The MNIST dataset contains ten handwritten digits, from 0 to 9. Each digit is a sequence of 784-dimensional vectors.

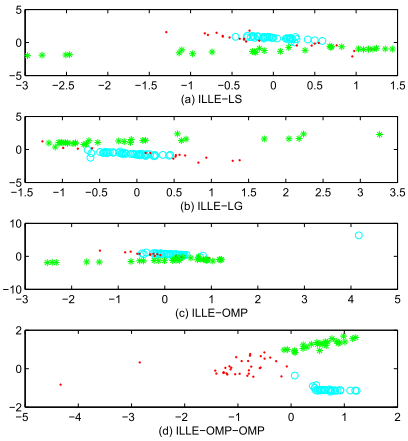
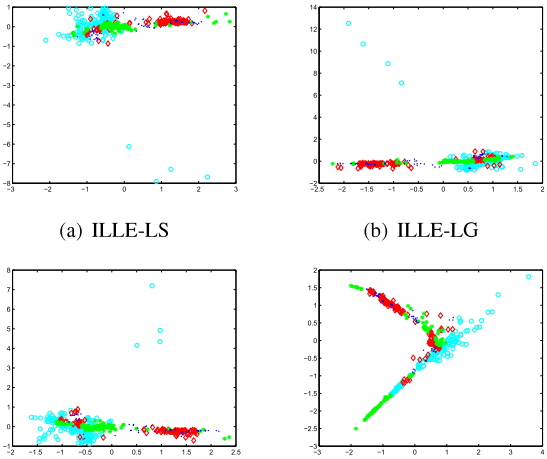



Fig. 3. The visualization of the Wine test data set obtained with ILLE-LS(a), ILLE-LG(b), ILLE-OMP(c) and ILLE-OMP-OMP (d)



(a) ILLE-LS (b) ILLE-LG

(c) ILLE-OMP (d) ILLE-OMP-OMP

Fig. 4. The visualization of the Vehicle test data set obtained with ILLE-LS(a), ILLE-LG(b), ILLE-OMP(c) and ILLE-OMP-OMP (d)

TABLE IV. ORIGINAL TEST SET RECOGNITION RATE OF KNN ON WINE (%)

| $K$ | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|
| Recognition rate | 96.20 | 96.46 | 96.46 | 96.84 | 96.48 |
| $K$ | 9 | 10 | 11 | 12 | 13 |
| Recognition rate | 96.86 | 96.62 | 96.91 | 96.91 | 97.05 |

TABLE V. ORIGINAL TEST SET RECOGNITION RATE OF KNN ON VEHICLE(%)

| $K$ | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|
| Recognition rate | 69.04 | 69.40 | 68.86 | 68.40 | 68.58 |
| $K$ | 9 | 10 | 11 | 12 | 13 |
| Recognition rate | 68.16 | 68.10 | 68.05 | 67.17 | 67.63 |

Compared with Wine and Vehicle datasets, MNIST dataset has more samples with high dimensionality. So, we not only present the recognition rate, but also study the running time of these methods. CPU running time includes the time for dimensionality reduction and for KNN classification.

Because of large memory requirements for these methods, we only select five classes, or classes 1, 3, 7, 8 and 9. These digits have similar shapes, such as 1 and 7 or 3, 8 and 9. The number of each class is 200 in the training set and 500 in the test set, respectively. Thus, the training set comprises 1000 samples whereas the test set consists of 2500 samples. We perform experiments on dimensionality reduction under the condition of different $K$. For visualization, let $d$=2. The average recognition rate curves on the test embedded data against $K$ is shown in Figure 5 in which $S = 3$. The K nearest neighbor classifier is also performed in $R^D$ with different $K$ as shown in Table VI. Results of classification are presented in Table VII ( $k = 13, d = 2, S = 3$ ), which shows the average recognition rate attained in classifying a certain class of digits. CPU running time of all methods are given in Table VIII.

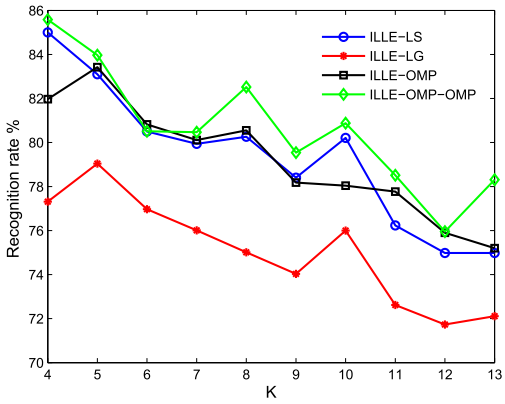



Fig. 5. Average recognition rates on the mnist dataset

TABLE VI. THE ORIGINAL TEST SET RECOGNITION RATE OF KNN ON MNIST (%)

| $K$ | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|
| Recognition rate | 92.79 | 93.11 | 92.64 | 92.90 | 92.32 |
| $K$ | 9 | 10 | 11 | 12 | 13 |
| Recognition rate | 92.62 | 91.85 | 92.40 | 91.73 | 92.13 |

TABLE VII. RECOGNITION RATES ON EACH DIGIT (%)

| $K$=13 | LLE-LS | LLE-LG | LLE-OMP | LLE-OMP-OMP |
|---|---|---|---|---|
| 1 | 96.72 | 97.56 | 97.08 | 99.08 |
| 3 | 81.76 | 81.64 | 82.88 | 79.84 |
| 7 | 51.60 | 58.64 | 54.76 | 60.48 |
| 8 | 64.52 | 48.96 | 64.52 | 75.24 |
| 9 | 74.52 | 62.16 | 73.52 | 71.52 |
| average | 73.82 | 69.79 | 74.55 | 77.23 |

TABLE VIII. CPU TIME ON MNIST (SEC.)

| KNN in $R^D$ | LLE-LS | LLE-LG | LLE-OMP | LLE-OMP-OMP |
|---|---|---|---|---|
| 49.51 | 6.05 | 5.78 | 7.36 | 6.79 |

Observation on Figure 5 indicates that ILLE-OMP-OMP has a higher recognition rate than ILLE-LS, ILLE-LG or ILLE-OMP on MNIST test dataset. After dimensionality reduction, the recognition rate decreases. But, as Figure 6 shows, dimensionality reduction makes data visualization. Figure 6 is the visualization results on MNIST test dataset obtained with the four linear incremental methods under $K = 8$, $d = 2$ and $S = 3$. The CPU time on Table VIII reflects the benefits of dimensionality reduction.

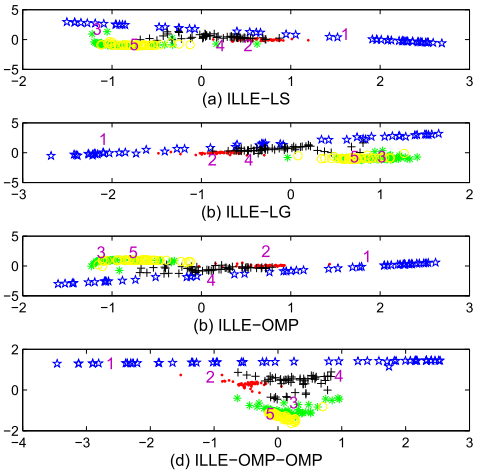



Fig. 6. The visualization of the MNIST test data set obtained with ILLE-LS(a), ILLE-LG(b), ILLE-OMP(c) and ILLE-OMP-OMP. The blue '⋆' is digit 1, red '.' is digit 3, green '∗' is digit 7, black '+' is digit 8 and yeloow 'o' is digit 9

TABLE IX. DETAILS OF BENCHMARK FACE DATA SETS

| Dataset | Samples | Dimensionality | # Classes |
|---|---|---|---|
| ORL | 400 | 2576 | 40 |
| YALE | 165 | 4096 | 15 |
| Extended Yale | 2414 | 1024 | 38 |

### C. Face data sets

In this study, three public face data sets are considered. The ORL face dataset consists of 10 images for each of the 40 human subjects, which were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). The images were taken with a tolerance for some tilting and rotation of the face up to 20 $^o$. The Yale face dataset contains 11 gray scale images for each of the 15 individuals. The images demonstrate variations in lighting condition (left-light, center-light, right-light), facial expression (normal, happy, sad, sleepy, surprised and wink), and with/without glasses. The extended Yale face database B contains 16128 images of 28 human subjects under nine poses and 64 illumination conditions. This dataset now has 38 individuals.

The ORL face data is reduced to 56*46 pixels by using random projection, which is used in [20]. The Yale face data used here is the cropped images which are resized to 64*64 pixels. For the extended Yale face database B, we simply use the cropped images and resize them to 32*32 pixels. The details of these three data sets are described in Table IX. Some instances of these three face samples are shown in Figure 7-9.

For each face dataset, we conduct 10 trials. In one trial, the percentage of training samples is set to be 50% and the remaining 50% data is used for testing. The partition is done randomly.

As the training samples of ORL and Yale for each type is only five, let $K = 4$ in ILLE-LS and ILLE-LG. For ILLE-OMP-OMP, $K = 8$ and $S = 4$. In ILLE-OMP, $K = 4$ for the training process, and $K = 8$ and $S = 4$ for the test

Fig. 7. Some samples on ORL dataset

Fig. 8. Some samples on Yale dataset

Fig. 9. Some samples on Extended Yale dataset

process. A face image is recognized by using KNN classifier ($K = 4$) applied in the low dimensional space. The face data have a rather high dimensionality. Thus, it is difficult to both visualize the face data and get good classification performance at the same time. We compare the classification performance of four methods when varying the dimensionality of embedding space $d$. The maximal embedded dimensionality is less than the number of training samples. Figures 10 and 11 illustrate the average recognition rate on the test data in ORL and Yale datasets. From Figure 10, ILLE-OMP-OMP is the best method and outperforms when varying the embedded dimensionality from 20 to 120. On the Yale dataset, ILLE-OMP-OMP is better than other methods when varying $d$ from 30 to 50. In other cases, ILLE-OMP-OMP is comparable to others. Note that the performance of ILLE-LG is very bad on both datasets.

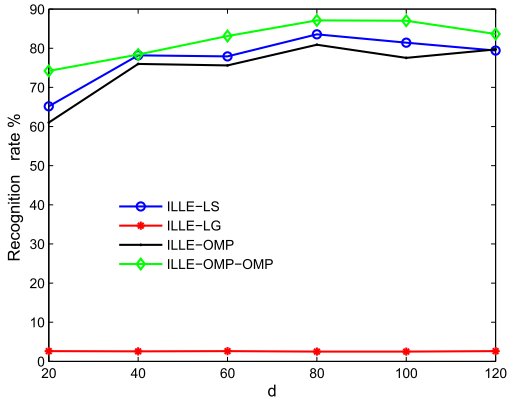



Fig. 10. The average recognition rate on the ORL test data set

The number of the Extended Yale dataset is relatively larger

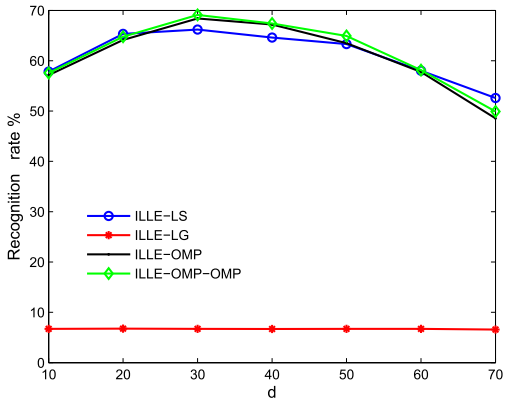



Fig. 11. The average recognition rate on the Yale test data sett

than that of Yale or ORL. So we let $K = 10$ in ILLE-LS and ILLE-LG, while in ILLE-OMP-OMP, $K = 20$, $S = 10$. In ILLE-OMP, $K = 4$ for the training set, and $K = 8$ and $S = 4$ for the test set. In KNN classifier, $K = 10$. Figure 12 gives the average recognition rate on the test data in the Extended Yale dataset. This figure also indicates ILLE-OMP-OMP outperforms other methods when $d$ varies from 100 to 800. On this dataset, ILLE-LG also has a bad performance.

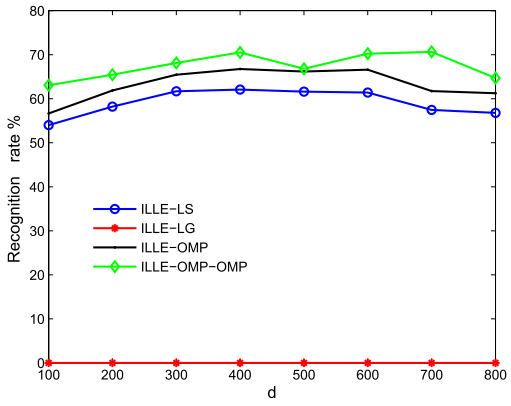



Fig. 12. The average recognition rate on the Extended Yale test data sett

The best performance obtained by four methods are shown in Table X, including KNN, ILLE-OMP-OMP, ILLE-LS, and ILLE-OMP. Since the recognition rate of ILLE-LG is not comparable, the corresponding result is not given here. On these face datasets, ILLE-OMP-OMP performs well.

TABLE X. THE MAXIMAL AVERAGE RECOGNITION OBTAINED WITH THREE FACE DATA SETS %

| | ORL | Yale | Extended Yale |
|---|---|---|---|
| KNN | 82.45 | 64.86 | 64.86 |
| ILLE-LS | 83.57 | 66.19 | 62.07 |
| ILLE-OMP | 80.90 | 68.41 | 66.76 |
| ILLE-OMP-OMP | 87.12 | 69.10 | 70.65 |

## V. CONCLUSION

We propose LLE based on OMP method for classification. Our algorithm benefits from two important properties: (a) In the second step of LLE, OMP is used for finding the linear weight coefficients, which can select a few good neighbors from the selected neighbors and (b) In the incremental learning of LLE, OMP is also used for finding the linear weight coefficients of a new sample. We apply our method to the face recognition problem as well as the handwriting digits recognition task. Classification experiments on these datasets show that our approach outperform other incremental methods. Although our method can be applied to classification tasks, it does not utilize the label information of training data. In the future, we try to introduce supervised information in out method and to get better classification performance.

## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61373093, 61033013, and 61271301, by the Natural Science Foundation of Jiangsu Province of China under Grant Nos. BK2011284 and BK201222725, by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No.13KJA520001 and by the Qing Lan Project.

## REFERENCES

[1] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323-2326, 2000.

[2] J. B. Tenenbaum, V. de. Silva and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319-2323, 2000.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *TNeural Computation*, vol. 15, no .6, pp. 1373-1396, 2003.

[4] D. L. Donoho and C. Grimes, "Hessian eigenmaps: new locally linear embedding techniques for high dimensional data," *PNAS*, vol. 100, no. 10, pp. 5591-5596, 2003.

[5] Z. Zhang and H. Zha, "Principal manifolds and nonlinear dimension reduction via local tangent space alignment," *SIAMS scientific Computing* , vol. 26, pp. 313-338, 2005.

[6] M. H. C. Law and A. K. Jain, "Incremental Nonlinear Dimensionality Reduction by Manifold Learning," *TPAMI* , vol. 28, pp. 377-391, 2006.

[7] Y. Zhang, Y. Wang, C. Li and K. Wang, "Kernel based Incremental Learning Isomap Algorithm," *Proc. IEEE International Conference. Information and Automation* , pp. 184-189, 2008.

[8] O. Kouropteva, O. Okun, A. Hadid, M. Soriano, S. Marcos and M. Pietikäinen, "Beyond locally linear embedding algorithm," *Technical Report MVG-01-2002*.

[9] O. Kouropteva, O. Okun and M. Pietik, "Incremental Locally Linear Embedding Algorithm," *SCIA*, vol.3540, pp. 521-530, 2005.

[10] L. K. Saul and R. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *The Journal of Machine Learning Research*, vol.4, pp. 119-155, 2003.

[11] L. Zhang, Y. Q. Leng and J. W. Yang, "Orthogonal Matching Pursuit-Based Incremental Locally Linear Embedding Algorithm," *International Journal of Autonomous and Adaptive Communications Systems*, 2013.

[12] S. Chen, S. A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J.Contr.*, vol.5, pp. 1873-1896, 1998.

[13] Y. C. Pati, R. Rezaiifar and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," *Proc. Record of the Twenty-Seventh Asilomar Conference. Signal, Systems and Computers*, vol. 1, pp. 40-44, 1993.

[14] D. L. Donoho, Y. Tsaig, I. Drori and J. L. Starck, "Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit," *IEEE Trans. on Information Theory*, vol. 58, pp. 1094-1121, 2012.

[15] L. Zhang and W. Zhou, "On the sparseness of 1-norm support vector machines," *Neural Networks*, vol. 23, pp. 373-385, 2010.

[16] P. M Murphy and D. W Aha(1992) UCI machine learning repository, http://archive.ics.uci.edu/ml/

[17] Y. LeCun, C. Cortes and C. J. C Burges (1998) The MNIST database of handwritten digits, http://yann.lecun.com/exdb/mnist/

[18] H. Tony and S. Ferdinando (1995) Real-Time Hand Tracking and Gesture Recognition Using Smart Snakes. Technical Report 95.1, http://www.cl.cam.ac.uk/Research/DTG/publications/public/files/tr.95.1.ps.Z

[19] K. C. Lee, J. Ho and D. Kriegman (2001) Nine Points of Lights: Acquiring Subspaces for Face Recognition under Variable Lighting. CVPR pp. 519–526

[20] N. Goel, G. Bebis and A. Nefian, "Face recognition experiments with random projections," *in: SPIE Conference on Biometric Technology for Human Identi- fication*, 2005.

2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# Feature Ensemble Learning based on Sparse Autoencoders for Image Classification

Yaping Lu
School of Computer Science and Technology & Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, Jiangsu, China
20134227010@stu.suda.edu.cn

Li Zhang
School of Computer Science and Technology & Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, Jiangsu, China
zhangliml@suda.edu.cn

Bangjun Wang
School of Computer Science and Technology & Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, Jiangsu, China
wangbanjun@suda.edu.cn

Jiwen Yang
School of Computer Science and Technology & Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215006, Jiangsu, China
jwyang@suda.edu.cn

***Abstract*—Deep networks are well known for their powerful function approximations. To train a deep network efficiently, greedy layer-wise pre-training and fine tuning are required. Typically, pre-training, aiming to initialize a deep network, is implemented via unsupervised feature learning, with multiple feature representations generated. However, in general only the last layer representation is to be employed because of its abstraction and compactness being the best with comparisons to the ones of lower layers. To make full use of the representations of all layers, this paper proposes a feature ensemble learning method based on sparse autoencoders for image classification. Specifically, we train three softmax classifiers by using the representations of different layers, instead of one classifier trained by applying the last layer representation. Of the three softmax classifiers, two are obtained by training stacked auto-encoders with fine tuning, and the other one is obtained by directly using a concatenation of two representations. To improve accuracy and stability of a single softmax classifier, the ensemble of multiple classifiers is considered, and some Naive Bayes combination rules are introduced to integrate the three classifiers. Experimental results on the MNIST and COIL datasets are presented, with comparisons to other classification methods.**

***Keywords—deep network; feature representation; feature ensemble; autoencoder; softmax; Naive Bayes***

## I. INTRODUCTION

Deep networks can compactly represent a significantly larger set of highly nonlinear and highly varying functions than shallow networks. Consequently, there has been significantly recent interest in multilayered or deep models for representation of general data [1], such as deep belief networks (DBNs) [2], convolutional restricted Boltzmann machines (RBMs) [3] and hierarchical sparse autoencoders [4], with a particular focus on imagery and audio signals. For these deep models, upper layers are supposed to represent high-level abstractions that explain the input observation, whereas lower layers extract low-level features from it [5]. While the theoretical benefits of deep networks in terms of their compactness and expressive power have been appreciated for many decades, until recently it was not clear how to train such deep networks [6]. Since gradient-based optimization starting from random initialization appears to often get stuck in poor solutions [5]. In 2006, Hinton et al. proposed a fast, greedy layer-wise unsupervised learning algorithm to train DBNs efficiently [7], of which the key point is that, first train one layer at a time in a greedy way, namely pre-training, and then tune the whole network in terms of the final criterion, namely fine tuning. By means of this strategy, a satisfied and stable result is obtained for deep networks. As a way to implement pre-training, unsupervised feature learning plays a very important role in the training of deep networks, and is often used to produce pre-processors and feature extractors for image analysis systems [8]. In general, multiple feature representations would be generated when training a deep network, whereas only the last layer representation is to be employed because of its abstraction and compactness being the best with comparisons to the ones of lower layers.

Sparse autoencoder (SAE) [9], being a kind of model for unsupervised feature learning, is an autoencoder, imposed with a sparseness constraint on the hidden units, that tries to learn an approximation to the identity function so as to output is similar to input. Typically, a representation, instead of raw data, learned via a sparse autoencoder is used as the input to train a classifier. Softmax regression [10] is a common classifier that generalizes logistic regression to classification problems where the class label can take on more than two possible values. A deep network that combines multiple sparse autoencoders and a softmax classifier as a whole network is called stacked autoencoders [11], on which we perform fine tuning so as to obtain a final classifier with higher performance.

It is well known that an ensemble of multiple classifiers is widely considered to be an effective technique for improving accuracy and stability, with comparisons to a single classifier. However, to ensure to get better performance, there are two important issues needed to be taken into account, namely, one being the diversity and accuracy of each classifier, and the other being the combination rules or fusion rules [12]. There

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

are two methods to implement the diversity for ensemble learning [13], one of which is to train multiple classifiers by employing different feature sets [14, 15], just being the method used in this paper. Moreover, the accuracy of an individual classifier is also very important, since the poor classifiers can suppress correct predictions of good classifiers [12]. Then, the final issue is the combination rules of multiple classifiers. Provided the labels are available, a simple voting rule [16] and the Naive Bayes combination methods including MAX rule, MIN rule and AVG rule can be used.

As discussed above, only the representation of last layer, instead of raw data, is used to train a classifier. However, it is true that multiple representations can be obtained when a deep network is trained. Is there a way to utilize all the representations, instead of the last one? Thus, in this paper, as an improvement, we propose a feature ensemble learning method based on sparse autoencoders for image classification. Specifically, we take fully all these representations learned by pre-training two sparse autoencoders into account in the way that utilizing the two representations to train three softmax classifiers, namely, the so-called feature ensemble. In detail, the first and the second softmax classifiers are achieved by respectively exploiting the corresponding two representations learned via sparse autoencoders, of course, with fine tuning used for the classification performance, and as to the third softmax classifier, we directly employ the concatenation of two representations as the input of softmax classification model. Note that although the stacked autoencoder that corresponds to the first representation is a shallow network, we would still perform fine tuning on it, since this can significantly improve the performance of classifier. The reason why we choose the present architecture is that we expect to use the way of feature ensemble to fully utilize all the representations so as to finally improve the classification performance. As for the deep motivations, it is because the representations with different abstraction levels have different advantages for classification, so the feature ensemble method can integrate this advantages so as to improve the performance finally.

After we have finished the training of three softmax classifiers via feature ensemble, next, what we need to consider is that how to integrate these classifiers so as to improve recognition performance efficiently. Here we implement diversity for classifiers by employing different feature sets to train the three softmax classifiers, and in some ways fine tuning and concatenation make the accuracies of the three softmax classifiers be guaranteed. Therefore, combining the three softmax classifiers to predict new data is feasible and promising. Finally, we choose the Naive Bayes combination methods as our combination rule, including MAX rule, MIN rule and AVG rule.

The remainder of the paper is organized as follows. In Section 2, we overview some related works about sparse autoencoder, softmax regression and deep network. The detailed processes of training the three softmax classifiers are discussed in Section 3. Experimental results on the MNIST and COIL datasets are presented in Section 4, with conclusions provided in Section 5.

## II. RELATED WORKS

In this section, we briefly introduce some models used in our paper, including sparse autoencoder used to obtain feature representations, softmax regression used to classify images, and deep network composed of the previous two models.

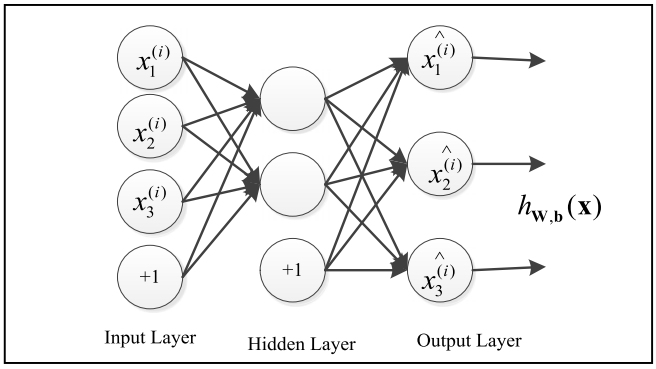



Fig. 1. An autoencoder. The circles labeled "+1" are bias units corresponding to the intercept terms. $h_{\mathbf{W},\mathbf{b}}(\mathbf{x})$ is an activation function, which the autoencoder tries to learn so as to the output $\hat{x}_j^{(i)}$ is similar to the input $x_j^{(i)}$. In general, we call the process from input layer to hidden layer as "encode", from hidden layer to output layer as "decode".

### A. Sparse Autoencoder

Sparse autoencoder is an autoencoder, an unsupervised learning method that applies back propagation and sets the target values to be equal to the inputs, imposed with a sparseness constraint on the hidden units. In other words, a sparse autoencoder is trying to learn an approximation to the identity function, so as to output is similar to input.

Suppose the $i$ th sample $\mathbf{x}^{(i)}$ with label $y^{(i)}$ is represented as $(\mathbf{x}^{(i)}, y^{(i)})$ where $\mathbf{x}^{(i)} \in \mathbb{R}^d, y^{(i)} \in \{1,2,...,k\}$, and $d$ denotes the dimensionality of samples and $k$ is the number of classes. Let $x_j^{(i)}, j \in \{1,2,...,d\}$ represent the $j$ th feature for the sample $\mathbf{x}^{(i)}$. $m$ samples are denoted as $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^m$. Assume that $d$ is 3, and the number of hidden units is 2, and $\hat{x}_j^{(i)}$ denotes the $j$ th output of output layer for the $i$ th sample, so the illustration of an autoencoder is given in Fig. 1.

For a sparse autoencoder, its cost function $J(\mathbf{W},\mathbf{b})$ requires minimizing

$$J(\mathbf{W},\mathbf{b}) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{\mathbf{W},\mathbf{b}}\left(\mathbf{x}^{(i)}\right)-\mathbf{x}^{(i)}\right\|^2\right)\right] + \frac{\lambda_1}{2}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}\left(W_{ji}^{(l)}\right)^2 + \beta\sum_{j=1}^{s_2} KL\left(\rho \,\|\, \hat{\rho}_j\right) \quad (1)$$

where $h_{\mathbf{W},\mathbf{b}}(\mathbf{x})$ is an activation function, $\mathbf{W}$ and $\mathbf{b}$ denote the weights and biases of network respectively, also being our optimal parameters. The first term tries to minimize the difference between the output and the input. The second term represents weight decay term in order to avoid over-fitting,

where $\lambda_1$ is a weight decay parameter, and $L$ is the number of autoencoder network layers (in the context of this paper, $L$ is set to be 3), and $s_l$ represents the number of units for the $l$ th layer. $W_{ji}^{(l)}$ denotes the weight between the $i$ th unit of layer $l$ and the $j$ th unit of layer $l+1$, and $b_i^{(l)}$ is the bias associated with unit $i$ in layer $l+1$. The last term is a sparse penalty term where $\beta$ controls the weight of this term, and $\rho$ is a sparseness parameter, and $\hat{\rho}_j$ denotes the average activation of hidden unit $j$, namely $\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(i)}\right]$ in which $a_j^{(i)}$ represents the activation of the $j$ th unit of hidden layer for the $i$ th sample, and $KL(\cdot)$ is the Kullback-Leible (KL) divergence given by

$$KL\left(\rho \,\|\, \hat{\rho}_j\right) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho)\log \frac{1-\rho}{1-\hat{\rho}_j}.$$

Typically, $\rho$ is set to be a small value close to 0. In other words, we would like the average activation of each hidden unit $j$ to be close to 0. To satisfy this constraint, the hidden unit's activations must mostly be near 0. To achieve this, we add an extra penalty term (namely the last term) to our cost function that penalizes $\hat{\rho}_j$ deviating significantly from $\rho$. The reason why we choose KL divergence as our penalty term is that KL divergence is a standard function for measuring how different two different distributions are, and it has the property that $KL\left(\rho \,\|\, \hat{\rho}_j\right) = 0$ if $\hat{\rho}_j = \rho$, and otherwise it increases monotonically as $\hat{\rho}_j$ diverges from $\rho$. Thus, minimizing this penalty term has the effect of causing $\hat{\rho}_j$ to be close to $\rho$.

The optimization problem (1) can be solved by using the back propagation algorithm, and the L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) method is used to update the parameters $\mathbf{W}$ and $\mathbf{b}$.

### B. Softmax Regression

Softmax Regression is a classification model generalizing logistic regression to classification problems where the class label $y$ can take on more than two possible values. Again, suppose $d$ and $k$ are 3, then the softmax model is as Fig. 2.

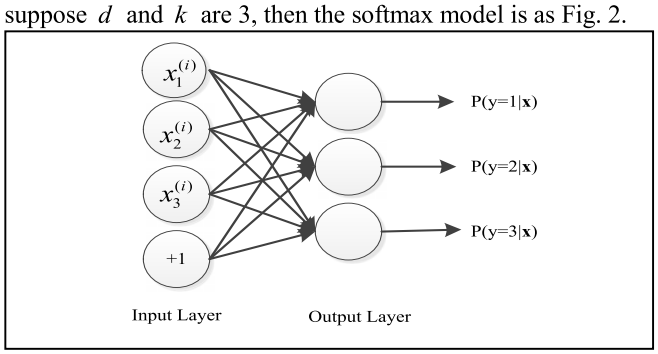



Fig. 2. Softmax model. This model outputs the possibility of each class given a sample $\mathbf{x}^{(i)}$.

For a softmax classifier, the probability output function has the form

$$h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = \begin{bmatrix} p(y^{(i)}=1 \mid \mathbf{x}^{(i)};\boldsymbol{\theta}) \\ p(y^{(i)}=2 \mid \mathbf{x}^{(i)};\boldsymbol{\theta}) \\ \vdots \\ p(y^{(i)}=k \mid \mathbf{x}^{(i)};\boldsymbol{\theta}) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\boldsymbol{\theta}_j^T \mathbf{x}^{(i)}}} \begin{bmatrix} e^{\boldsymbol{\theta}_1^T \mathbf{x}^{(i)}} \\ e^{\boldsymbol{\theta}_2^T \mathbf{x}^{(i)}} \\ \vdots \\ e^{\boldsymbol{\theta}_k^T \mathbf{x}^{(i)}} \end{bmatrix}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{k\times(d+1)}$ is the parameters of softmax model (bias unit considered together), and $\boldsymbol{\theta}_j$ means the $j$ th row of matrix $\boldsymbol{\theta}$ which could be found by minimizing the following cost function $J(\boldsymbol{\theta})$:

$$J(\boldsymbol{\theta}) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{y^{(i)}=j\} \log \frac{e^{\boldsymbol{\theta}_j^T \mathbf{x}^{(i)}}}{\sum_{l=1}^{k} e^{\boldsymbol{\theta}_l^T \mathbf{x}^{(i)}}}\right] + \frac{\lambda_2}{2}\sum_{i=1}^{k}\sum_{j=1}^{d+1}\theta_{ij}^2 \quad (2)$$

where $\theta_{ij}$ denotes the weight between the $j$ th unit of input layer and the $i$ th unit of output layer, and $\lambda_2$ is also a weight decay parameter, and $1\{y^{(i)}=j\}$ is the indicator function with

$$1\{y^{(i)}=j\} = \begin{cases} 1, & if \;\; y^{(i)}=j \\ 0, & if \;\; y^{(i)}\neq j \end{cases}.$$

Once we have the cost function of the softmax model, the similar minimization procedure to SAE model is performed, resulting in the optimal parameters $\boldsymbol{\theta}$. For more details, refer to [10].

### C. Deep Network

Recall that after using the feature representations learned by pre-training a sparse autoencoder as the input to train a softmax model, to enhance the classification performance, a fine tuning process is done on the whole network, typically being a deep network.

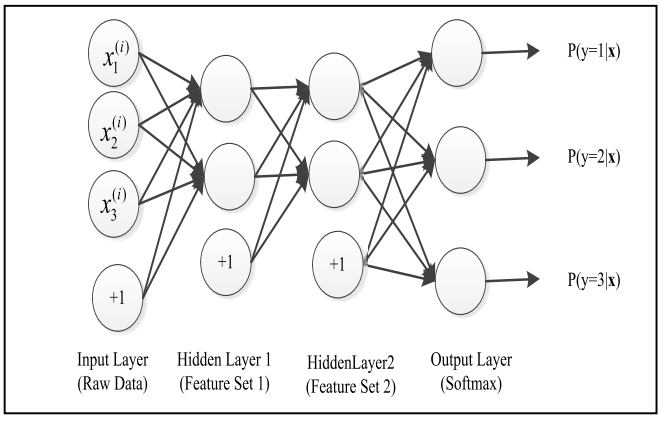



Fig. 3. Stacked Autoencoders. Here, as we can see, the whole deep network contains four layers, specifically, input layer with three units, two hidden layers with two units for each, and output layer with three units. Namely, $d$ is assumed to be 3 as well as $k$.

Generally, if a neural network contains more than two hidden layers, then we call it a deep network. Trying to train such a deep network with an efficient method is called deep learning. As we know, in 2006, Hinton et al. has proposed a feasible and satisfying strategy to implement the training on DBNs in [7]. Instead of DBNs, here, we would train or fine tune stacked autoencoders network, also being a deep network. We give an illustration about this model in Fig. 3.

If we want to fine tune such a deep network, essentially a multilayered softmax classifier with the cost function of (2), the parameters learned by pre-training, instead of random generation, are selected to initialize the deep model. What's more important, when calculate the partial derivatives of cost function with respect to weights and biases, pay attentions to that the activation function of two hidden layers is different from output layer. After fine tuning is finished, a better and more stable softmax classifier is generated.

## III. FEATURE ENSEMBLE LEARNING

Generally, only the feature representation of last layer, instead of raw data, is used to train a classifier. To utilize the representations of all layers, we integrate them by training multiple classifiers and employ some combination rules to make the final decision. Specifically, we discuss the case of two sparse autoencoders. Actually, our method can be extended to more than two.

As discussed above, in this paper, we would train two sparse autoencoders so as to get two representations of the raw data. The first representation is used as the input of the second sparse autoencoder in order to get the second representation. Then, the two representations are concatenated to form the third representation of the raw data. Now, we have all the conditions to train the three classifiers corresponding to the three representations. Note that for the first and second classifiers, we need to perform fine tuning on the whole network (stacked autoencoders) composed of the input layer, feature representation layers and the softmax layer so as to improve the performance of the classifiers. Even if the whole network for the first classifier is a shallow network, we would still do so. For the clear understanding of the whole process, Fig. 4 gives an overview of the process of the feature ensemble. Note that the fine tuning process does not illustrate in the Fig. 4. Next, detailed descriptions for the training of three classifiers are to be presented, respectively.

### A. Softmax 2 Classifier

To make the whole system easy to understand, we shall begin with the training of the softmax 2 classifier. According to Fig. 4, we can get two kinds of feature representations corresponding to raw input. Specifically, we first train a sparse autoencoder (SAE 1 in Fig. 4) with $HS_1$ hidden units on raw input $\mathbf{x} \in \mathbb{R}^d$ in order to obtain the parameters between the input layer and the hidden layer, represented as $\mathbf{W}^{(1,1)} \in \mathbb{R}^{HS_1 \times d}$ and $\mathbf{b}^{(1,1)} \in \mathbb{R}^{HS_1 \times 1}$. Then the raw data $\mathbf{x}$ can be transformed into its first feature representation $\mathbf{f}_1$ in the Feature Set 1, which could be represented by

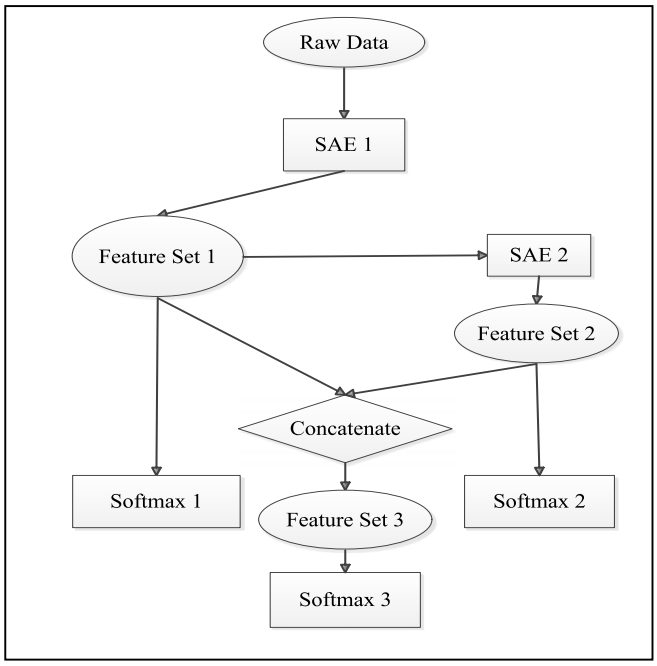



Fig. 4. Overview of Feature Ensemble, where rectangle represents the model such as sparse autoencoder or softmax, and ellipse denotes the input or output of a model, and the rhombus means some operation.

$$\mathbf{f}_1 = h_{\mathbf{W}^{(1,1)},\mathbf{b}^{(1,1)}}(\mathbf{W}^{(1,1)}\mathbf{x} + \mathbf{b}^{(1,1)}).$$

Next, again, we train the other sparse autoencoder (SAE 2 in Fig. 4) with $HS_2$ hidden units on the Feature Set 1 so as to get the parameters between hidden layers, represented as $\mathbf{W}^{(2,1)} \in \mathbb{R}^{HS_2 \times HS_1}$ and $\mathbf{b}^{(2,1)} \in \mathbb{R}^{HS_2 \times 1}$. Similarly, $\mathbf{x}$ can be transformed into its second feature representation $\mathbf{f}_2$ in the Feature Set 2, which could be represented by

$$\mathbf{f}_2 = h_{\mathbf{W}^{(2,1)},\mathbf{b}^{(2,1)}}(\mathbf{W}^{(2,1)}\mathbf{f}_1 + \mathbf{b}^{(2,1)}).$$

In the end, we would train a classification model Softmax 1 (see Fig. 4) on the input Feature Set 2 to obtain the optimal parameters $\boldsymbol{\theta}$ in (2), denoted as $\boldsymbol{\theta}_2 \in \mathbb{R}^{k \times (HS_2+1)}$.

Now, by the pre-trainings above, we have got the first classifier Softmax 2 based on the Feature Set 2. However, its classification performance is unsatisfying for the way of greedy layer-wise pre-training, so that we need to operate fine tuning on the whole network consisting of the input layer of SAE 1, the hidden layer of SAE 1, the hidden layer of SAE 2 and the output layer of Softmax 1 in Fig. 3. Before fine tuning, the initial parameters between two layers of the whole network from left to right are set to be $\{\mathbf{W}^{(1,1)}, \mathbf{b}^{(1,1)}\}$, $\{\mathbf{W}^{(2,1)}, \mathbf{b}^{(2,1)}\}$ and and $\boldsymbol{\theta}_2$. Then, with the raw data as input, the back propagation algorithm is employed, based on which the final optimal parameters about the whole network or deep network are calculated. Finally, we obtain the first classifier Softmax 2. Recall that the activation functions for two hidden layers and output layer are different so that we should pay attentions to the process of back propagation.

### B. Softmax 1 Classifier

After the first classifier Softmax 2 has been achieved, we have got what it takes to train the second one. In other words, on the basis of the first classifier, all the parameters employed to train the second one are available. Specifically, the Feature Set 1 is used as the input to train Softmax 1 classifier (see Fig. 4), resulting in the model optimal parameters $\boldsymbol{\theta}$ in (2), denoted as $\boldsymbol{\theta}_1 \in \mathbb{R}^{k\times(HS_1+1)}$, obtained.

So far, the second classifier Softmax 1 is generated, corresponding to the Feature Set 1. Nevertheless, aiming for the better classification performance, we also perform the fine tuning on the whole network composed of the input layer of SAE 1, the hidden layer of SAE 1 and the output layer of Softmax 1, just like the model in Fig. 3 without the second hidden layer. Before fine tuning, the initial parameters between the input layer and the hidden layer, and the ones between the hidden layer and the output layer are set to be $\{\mathbf{W}^{(1,1)}, \mathbf{b}^{(1,1)}\}$ and $\boldsymbol{\theta}_1$ respectively. After fine tuning, the second classifier Softmax 1 can be achieved finally. Also, we must be careful to the activation functions of the hidden layer and output layer when executing the back propagation.

Here, as we can see, the whole network fine-tuned with three layers is a shallow network instead of a deep network. Although the fine tuning is not employed to a shallow network generally, we still do it just because this is able to improve the accuracy and stability of the second classifier, so as to contribute to the performance of the following ensemble of multiple classifiers [12].

### C. Softmax 3 Classifier

The two classifiers Softmax 1 and Softmax 2 with fine tuning above are trained on the Feature Set 1 and Feature Set 2, respectively. After taking into full account the representations, namely Feature Set 1 and Feature Set 2, we use the concatenation of these two representations as the input of our third classifier Softmax 3 that is to be trained (see, Fig.4). Specifically, Feature Set 1 and Feature Set 2 are concatenated into one column, denoted as Feature Set 3. Namely, the corresponding feature of $\mathbf{x}$ can be represented as $\left[\mathbf{f}_1^T, \mathbf{f}_2^T\right]^T$.

Then, the Softmax 3 is trained by using the softmax model like Fig. 2 with the representation Feature Set 3 being the input. After minimizing the cost function, we obtain the last classifier, Softmax 3 with parameters $\boldsymbol{\theta}_3 \in \mathbb{R}^{k\times(HS_1+HS_2+1)}$.

### D. Naive Bayes Combination Methods for Voting

After all the work above done, the final thing is to give the voting rule for the ensemble of the three softmax classifiers. Here, the Naive Bayes combination methods are considered, in which assume that individual classifiers are mutually independent; hence the name "naive" [16, 17]. Three naive Bayes methods, namely MAX rule, MIN rule and AVG rule, are given below.

For a new sample $\mathbf{x}$ that is to be tested, when its label $y$ takes on the different possible values $j \in \{1,2,...,k\}$, we can get the corresponding predicted probabilities for the softmax classifier $n \in \{1,2,3\}$, here denoted as $\mathbf{P}_{nj}(\mathbf{x})$ which can be expressed as

$$\mathbf{P}_{nj}(\mathbf{x}) = \begin{cases} h_{\boldsymbol{\theta}_1}(\mathbf{f}_1), & if\ n=1 \\ h_{\boldsymbol{\theta}_2}(\mathbf{f}_2), & if\ n=2 \\ h_{\boldsymbol{\theta}_3}\left(\left[\mathbf{f}_1^T, \mathbf{f}_2^T\right]^T\right), & if\ n=3 \end{cases}.$$

The final value of label $y$ is determined by the following rules.

*1) MAX Rule*

Given a new sample $\mathbf{x}$, the MAX rule is to get its label $y$ by

$$y = \arg \max_{j\in\{1,2,...,k\}} \max_{n\in\{1,2,3\}} \mathbf{P}_{nj}(\mathbf{x}). \quad (3)$$

*2) MIN Rule*

Given a new sample $\mathbf{x}$, the MIN rule is to get its label $y$ by

$$y = \arg \max_{j\in\{1,2,...,k\}} \min_{n\in\{1,2,3\}} \mathbf{P}_{nj}(\mathbf{x}). \quad (4)$$

*3) AVG Rule*

Given a new sample $\mathbf{x}$, the AVG rule is to get its label $y$ by

$$y = \arg \max_{j\in\{1,2,...,k\}} \sum_{n=1}^{N} \mathbf{P}_{nj}(\mathbf{x}) / N \quad (5)$$

where $N = 3$ here.

## IV. Experimental Results

In the examples below, we take into account two datasets, MNIST dataset [18] and COIL dataset [19], to verify our strategy proposed here. Moreover, K-nearest neighbor (KNN) method is used as the comparison under the same datasets.

All numerical experiments are performed on the personal computer with a 3.40GHz Intel(R) Core(TM) i3-3240 CPU and 3.15G bytes of memory. This computer runs on Windows 7, with MATLAB 8.1.0.

### A. Parameter Settings

As we can see, the sparse autoencoder construction in (1) and the softmax classifier construction in (2) may appear relatively complex, while the number of parameters that need to be set is not particularly large, and they are initialized so as to allow the sparse autoencoder to get good filters. Specifically, for the two sparse autoencoders $\lambda_1 = 3e-3$, $\beta = 3$, $\rho = 0.1$, $L = 3$, $HS_1 = 100$, $HS_2 = 100$ and the activation function is set to be sigmoid function, namely $h_{\mathbf{W},\mathbf{b}}(\mathbf{x}) = 1/(1+e^{-(\mathbf{W}\mathbf{x}+\mathbf{b})})$ with $\mathbf{W}$ and $\mathbf{b}$ initialized randomly for the first time, and for the softmax classifier $\lambda_2 = 1e-4$ and $\boldsymbol{\theta}$ is also initialized randomly for the first time. The same parameters are used in all examples and in all layers of the "deep" network, essentially a

deep softmax classifier. The values of $m$, $k$ and $d$ depend on the specific dataset, and these parameters are specified for the specific examples. In all experiments, we consider the L-BFGS as optimization algorithm with 400 iterations for sparse auto-encoders and deep networks, and 100 iterations for softmax classifiers.

### B. Experiments on the MNIST Dataset

Consider the widely studied MNIST data, which has a total of 60,000 training images and 10,000 testing images, each $28 \times 28$, for handwritten digits 0 through 9 (this means $k = 10$). In Fig. 5, we randomly give 100 images of the MNIST dataset.

In our experiments, 23,000 sequential images from the beginning of the whole training set are selected as our unlabeled training set to pre-train our two sparse autoencoders so as to get the optimal parameters $\{\mathbf{W}^{(1,1)}, \mathbf{b}^{(1,1)}\}$ and $\{\mathbf{W}^{(2,1)}, \mathbf{b}^{(2,1)}\}$ corresponding to the two representations, and the last 20,000 images of the whole training set are selected as training set to train our softmax classifiers. Note that, before training the classifier, the training set with labels needs to be transformed to another corresponding representation via the parameters $\{\mathbf{W}^{(1,1)}, \mathbf{b}^{(1,1)}\}$ and $\{\mathbf{W}^{(2,1)}, \mathbf{b}^{(2,1)}\}$ learned through the unlabeled data.

After the three softmax classifiers have been trained with the unlabeled training set and labeled training set above, we use the whole 10,000 testing images of MNIST data to test them and report the recognition rates for these three classifiers. In our method, by the Naive Bayes combination methods discussed above, we can obtain the corresponding ensemble results. For comparison, KNN classifier is performed by using the same training set with labels and the same testing set. We vary the parameter K from 1 to 10, and get the best recognition rate of KNN classifier with K=4. All experimental results are listed in TABLE I.

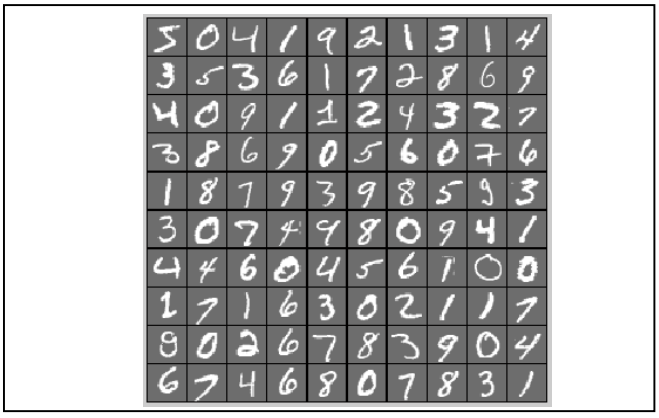


Fig. 5. Images from MNIST dataset.

As we can see in TABLE I, our methods get better recognition rates with comparison to other 4 classifiers. Moreover, when we utilize the Naive Bayes rules to make decision, the AVG rule performs more efficiently than the MIN rule and the MAX rule. For a new sample, if Softmax 2 misclassifies it, while the other 2 classifiers get the expected prediction, then the AVG method has a good chance of correcting the output of Softmax 2, and finally improves the performance.

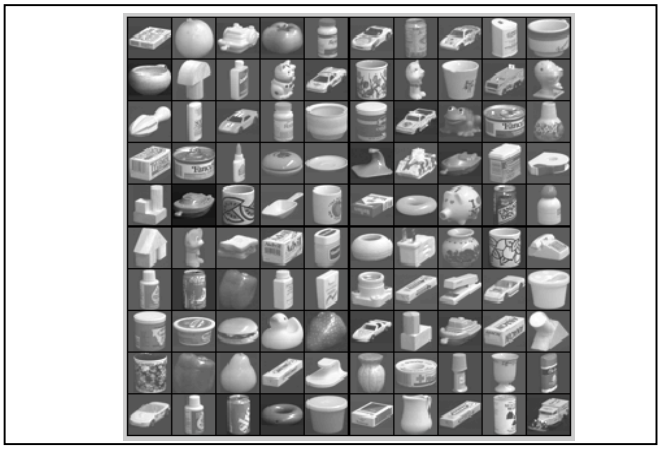


Fig. 6. Images from COIL dataset.

### C. Experiments on the COIL Dataset

The COIL dataset is considered next. It has a total of 7,200 examples, each 32×32, with 100 classes (this means $k = 100$), 72 examples for each class. The original images of COIL dataset are color ones, and here we change them into grayscale ones. Fig. 6 shows 100 images from the COIL dataset, one for each class.

In our experiments, since the limitation of the number of examples, we select 6,000 samples, namely 60 examples for each class, as our training set to pre-train the two sparse autoencoders and train the three classifiers. In other words, being different from above method, here the unlabeled training set and the labeled training set are the same. The remainder 1,200 examples, namely 12 examples for each class, are used as the test set. Similar to the MNIST dataset, we report the results of KNN, three softmax classifiers and their ensemble in TABLE II. From TABLE II, we have the same conclusion as TABLE I. The feature ensemble methods achieve the best classification performance.

For the two tables above, since the representations learned by SAEs have different abstraction levels, the diversity for the three softmax classifiers is guaranteed. As a result, the Naive Bayes methods can be employed to improve the recognition performance. Specifically, Feature Set 2 in Fig. 4 is more abstract than Feature Set 1 so that Softmax 2 being a deep network should work better than Softmax 1 that is a shallow network, but we have the opposite result in our experiments. This is because the number of unlabeled training set for SAE is not enough. Since a deep network with fine tuning is more powerful in the function approximation than a softmax regression model, Softmax 2 performs better than Softmax 3. Besides, although Feature Set 1 is related to Feature Set 2, we still consider to use their concatenation to train our third classifier because compared with the classifier that trained directly with only one feature set as input, Softmax 3 indeed has a better recognition rate so as to improve the whole performance of feature ensemble.

TABLE I. CLASSIFICATION PERFORMANCE FOR THE MNIST DATA SET WITH COMPARISONS TO OTHER METHODS

| Classifier | KNN(K=4) | Softmax 1 | Softmax 2 | Softmax 3 | Our Method | | |
|---|---|---|---|---|---|---|---|
| | | | | | *MAX* | *MIN* | *AVG* |
| Accuracy (%) | 95.92 | 96.07 | 95.87 | 95.22 | 96.53 | 96.50 | 96.70 |

TABLE II. CLASSIFICATION PERFORMANCE FOR THE COIL DATA SET WITH COMPARISONS TO OTHER METHODS

| Classifier | KNN(K=4) | Softmax 1 | Softmax 2 | Softmax 3 | Our Method | | |
|---|---|---|---|---|---|---|---|
| | | | | | *MAX* | *MIN* | *AVG* |
| Accuracy (%) | 84.00 | 89.41 | 86.75 | 86.75 | 90.50 | 91.42 | 91.67 |

## V. CONCLUSION

This paper proposes a feature ensemble method based on sparse autoencoders for image classification. In our method, there are three softmax classifiers, each of which is to train a different feature set. Finally, some Naive Bayes combination rules can be used for ensemble the outputs of these classifiers. From the results of the experiments on the MNIST dataset and COIL dataset, we can see that feature ensemble learning based on sparse autoencoder has better classification performance, with comparisons to KNN classifier and the other three softmax classifiers.

Although, here, we need to train multiple classifiers with the cost of time, fortunately, this can be tolerated for neural network (here, sparse autoencoder, softmax model and deep network) since the prediction for a new sample would be fast. For the future work, we may try to improve the representation of Feature Set 2 to the raw data so as to enhance the performance of Softmax 3, since the Feature Set 2 is obtained by greedy layer-wise pre-training and can't reconstruct to the raw data well. Additionally, the proposed algorithm is only compared with individual classifiers, but, for the future work, it is indeed promising to compare the results with the ones of using other classifiers with the same ensemble strategy.

## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61373093 and 61033013, by the Natural Science Foundation of Jiangsu Province of China under Grant Nos. BK2011284 and BK201222725, by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No.13KJA520001, and by the Qing Lan Project.

## REFERENCES

[1] B. Chen, G. Polatkan, G. Sapiro, D. Blei, D. Dunson, and L. Carin, "Deep learning with hierarchical convolutional factor analysis," IEEE Tran. Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1887-1901, 2013.

[2] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol. 313, no.5786, pp. 504-507, 2006.

[3] M.R.M. Norouzi and G. Mori, "Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning," Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2009.

[4] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," Proc. Int'l Conf. Machine Learning, 2008.

[5] Y. Bengio, P. Lamblin, D. Popovici and H. Larochelle, "Greedy layer-wise rraining of deep networks," NIPS, 2007.

[6] Y. Bengio, "Learning deep architectures for AI," Foundations and Trends in Machine Learning 21(6), pp. 1601-1621, 2009.

[7] G. Hinton, S. Osindero, and Y. The, "A fast learning algorithm for deep belief nets," Neural Computation, 18, pp. 1527-1554, 2006.

[8] M. A. Ranzato, C. Poultney, S. Chopra, and Y LeCun, "Efficient learning of sparse representations with an energy-based model," NIPS, 2006.

[9] A. Ng, "Sparse autoencoder," CS294A Lecture Notes for Stanford University, 2011.

[10] A. Ng, "Generalized linear models," CS229 Lecture Notes for Stanford University, part 3, 2003.

[11] A. Ng, J. Ngiam, C.Y. Foo, Y. Mai, and C. Suen, "UFLDL tutorial: building deep networks for classification," an online tutorial, 2013.

[12] L. Zhang and W.D. Zhou, "Sparse ensembles using weighted combination methods based on linear programming," Pattern Recongnition, 44, pp. 97-106, 2011.

[13] E.K. Tang, P.N. Suganthan, and X. Yao, "An analysis of diversity measures, Machine Learning," Machine Learning, 65, pp. 247-271, 2006.

[14] T.K. Ho, J.J. Hull, and S.N. Srihari, "Decision combination in multiple classifier systems," IEEE Transactions on Pattern Analysis and Machine Intelligence, 16, pp. 66-75, 1994.

[15] S.D. Bay, "Combining nearest neighbor classifiers through multiple feature subsets," Proceeding of the 15th International Conference on Machine Learning, pp. 37-45, 1998.

[16] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On combing classifiers," IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, pp. 226-239, 1998.

[17] L.I. Kuncheva, "Combining pattern classifier: method and algorithms," Wiley, Hoboken, NJ, 2004.

[18] "http://yann.lecun.com/exdb/mnist/".

[19] S.A. Nene, S.K. Nayar, and H. Murase, "Columbia object image library( COIL-100)," Techniqual Report, CUCS-006-96, Department of Comp. Science, Columbia University, 1996.

2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

# Solving Unbalanced Problems in Similarity Learning using SVM Ensemble

Peipei Xia
School of Computer Science and Technology
& Provincial Key Laboratory for
Computer Information Processing Technology
Soochow University
Suzhou 215006, Jiangsu, China
Email: 20124227054@suda.edu.cn

Li Zhang
School of Computer Science and Technology
& Provincial Key Laboratory for
Computer Information Processing Technology
Soochow University
Suzhou 215006, Jiangsu, China
Email: zhangliml@suda.edu.cn

***Abstract*—Similarity learning is one of the most fundamental notions in machine learning and pattern recognition. In real-world problems, the number of the paired-samples in similarity set is far less than the ones in dissimilarity set. In other word, there is an unbalanced problem in the paired-samples of similarity learning. This paper presents a scheme of SVM ensemble to solve it. In our scheme, we randomly select some of samples to construct paired-samples, not producing all the paired-samples, and introduces multiple classifiers to obtain higher stability and reliability. As a result, the SVM ensemble can effectively decrease the number of paired-samples in similarity learning and solve the unbalanced data learning to some degree. In the experiments, the SVM ensemble is compared with some classic unbalanced learning algorithms. The results on classification tasks show that the SVM ensemble gains better performance.**

## I. Introduction

In machine learning and pattern recognition, many classic algorithms adopt similarity or distance metric, such as $K$-NN (nearest neighbors) [1], SVM (Support Vector Machine) [2], [3], [4], etc. Traditionally, similarity or distance metric is preferentially appointed in these algorithms, such as Euclidean distance, which is independent of specific problems. The traditional way can not be applied to many complex tasks of information analysis, recognition, retrieval and others. Thus, similarity learning and distance metric is becoming a popular research subject in the field of machine learning, pattern recognition, image sciences, computer vision, etc.

In similarity learning, it requires to construct paired patterns (paired-samples), which are the objects for machines to learn. When solving real-world problems especially multi-class problems, it is quite obvious that most of the paired-samples are made up of dissimilar patterns, only a few ones consist of similar patterns. In other word, this is an unbalanced problem. It is well known that a balanced dataset provides improved overall classification performance compared to an unbalanced dataset for some classifiers [5], [6], [7]. Consequently, it is necessary to solve unbalanced problems in similarity learning.

There may exist two unbalanced situations in a dataset. The first type is between-class unbalance or class unbalance, in which case some classes have much more instances than others [8]. The other type is within-class unbalance or case unbalance, in which case some subsets have much fewer instances than other subsets in one class [9]. Here, by unbalanced problem, we mean the first one. By convention, in unbalanced dataset, we call the classes containing more instances the majority (common) class, while the ones containing fewer instances are called the minority (rare) classes.

Methods for solving unbalanced problems can be grouped into two categories, or the data level and algorithmic level. The main idea behind the methods at data level is to alter the distribution of data in order to provide a balanced dataset for classifiers. The methods at algorithmic level improve the performance by modifying algorithms themselves [8], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. We focus on the methods at data level which include some classical sampling methods [8]. Typically, sampling methods in unbalanced data learning is to modify the unbalanced dataset to get a relatively balanced distribution.

The simplest sampling methods are random over-sampling and random under-sampling. The former augments the minority class by exactly replicating the randomly selected examples from the minority class, while the latter randomly removes some examples from the majority class. The random over-sampling and under-sampling methods appear to be functionally equivalent since they both alter the size of the original dataset and can actually provide the same proportion of balance. However, this commonality is only superficial. Random over-sampling may make the decision regions of the learner smaller and more specific, leading to over-fitting [10]. Random under-sampling may cause classifiers to miss important concepts pertaining to the majority class. Thus, many modified sampling methods have been presented. Kubat et al. presented a heuristic under-sampling method which balances a dataset through just eliminating the noise and redundant examples from the majority class [11]. Synthetic sampling is another important part of sampling methods. The synthetic minority over-sampling technique (SMOTE) is a powerful method that has shown a great deal of success in various applications [12]. SMOTE generates new synthetic examples along the line between the minority examples and their randomly selected nearest neighbors. Han et al. presented two novel over-sampling methods based on SMOTE, or borderline-SMOTE1 and borderline-SMOTE2 [13]. The examples on the borderline and the ones nearby (called borderline examples in [13]) are more apt to the misclassified than the ones far from borderline, and more important for classification. Borderline-SMOTE

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

methods only strengthen the borderline minority examples. First, this method finds out the borderline minority examples; then, synthetic examples are generated from them and added to the original dataset. It shows that synthetic sampling methods are effective in dealing with unbalanced datasets. However, the data generation methods discussed above have a highly computational complexity. Noting that the essential problem of over-fitting is caused by random over-sampling, Mease et al. proposed a much simpler technique, over/under-sampling with jettering (JOUS-Boost) [10]. In each iteration of boosting, JOUS-Boost introduces independently and identically distributed (iid) noise into minority examples and generates new samples. This algorithm is relatively simple compared to synthetic sampling counterparts and also incorporates the benefits of boosted ensembles to improve performance. Based on the characteristics of the given data distribution, four $K$NN under-sampling methods were proposed, namely, NearMiss-1, NearMiss-2, NearMiss-3 and the "most distant" method [14].

The integration of sampling strategies with ensemble learning has also been studied. For example, the SMOTEBoost algorithm is based on the idea of integrating SMOTE with AdaBoost.M2 [15]. Specifically, SMOTEBoost introduces synthetic sampling in each boosting iteration. Another integrated approach, the DataBoost-IM method, combines the data generation techniques introduced in [17] with AdaBoost.M1 to achieve high predictive accuracy for the minority class without sacrificing accuracy on the majority class [16]. The Granular Support Vector Machines-Repetitive Under-sampling algorithm (GSVM-RU) was proposed in [18] to integrated SVM with under-sampling. The GSVM-RU method uses the SVM itself as a mechanism for under-sampling to sequentially develop multiple information granules with different information samples, which are later combined to develop a final SVM for classification.

The methods mentioned above could be directly used to solve the unbalanced problem in similarity learning. When applying these methods, it requires to generate an unbalanced problem first. In other words, all paired-samples which are unbalanced must be obtained from the original dataset, and then these methods are used for generating a balanced dataset. However, it is time-consuming for generating all paired-samples. Even if we generate all paired-samples, we still try to discard most of them by using these classical sampling methods. Thus, it costs a lot. In this paper, we apply SVM ensemble to solve unbalanced problem in similarity learning. In our scheme, we randomly select some of samples to construct paired-samples to decrease the number of paired-samples and balance the data distribution. The randomly selected samples would play an important role in the performance of SVM. To eliminate the randomness in selecting samples, ensemble learning is introduced here. Since selecting samples randomly could bring the diversity for ensemble learning, it is appropriate for introducing ensemble techniques.

The structure of this paper is organized as follows. Section II gives a brief introduction to unbalanced problems in similarity learning and reviews some classic under-sampling methods. Section III describes our proposed similarity learning method solving unbalanced problems using SVM ensemble in details. Section IV compares the proposed method, SVM ensemble, with other under-sampling methods and gives experimental results. Section V draws conclusions.

## II. Related work

In this section, we have a brief review on unbalanced problems in similarity learning and some classic sampling methods in unbalanced data learning.

### A. Unbalanced problem in similarity learning

Let the set of multi-class training samples be $X = \{\mathbf{x}_i,\ y_i\}_{i=1}^{l}$, where $\mathbf{x}_i \in R^d$ is the $i$th training sample, $y_i \in \{1, 2, \cdots, c\}$ is the label of $\mathbf{x}_i$, $l$ is the number of training samples, and $c$ is the number of classes. As Phillips described in [31], two sets are generated in difference space. One is the within-class differences set $S$, the other one is the between-class differences set $D$. In this way, we can formulate a multi-class classification problem into a classic binary one, which can be easily dealt with the traditional SVM. The within-class differences set $S$ and between-class differences set $D$ are formally described as

$$S = \{(\mathbf{x}_i - \mathbf{x}_j) | \mathbf{x}_i \sim \mathbf{x}_j, y_i = y_j, i, j = 1, \cdots, l\} \quad (1)$$

and

$$D = \{(\mathbf{x}_i - \mathbf{x}_j) \,|\mathbf{x}_i \not\sim \mathbf{x}_j, y_i \neq y_j, i, j = 1, \cdots, l\} \quad (2)$$

where $(\mathbf{x}_i - \mathbf{x}_j)$ is a paired-sample consisting of two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ in difference space. $\mathbf{x}_i \sim \mathbf{x}_j$ and $\mathbf{x}_i \not\sim \mathbf{x}_j$ indicate the two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same class (within-class) and in different class (between-class), respectively. Define that the labels of the paired-samples in the within-class differences set $S$ are $+1$, while the labels of the paired-samples in the between-class differences set $D$ are $-1$.

Obviously, the number of the generated training paired-samples is about $l^2$, which leads to an excessive number of training paired-samples. In addition, the number of the constructed dissimilar paired-samples is much larger than the number of the similar ones in multi-class problems. This is an unbalanced problem to be solved in similarity learning.

### B. Classic sampling methods in unbalanced data learning

Sampling methods play an important role in solving unbalanced problems. Generally speaking, these methods provide a balanced distribution by using different ways. In other word, these methods add some instances into the minority class or remove some examples from the majority class according to the required proportion of balance.

Random over-sampling and random under-sampling are the simplest sampling methods. Random over-sampling augments the original minority class by exactly copying several randomly selected examples, while random under-sampling removes some instances from the majority class randomly. At the first blush, random over-sampling and under-sampling methods seem to be functionally equivalent. Nevertheless, each of them introduces its own drawbacks. In the case of random under-sampling, the disadvantage is quite obvious: removing data means missing information, maybe very important to classifiers. Thus, it is probable to degrade the performance on the majority class. In the situation of random over-sampling, the problem is not visualized enough. Several copies of certain

examples may become "tied", making the decision region smaller and more specific, eventually leading to over-fitting problems [10].

Synthetic sampling with data generation is another significant part of sampling. Synthetic minority over-sampling technique (SMOTE) is a representative one in this community [12]. The SMOTE algorithm generates new synthetic instances along the line between the minority instance and its selected nearest neighbors. The detailed steps of SMOTE are illustrated as follows. For each instance $\mathbf{x}$ in the minority class, we first find its $K$ nearest neighbors in the same class. Then we randomly select some of the neighbors according to the amount of SMOTE. Next, for every selected neighbors $\mathbf{x}'$, we compute the difference $\mathbf{dif}$ between $\mathbf{x}$ and $\mathbf{x}'$, or $\mathbf{dif} = \mathbf{x}' - \mathbf{x}$, multiply $\mathbf{dif}$ by a random number $gap$ between 0 and 1. Finally, we get the new synthetic example $\mathbf{Synthetic}$. The formal description is

$$\mathbf{Synthetic} = \mathbf{x} + gap \times \mathbf{dif}$$

In summary, SMOTE has overcome the drawback of random over-sampling that may cause overfitting, and also made the decision region larger and more general.

Zhang et al. proposed four informed under-sampling methods based on $K$-Nearest Neighbor ($K$NN) classifier, namely, NearMiss-1, NearMiss-2, NearMiss-3 and the "most distant" method [14]. The NearMiss-1 method selects the majority class examples that are close to *some* of the minority examples. In this method, the majority examples would be selected when their average distances to three closest minority examples are smallest. NearMiss-2 selects majority examples that are close to *all* minority examples. That is, we select the majority examples whose average distances to three farthest minority examples are smallest. In NearMiss-3, for each example in minority class, we select a given number of the closest majority examples. This method guarantees every minority example is surrounded by some majority class. Finally, the "most distant" method selects the majority class examples whose average distance to the three closest minority class examples are the largest.

These classical methods could be directly used to solve the unbalanced problem in similarity learning. However, all paired-samples which are unbalanced must be obtained from the original dataset when applying these methods. In fact, it is time-consuming for generating all paired-samples. Even if we could generate all paired-samples, we still want to discard most of them by using these classical sampling methods. Thus, it costs a lot.

## III. SVM ENSEMBLE FOR UNBALANCED PROBLEM IN SIMILARITY LEARNING

In this section, we apply SVM ensemble to solve unbalanced problem in similarity learning. Typically, SVM ensemble learning consists of two sub-procedures. The first is how to generate individual SVMs. The second is how to combine the predictions into a final result. The proposed method will be described from these two parts.

### A. Generating individual SVMs

As described in Section II-A, traditionally, there exists an unbalanced problem in similarity learning, which has an influence up on the performance of standard SVM. To avoid this, the paired-samples must be balanced. Our strategy is that for each example $\mathbf{x}$ in the original training set $X$, just pick the same amount of the examples from the same class and the different class, not all the examples in $X$, to construct paired-samples. In ensemble learning, individual learners should be as diverse as possible. Thus, we randomly select the examples to construct paired-samples for each individual SVM. In this way, the individual SVMs can be definitely diverse. In the same time, it guarantees that each example in the original training set is with the same weight.

Here we display the sub-procedure of generating individual SVMs in formal description. Let $X = \{\mathbf{x}_i,\ y_i\}_{i=1}^{l}$ be the original training set. For each individual SVM, it is necessary to construct a new training set of paired-samples in difference space. Assume that there is $N$ individual SVMs totally, then $N$ new training sets $Z_1, Z_2, \cdots, Z_N$ are required. To construct $Z_n, n = 1, 2, \cdots, N$, we first randomly select some samples $\mathbf{x}_j$ in the original training set. For each of them, then we randomly select $k$ examples from the its class and $k$ examples from the different class, and respectively generate the within-class difference set $S_n$ and the between-class difference set $D_n$ by using them. Finally, $Z_n = S_n \cup D_n$.

After gaining $Z_n, n = 1, 2, \cdots, N$, use $N$ standard SVMs to train. We can get $N$ train models. The standard SVM is to solve the convex quadratic programming problem as follows:

$$\begin{aligned} \max_{\alpha} \ & \sum_{i=1}^{m} \alpha_i - \tfrac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j v_i v_j k\left(\mathbf{z}_i, \mathbf{z}_j\right) \\ s.t. \ & \sum_{i=1}^{m} \alpha_i v_i = 0 \\ & 0 \le \alpha_i \le C\,, \quad i = 1, 2, \cdots, m \end{aligned} \tag{3}$$

where $\mathbf{z}_i \in R^d$ is the paired-sample, $v_i \in \{-1, +1\}$ is the label for $\mathbf{z}_i$, $k\left(\mathbf{z}_i, \mathbf{z}_j\right)$ is the kernel function, $\alpha_i$ is the Lagrange multiplier, $m$ is the number of the paired-samples, and $C > 0$ is the regular factor.

In the test procedure of SVM, we input the paired-samples in test set into the training model obtained above. The discriminant function is described by

$$\hat{v} = f\left(\mathbf{z}\right) = \operatorname{sgn}\left(\sum_{i=1}^{m} \alpha_i v_i k\left(\mathbf{z}_i, \mathbf{z}\right) + b\right) \tag{4}$$

where $\operatorname{sgn}(\cdot)$ is the sign function, $\mathbf{z}$ is a test paired-sample, and $\hat{v}$ is the estimated label of $\mathbf{z}$. For any $0 < \alpha_j < C$, the threshold $b$ can be computed under KKT conditions by the following equation:

$$b = v_j - \sum_{i=1}^{n} \alpha_i v_i k\left(\mathbf{z}_i, \mathbf{z}\right) \tag{5}$$

### B. Combining multiple outputs

Traditionally, a test set for similarity learning is formed in this way: for a test example $\mathbf{x}$, construct the paired-samples with each examples in the original training set. Since the scale of test set is quite large when the original training set contains too much examples, it is a big challenge in computation complexity. To speed up the algorithm, it requires to cut down the scale of test set to some degree. We randomly

select $k'$ examples in each class from the original training set, and express the selected subset of the original training set as $X'$. For a test example $\mathbf{x}$, construct paired-samples with the examples in $X'$ instead of all the examples in the original training set. Until now, the test set $T$ is obtained.

Each individual SVM corresponds to a specific train model. For each paired-sample in $T$, there results in $N$ outputs obtained from all SVMs. In the following, we describe the combination rules for these outputs.

It is well known that the test paired-sample $\mathbf{z}$ is made up of two samples, the test sample $\mathbf{x}$ and any example $\mathbf{x}_i$ in $X$. Also, (4) only determines these two samples are similar or not, and cannot indicate the class attribute of the unseen samples $\mathbf{x}$. For the application of classification, it needs to utilize similarity to determine the class of test samples.

To determine the class label of an unseen sample $\mathbf{x}$, we construct the set of test paired-samples $T = \{\mathbf{z'}_i\}_{i=1}^{k' \times c}$, where $\mathbf{z'}_i = (\mathbf{x} - \mathbf{x}_i) \in R^d$ constructed with the original training samples $\mathbf{x}_i$ in $X'$. Note that the training samples in $X'$ are rearranged according to their labels, or the first $k'$ samples belongs to class 1 and so on. By (4), we can obtain $\hat{v}_{ni}, n = 1, \cdots, N, i = 1, \cdots, k' \times c$. Each $\hat{v}_{ni}$ represents whether the test sample $\mathbf{x}$ is similar to the training sample $\mathbf{x}_i$ according to the $n$th SVM. Of course, the labels of $\mathbf{x}_i$ are known since they are training samples.

To combine the outputs of multiple SVMs, we introduce some combination rules, including "voting then classifying" (or VC) rule, the "classifying then voting" (or CV) rule, the maximum (or MAX) rule, the minimum (or MIN) rule and the average (or AVG) rule. In the following, we describe these rules.

*1) VC rule:* For the test paired-sample $\mathbf{z}_i$, the VC rule first combines the $N$ similarities obtained from $N$ SVMs. Namely,

$$\hat{v}'_i = \operatorname{sgn}\left(\frac{\sum_{n=1}^{N}(\hat{v}_{ni}+1)}{2N} - \frac{1}{2}\right) \tag{6}$$

If $\hat{v}'_i = 1$, then we think that the test sample $\mathbf{x}$ is similar to the training sample $\mathbf{x}_i$. Otherwise, they are not similar when $\hat{v}'_i = -1$. Then we compute the similarity probability of the text sample in each class, which is defined as

$$P_j(\mathbf{x}) = \frac{\sum_{i=(j-1)\times k'+1}^{j\times k'}(\hat{v}'_i + 1)}{2k'}, j = 1, \cdots, c \tag{7}$$

We classify the test sample $\mathbf{x}$ according to the maximum similarity probability. Namely,

$$\hat{y} = \arg\max_{j=1,\cdots,c} P_j(\mathbf{x}) \tag{8}$$

*2) CV rule:* In the CV rule, the estimated classification labels on the test sample $\mathbf{x}$ are first obtained, and then these classification results are combined by some way. The similarity probability in each class generated by the $n$th classifier is expressed as:

$$P_{nj}(\mathbf{x}) = \frac{\sum_{i=(j-1)\times k'+1}^{j\times k'}(\hat{v}_{ni} + 1)}{2k'}, j = 1, \cdots, c, n = 1, \cdots, N \tag{9}$$

By using these probabilities, we give the classification results

$$\hat{y}_n = \arg\max_{j=1,\cdots,c} P_{nj}(\mathbf{x}) \tag{10}$$

where $\hat{y}_n$ is the estimated classification label on $\mathbf{x}$ obtained from the $n$th classifier. According to the majority voting rule, these classification labels $\hat{y}_n, n = 1, \cdots, N$ determine the final estimate label for $\mathbf{x}$.

*3) MAX rule:* Given the similarity probability $P_{nj}, n = 1, \cdots, N, j = 1, \cdots, c$, the MAX rule is to estimate the label for $\mathbf{x}$ by

$$\hat{y} = \arg\max_{j=1,\cdots,c} \max_{n=1,\cdots,N} P_{nj} \tag{11}$$

*4) MIN rule:* Given the similarity probability $P_{nj}, n = 1, \cdots, N, j = 1, \cdots, c$, the MIN rule is to estimate the label for $\mathbf{x}$ by

$$\hat{y} = \arg\max_{j=1,\cdots,c} \min_{n=1,\cdots,N} P_{nj} \tag{12}$$

*5) AVG rule:* The AVG Rule is parallel to the MAX rule and MIN rule. The only difference is that the final similarity probability of each class is the average value of the results in individual SVMs, which can be expressed by

$$\hat{y} = \arg\max_{j=1,\cdots,c} \frac{\sum_{n=1}^{N} P_{nj}}{N} \tag{13}$$

## IV. Experiments

In order to validate the effectiveness of the proposed method for solving unbalanced problems, we select some popular datasets, including Iris dataset from UCI [32], the MNIST database of handwritten digits [33] and the UMIST Face Database [34]. The compared methods are four representative under-sampling methods, which are random under-sampling, NearMiss-1, NearMiss-2, and NearMiss-3 method.

Some experiment settings are explained here, such as the data pre-processing and some parameters. First, a data normalization processing is needed. All the selected datasets are mapped into the interval $[0, 1]$. Second, the kernel function of SVM is the popular radial basis function (RBF), which is denoted by $k(\mathbf{z}, \mathbf{z}') = \exp\left\{-\gamma\|\mathbf{z} - \mathbf{z}'\|^2\right\}$, where $\gamma > 0$ is the kernel parameter. The value of $\gamma$ is determined according to the training examples [36]. There is another parameter related to SVM, which is called the regular factor $C$. In our experiment, $C$ is determined via 5 fold cross-validation, ranging from $\{10^{-3}, 10^{-2}, \cdots, 10^3\}$. The results of cross-validation show that the performance is best when $C$ is equal to 10. Hence, we set $C = 10$ in the following experiments. The four sampling methods sample the data to a completed balanced data distribution.

### A. Iris dataset

The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the empirical analysis of machine learning algorithms. Iris dataset is one of the most popular datasets.

The Iris dataset contains 3 classes. Each class describes a type of iris plants. Note that one class is linearly separable

from the other two, while the latter are not linearly separable from each other. 50 instances are included in each class, 150 in total. Each sample has 4 attributes. In the experiment, half of the samples are selected randomly as training samples in each class, while the rest as test samples. We perform 10 times and report the average result.

To construct the training set of paired-samples for SVMs, we first randomly select $k$ samples in each class from the training set. For each selected sample, we randomly select $k$ other examples in the same class and $k$ examples in the different class. In the case of constructing test paired-samples, we randomly select $k'$ examples from each class for a test sample. Here, let $k = k' = 10$.

We compare the performance of five combination rules when varying the number of individual SVMs $N$. The number of individual SVMs $N$ takes value from the set $\{5, 10, 15, 20, 30, 40, 50, 100\}$. Figure 1 and Table I show test errors versus (vs.) different $N$ for our method, and Table II gives test errors obtained from other four methods. The performance index "Number" in these tables means the number of paired-samples in the training set in SVM ensemble. While in the four under-sampling methods, it means the number of paired-samples after/without under-sampling.

The running time in Table I includes the time of constructing training and test paired-sample sets, the time of training all SVMs and the time of test all samples in the test set. Note that the running time in Table I includes the time of training $N$ SVMs, while the running time in Table II just contains that of one SVM.

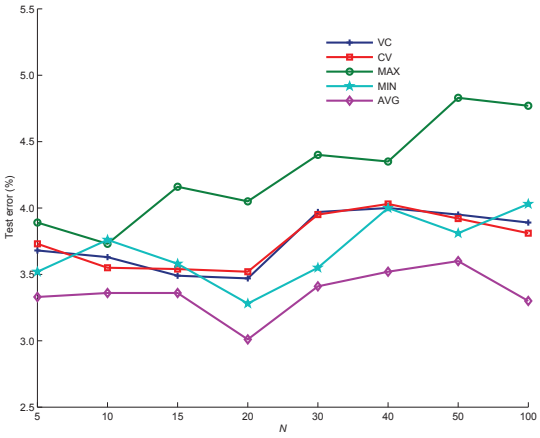



Fig. 1. Test errors versus different $N$ on the Iris dataset

From Figure 1, it is obvious that the AVG rule is the best one with highest accuracy and strongest stability among all the five rules. From Table I, we can see that the number of SVMs has a little effect on the classification performance, but has a great effect on the running time. Table II indicates that the accuracy of the four under-sampling methods is quite high, which is compared to our methods. However, the time spent on the sampling procedure cannot be ignored, especially in NearMiss-1, NearMiss-2 and NearMiss-3. When the scale of dataset is much larger, the re-sampling time is insufferable, which will be reflected in the following experiments. When $N$ is small enough, the running time is competitive to the other methods.

In the proposed method, the accuracy is pretty good among all the five rules when $N$ is 20. Considering both accuracy and running time, we choose the AVG rule as the combination rule of individual SVMs, and set $N$ to 15 in the following experiments.

### B. MNIST database of handwritten digits

The MNIST database of handwritten digits is a popular database in machine learning and pattern recognition. The database has a training set of 60,000 instances and a test set of 10,000 instances, which is a subset of a larger set available from NIST. It contains 10 classes, namely class 0-9. Each instance is a gray level image with the size of 28×28 pixel. Figure 2 shows some samples from the MNIST database. In this experiment, we just select five classes, class 1, 3, 7, 8, and 9. Considering the huge expense on sampling procedure in the under-sampling methods, 50 training examples and 100 test examples are picked from each class randomly. That is, there are 250 training examples and 500 test examples in total. As the dimension is quite high, dimensionality reduction is in badly need. Influenced by the recent upsurge of compressed sensing, random projection is becoming an effective and fast dimensionality reduction method [35], [36]. Here, random projection is used on the original samples. Namely,

$$\bar{\mathbf{x}}_i = \mathbf{R}\mathbf{x}_i \tag{14}$$

where $\bar{\mathbf{x}}_i \in R^{d'}$ is the corresponding sample of $\mathbf{x}_i$ in a projected subspace, $\mathbf{R} \in R^{d' \times d}$ is a random projection matrix, and $d'$ is the dimensionality of the projected subspace. $d'$ is set to 50 in this experiment. Similarly, we perform 10 trials and report an average result.

Fig. 2. Some samples in the MNIST database of handwritten digits

In the experiment of the MNIST database, we set $k = 10$ and $k' = 10$. As described in Section IV-A, we select the AVG rule as the combination strategy, and set $N$ to 15. The results are illustrated in Table III.

From Table III, it is easy to find out that the proposed method not only has a higher accuracy but also a much faster runtime. The other four methods consume too much time on sampling. Without considering the runtime of the four under-sampling methods, more examples can be included in the training set, which will improve the performance of our method.

### C. UMIST face database

The UMIST face database consists of 564 images of 20 people. Individuals cover a range of race/sex/appearance. Each

individual is shown in a range of poses from profile to frontal views. Figure 3 shows some samples in the UMIST face database. Each sample is an image of size 112×92. The dimensionality is quite high, so random projection is still used here. The dimensionality of projection space is also 50. The experiments is repeated 10 times. Since the number of the instances for each person is not the same and the number of individuals is relatively large, we select less examples to construct paired-samples, $k = 5$ and $k' = 5$. The results are shown in Table IV.

Fig. 3. Some samples in the UMIST face database

From Table IV, it is obvious that the prediction of the proposed method is best. Meanwhile, random under-sampling method has done a better job in running time. The UMIST database contains images of 20 people, thus the number of paired-samples in training set after random under-sampling is not very large. While in SVM ensemble, the SVM training procedure runs 15 times. The NearMiss-1, NearMiss-2 and NearMiss-3 method get a lower accuracy while pay a huge cost on cpu time, especially in NearMiss-2.

## V. Conclusion

This paper deals with similarity learning using SVM ensemble which has a great advantage in solving the unbalanced problems. This problem is caused by constructing paired-samples in traditional way. From the results on the Iris dataset, the proposed method gets a competitive accuracy compared to the other under-sampling methods while the advantage of speed is not outstanding. The reason is that the Iris dataset is in small size. Meanwhile, the AVG rule gets the best performance when varying $N$. On the MNIST database and UMIST face database, the proposed method outperforms the other four methods in both prediction and running time. To sum up, the proposed method is meaningful, especially when the scale of dataset is relatively larger.

## Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61373093, 61033013, and 61271301, by the Natural Science Foundation of Jiangsu Province of China under Grant Nos. BK2011284 and BK201222725, by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant No.13KJA520001 and by the Qing Lan Project.

## References

[1] T. Cover, P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol.13, no. 1, pp. 21-27, 1967.

[2] C.J.C. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp, 121-167, 1998.

[3] N. Cristianini and J. Shawe-Taylor, "Support Vector Machine," *Cambridge University Press*, 2000.

[4] L. Zhang, "Research on Support Vector Machines and Kernel Methods," Ph.D. thesis, Xidian University, 2009.

[5] G. M. Weiss, F. Provost, "The Effect of Class Distribution on Classifier Learning: An Empirical Study," *Technical Report ML-TR-43*, Dept. of Computer Science, Rutgers Univ. ,2001.

[6] J. Laurikkala, "Improving Identification of Difficult Small Classes by Balancing Class Distribution," *Proc. Conf. AI in Medicine in Europe: Artificial Intelligence Medicine*, pp. 63-66, 2001.

[7] A. Estabrooks, T. Jo, N. Japkowicz, "A Multiple Resampling Method for Learning from Imbalanced datasets," *Computational Intelligence*, vol. 20, pp. 18-36, 2004.

[8] N. V. Chawla, N. Japkowicz, "A. Kolcz, Editorial: Special Issue on Learning from Imbalanced datasets," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1-6, 2004.

[9] G. Weiss, "Mining with rarity: A unifying framework," *ACM SIGKDD Explorations Newsletter*, vol. 6 no. 1, pp. 7-19, 2004.

[10] D. Mease, A. J. Wyner, "A. Buja, Boosted classification trees and class probability/quantile estimation," *The Journal of Machine Learning Research*, vol. 8, pp. 409-439, 2007.

[11] M. Kubat, S. Matwin, "Addressing the Course of Imbalanced Training Sets: One-sided Selection," *International Conference on Machine Learning*, pp. 179-186, 1997.

[12] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002.

[13] H. Han, W.Y. Wang, B.H. Mao, "Borderline-SMOTE: A New Over-Sampling Method in Imbalanced datasets Learning," *Proc. Int'l Conf. Intelligent Computing*, pp. 878-887, 2005.

TABLE I. Test errors versus different $N$ on the Iris dataset

| $N$ | Number | Test error(%) | | | | | Running time(s) |
|---|---|---|---|---|---|---|---|
| | | VC | CV | MAX | MIN | AVG | |
| 5 | 1500 | 3.68 | 3.73 | 3.89 | 3.52 | 3.33 | 1.06 |
| 10 | 1500 | 3.63 | 3.55 | 3.73 | 3.76 | 3.36 | 2.17 |
| 15 | 1500 | 3.49 | 3.54 | 4.16 | 3.58 | 3.36 | 3.12 |
| 20 | 1500 | 3.47 | 3.52 | 4.05 | 3.28 | **3.01** | 4.10 |
| 30 | 1500 | 3.97 | 3.95 | 4.40 | 3.55 | 3.41 | 6.36 |
| 40 | 1500 | 4.00 | 4.03 | 4.35 | 4.00 | 3.52 | 8.63 |
| 50 | 1500 | 3.95 | 3.92 | 4.83 | 3.81 | 3.60 | 10.67 |
| 100 | 1500 | 3.89 | 3.81 | 4.77 | 4.03 | 3.30 | 21.86 |

TABLE II. Test errors obtained from four under-sampling methods on the Iris dataset

| Methods | Number | Test error(%) | Running time(s) |
|---|---|---|---|
| Random under-sampling | 3750/5625 | 3.10 | 0.65 |
| NearMiss-1 | 3750/5625 | 3.34 | 2.80 |
| NearMiss-2 | 3750/5625 | 2.89 | 3.00 |
| NearMiss-3 | 3750/5625 | 2.32 | 4.30 |

[14] J. Zhang, I. Mani, "KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction," *Proceedings of Workshop on Learning from Imbalanced Datasets*, 2003.

[15] N. V. Chawla, A. Lazarevic, L. O. Hall, K. W. Bowyer, "SMOTEBoost: Improving Prediction of the Minority Class in Boosting," *Knowledge Discovery in Databases: PKDD 2003*, Springer, Berlin Heidelberg, pp. 107-119, 2003.

[16] H. Guo, H.L. Viktor, "Learning from Imbalanced datasets with Boosting and Data Generation: The DataBoost IM Ap-proach," *ACM SIGKDD Explorations Newsletter*, vol. 6. no. 1, pp. 30-39, 2004.

[17] H. Guo, H.L. Viktor, "Boosting with Data Generation: Improving the Classification of Hard to Learn Examples," *Proc. Int'l Conf. Innovations Applied Artificial Intelligence*, pp. 1082-1091, 2004.

[18] Y. Tang and Y. Q. Zhang, "Granular SVM with Repetitive Undersampling for Highly Imbalanced Protein Homology Prediction," *Proc. Int'l Conf. Granular Computing*, pp. 457-460, 2006.

[19] M. Joshi , V. Kumar, R. Agarwal, "Evaluating Boosting Algorithms to Classify Rare Classes: Comparison and Improvements," *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on. IEEE*, pp. 257-264, 2001.

[20] G. Wu, Y. C. Edward, "Class-Boundary Alignment for Imbalanced Dataset Learning,", *ICML 2003 workshop on learning from imbalanced datasets II*, Washington, DC, pp. 49-56, 2003.

[21] B. Raskutti, A. Kowalczyk, "Extreme Re-Balancing for SVMs: A Case Study," *ACM SIGKDD Explorations Newsletter*, vol. 6. no. 1, pp. 60-69, 2004.

[22] B. Scholkopt, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation*, vol. 13, pp. 1443-1471, 2001.

[23] L. M. Manevitz, M. Yousef, "One-Class SVMs for Document Classification," *Machine Learning Research*, vol. 2, pp. 139-154, 2001.

[24] L. Zhuang, H. Dai, "Parameter Estimation of One-Class SVM on Imbalance Text Classification,", *Lecture Notes in Artificial Intelligence*, vol. 4013, pp. 538-549, 2006.

[25] H. J. Lee, S. Cho, "The Novelty Detection Approach for Difference Degrees of Class Imbalance," *ACM SIGKDD Explorations Newsletter*, vol. 4233, pp. 21-30, 2006.

[26] L. Zhuang, H. Dai, "Parameter Optimization of Kernel-Based One-Class Classifier on Imbalance Text Learning," *ACM SIGKDD Explorations Newsletter*, vol. 4099, pp. 434-443, 2006.

[27] N. Japkowicz, "Supervised versus Unsupervised Binary-Learning by Feedforward Neural Networks," *Machine Learning*, vol. 42, pp. 97-122, 2001.

[28] L. Manevitz, M. Yousef, "One-Class Document Classification via Neural Networks," *Neurocomputing*, vol. 70, pp.1466-1481, 2007.

[29] N. Japkowicz, "Learning from Imbalanced datasets: A Comparison of Various Strategies," *AAI workshop on learning from imbalanced data setsq*, pp. 10-15, 2000.

[30] N. Japkowicz, C. Myers, M. Gluck, "A Novelty Detection Approach to Classification," *Proc. Joint Conf. Artificial Intelligence*, pp. 518-523, 1995.

[31] P. J. Phillips, "Support vector machines applied to face recognition," *Advances in Neural Information Processing Systems 11*, 1998.

[32] P. M. Murphy, D. W. Aha, "UCI machine learning repository," *http://www.ics.uci.edu/ mlearn/MLRepository.html*, 1992.

[33] C. L. Liu, K. Nakashima, H. Sako, et al, "Handwritten digit recognition using state-of-the-art techniques, Frontiers in Handwriting Recognition, 2002", it Proceedings. Eighth International Workshop on. IEEE, pp. 320-325, 2002.

[34] "UMIST face database (n.d.)." *Retrieved June 6, 2003 from http://images.ee.umist.ac.uk/danny/database.html*

[35] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 210-227 , 2009.s

[36] L. Zhang, W. D. Zhou, P.-C. Chang, J. Liu, Z. Yan, T. Wang, F.-Z. Li, "Kernel Sparse Representation-Based Classifier," *IEEE Transactions on Signal Processing*, vol. 60, pp. 1684-1695, 2012.

TABLE III. TEST RESULTS ON THE MNIST DATABASE

| Methods | Number | Test error(%) | Running time(s) |
|---|---|---|---|
| SVM ensemble | 5000 | 13.58 | 481.2 |
| Random under-sampling | 25000/62500 | 18.10 | 1014.8 |
| NearMiss-1 | 25000/62500 | 36.16 | 5971.1 |
| NearMiss-2 | 25000/62500 | 39.24 | 7885.1 |
| NearMiss-3 | 25000/62500 | 20.98 | 11457.0 |

TABLE IV. TEST ERRORS ON THE UMIST DATABASE

| Methods | Number | Test error(%) | Running time(s) |
|---|---|---|---|
| SVM ensemble | 2820 | 5.11 | 292.7 |
| Random under-sampling | 8904/79524 | 7.50 | 72.5 |
| NearMiss-1 | 8904/79524 | 6.23 | 1200.6 |
| NearMiss-2 | 8904/79524 | 15.32 | 1645.8 |
| NearMiss-3 | 8904/79524 | 6.83 | 1576.1 |

# An Iterative Link-based Method for Parallel Web Page Mining

**Le Liu[1], Yu Hong[1], Jun Lu[2], Jun Lang[2], Heng Ji[3], Jianmin Yao[1]**
[1]School of Computer Science & Technology, Soochow University, Suzhou, 215006, China
[2]Institute for Infocomm Research, Singapore, 138632
[3]Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180, USA
giden@sina.cn,{tianxianer,lujun59,billlangjun}@gmail.com
jih@rpi.edu,jyao@suda.edu.cn

## Abstracts

Identifying parallel web pages from bilingual web sites is a crucial step of bilingual resource construction for cross-lingual information processing. In this paper, we propose a link-based approach to distinguish parallel web pages from bilingual web sites. Compared with the existing methods, which only employ the internal translation similarity (such as content-based similarity and page structural similarity), we hypothesize that the external translation similarity is an effective feature to identify parallel web pages. Within a bilingual web site, web pages are interconnected by hyperlinks. The basic idea of our method is that the translation similarity of two pages can be inferred from their neighbor pages, which can be adopted as an important source of external similarity. Thus, the translation similarity of page pairs will influence each other. An iterative algorithm is developed to estimate the external translation similarity and the final translation similarity. Both internal and external similarity measures are combined in the iterative algorithm. Experiments on six bilingual websites demonstrate that our method is effective and obtains significant improvement (6.2% F-Score) over the baseline which only utilizes internal translation similarity.

## 1 Introduction

Parallel corpora have played an important role in multilingual Natural Language Processing, especially in Machine Translation (MT) and Cross-lingual Information Retrieval(CLIR). However, it's time-consuming to build parallel corpora manually. Some existing parallel corpora are subject to subscription or license fee and thus not freely available, while others are domain-specific. Therefore, a lot of previous research has focused on automatically mining parallel corpora from the web.

In the past decade, there have been extensive studies on parallel resource extraction from the web (e.g., Chen and Nie, 2000; Resnik 2003; Jiang et al., 2009) and many effective Web mining systems have been developed such as STRAND, PTMiner, BITS and WPDE. For most of these mining systems, there is a typical parallel resource mining strategy which involves three steps: (1) locate the bilingual websites (2) identify parallel web pages from these bilingual websites and (3) extract bilingual resources from the parallel web pages.

In this paper, we focus on the step (2) which is regarded as the core of the mining system (Chunyu, 2007). Estimating the translation similarity of two pages is the most basic and key problem in this step. Previous approaches have tried to tackle this problem by using the information within the pages. For example, in the STRAND and PTMiner system, a structural filtering process that relies on the analysis of the underlying HTML structure of pages is used to determine a set of pair-specific structural values, and then the values are used to decide whether the pages are translations of one another. The BITS system filters out bad pairs by using a large bilingual dictionary to compute a content-based similarity score and comparing the score with a threshold. The WPDE system combines URL similarity, structure similarity with content-based similarity to discover and verify candidate parallel page pairs. Some other features or rules such as page size ratio, predefined hypertexts which link to different language versions of a web page are also used in most of these systems. Here, all of the mining systems are simply using the information within the page in the process of find-



*Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1216–1224,
October 25-29, 2014, Doha, Qatar. 

ing parallel web pages. In this paper, we attempt to explore other information to identify parallel web pages.

On the Internet, most web pages are linked by hyperlinks. We argue that the translation similarity of two pages depends on not only their internal information but also their neighbors. The neighbors of a web page are a set of pages, which link to the page. We find that the similarity of neighbors can provide more reliable evidence in estimating the translation similarity of two pages.

The main issues are discussed in this paper as follows:

- *Can the neighbors of candidate page pairs really contribute to estimating the translation similarity?*
- *How to estimate the translation similarity of candidate page pairs by using their neighbors?*

Our method has the following advantages:

**High performance**

The external and internal information is combined to verify parallel page pairs in our method, while in previous mining systems, only internal information was used. Experimental results show that compared with existing parallel page pair identification technologies, our method obtains both higher precision and recall (6.2% and 6.3% improvement than the baseline, respectively). In addition, the external information used in our method is a more effective feature than internal features alone such as structural similarity and content-based similarity.

**Language independent**

In principle, our method is language independent and can be easily ported to new language pairs, except for the language-specific bilingual lexicons. Our method takes full advantage of the link information that is language-independent. For the bilingual lexicons in our experiments, compared to previous methods, our method does not need a big bilingual lexicon, which is good news to less-resource language pairs.

**Unsupervised and fewer parameters**

In previous work, some parameters need to be optimized. Due to the diversity of web page styles, it is not trivial to obtain the best parameters. Some previous researches(Resnik, 2003; Zhang et al., 2006) attempt to optimize parameters by employing machine learning method. In contrast, in our method, only two parameters need to be estimated. One parameter remains stable for different style websites. Another parameter can be easily adjusted to achieve the best performance. Therefore, our method can be used in other websites with different styles, without much effort to optimize these parameters.

## 2 Related Work

A large amount of literature has been published on parallel resource mining from the web. According to the existing form of the parallel resource on the Internet, related work can be categorized as follows:

**Mining from bilingual websites**

Most existing web mining systems aimed at mining bilingual resource from the bilingual websites, such as PTMiner (Nie et al., 1999), STRAND (Resnik and Smith, 2003), BITS (Ma and Liberman, 1999), PTI (Chen et al., 2004). PTMiner uses search engines to pinpoint the candidate sites that are likely to contain parallel pages, and then uses the collected URLs as seeds to further crawl each web site for more URLs. Web page pairs are extracted based on manually defined URL pattern matching, and further filtered according to several criteria. STRAND uses a search engine to search for multilingual websites and generated candidate page pairs based on manually created substitution rules. Then, it filters some candidate pairs by analyzing the HTML pages. PTI crawls the web to fetch (potentially parallel) candidate multilingual web documents by using a web spider. To determine the parallelism between potential document pairs, a filename comparison module is used to check filename resemblance, and a content analysis module is used to measure the semantic similarity. BITS was the first to obtain bilingual websites by employing a language identification module, and then for each bilingual website, it extracts parallel pages based on their content.

**Mining from bilingual web pages**

Parallel/bilingual resources may exist not only in two parallel monolingual web pages, but also in single bilingual web pages. Jiang et al. (2009) used an adaptive pattern-based method to mine interesting bilingual data based on the observation that bilingual data usually appears collectively following similar patterns. They found that bilingual web pages are a promising source of up-to-date bilingual terms/sentences which cover many domains and application scenarios. In addition, Feng et al. (2010) proposed a new method

to automatically acquire bilingual web pages from the result pages of a search engine.

**Mining from comparable corpus**

Several attempts have been made to extract parallel resources from comparable corpora. Zhao et al. (2002) proposed a robust, adaptive approach for mining parallel sentences from a bilingual comparable news collection. In their method, sentence length models and lexicon-based models were combined under a maximum likelihood criterion. Smith et al. (2010) found that Wikipedia contains a lot of comparable documents, and adopted a ranking model to select parallel sentence pairs from comparable documents. Bharadwaj et al. (2011) used a SVM classifier with some new features to identify parallel sentences from Wikipedia.

## 3 Iterative Link-based Parallel Web Pages Mining

As mentioned, the basic idea of our method is that the similarity of two pages can be inferred from their neighbors. This idea is illustrated in Figure 1.

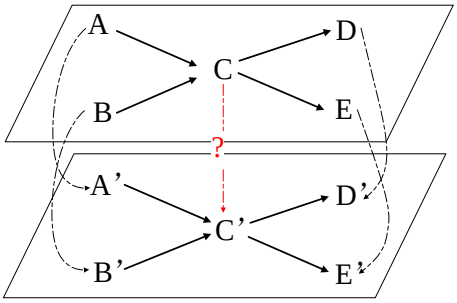



Figure 1 Illustration of the link-based method

In Figure 1, *A*, *B*, *C*, *D* and *E* are some pages in the same language; while *A'*, *B'*, *C'*, *D'* and *E'* are some pages in another language. The solid black arrows indicate the links between these pages. For example, page *A* points to *C*, page *B'* points to *C'* and so on. Then the page set {*A*, *B*, *D*, *E*} is called the neighbors of page *C*. Similarly, the page set {*A'*, *B'*, *D'*, *E'*} contains the neighbors of page *C'*. If the page pairs : <*A*, *A'*>, <*B*, *B'*>, <*D*, *D'*> and <*E*, *E'*> have high translation similarities, then it can be inferred that page *C* and *C'* have a high probability to be a pair of parallel pages. Every page has its own neighbors. For each web page, our method views link-in and link-out hyperlinks as the same. Thus, the linked pages will influence each other in estimating the translation similarity. For example, the similarities of two pairs <*A*, *A'*> and <*C*, *C'*> will influence each other. It is an iterative process. We will elaborate the process in the following sections.

Since our goal is to find parallel pages in a specific website, the key task is to evaluate the translation similarity of two pages (which are in different languages) as accurately as possible. The final similarity of two pages should depend both on their internal similarity and external similarity. The internal similarity means the similarity estimated by using the information in the page itself, such as the structure similarity and the content-based similarity of the two pages. On the other hand, the external similarity of two pages is the similarity depending on their neighbors. The final translation similarity is called the **Enhanced Translation Similarity** (*ETS*). The *ETS* of two pages can be calculated as follows:

$$ETS(e,c) = \alpha \cdot S_{ext}(e,c) + (1-\alpha) \cdot S_{in}(e,c), \alpha \in [0,1] \quad (1)$$

Where, $S_{in}(e,c)$ is the internal translation similarity of two pages: $e$ and $c$; $S_{ext}(e,c)$ represents the external translation similarity of pages $e$ and $c$. $ETS(e,c)$ indicates the final similarity of two pages, which combines the internal with external translation similarity.

In this paper, we conduct the experiments on English-Chinese parallel page pair mining. However, our method is language-independent. Thus, it can be applied to other language pairs by only replacing a bilingual lexicon. The symbol $e$ and $c$ always indicate an English page and a Chinese page respectively in this paper. In the following sections, we will describe how to calculate the $S_{in}(e,c)$ and $S_{ext}(e,c)$ step by step.

### 3.1 Preprocessing

The input of our method is a bilingual website. This paper aims to find English/Chinese parallel pages. So a 3-gram language model is used to identify (or classify) the language of a certain document. The performance of the language identification module achieves 99.5% accuracy through in-house testing. As a result, a set of English pages and a set of Chinese pages are obtained. In order to get the neighbors of a page, for each bilingual website, two networks are constructed based on the hyperlinks, one for English pages and another for Chinese pages.

### 3.2 The Internal Translation Similarity

Following Resnik and Smith (2003), three features are used to evaluate the internal translation similarity of two pages:

**The size ratio of two pages**

The length ratio of two documents is the simplest criterion for determining whether two documents are parallel or not. Parallel documents tend to be similar in length. And it is reasonable to assume that for text *E* in one language and text *F* in another language, length(*E*) ≈ *C*•length(*F*), where *C* is a constant that depends on the language pair. Here, the content length of a web page is regarded as its length.

**The structure similarity of two pages**

The HTML tags describe and control a web page's structure. Therefore, the structure similarity of two pages can be calculated by their HTML tags. Here, the HTML tags of each page are extracted (except the visual tags such as "B", "FONT".) as a linear sequence. Then the structure similarity of two pages is computed by comparing their linearized sequences. In this paper, the LCS algorithm (Dan, 1997) is adopted to find the longest common sequences of the two HTML tag sequences. The ratio of LCS length and the average length of two HTML tag sequences are used as the structure similarity of the two pages.

**The content-based translation similarity of two pages**

The basic idea is that if two documents are parallel, they will contain word pairs that are mutual translations (Ma, 1999). So the percentage of translation word pairs in the two pages can be considered as the content-based similarity. The translation words of two documents can be extracted by using a bilingual lexicon. Here, for each word in English document, we will try to find a corresponding word in Chinese document.

Finally, the internal translation similarity of two pages is calculated as follows:

$$S_{in}(e,c) = \beta \cdot S_{cb}(e,c) + (1-\beta) \cdot S_{struct}(e,c), \beta \in [0,1] \quad (2)$$

Where, $S_{cb}(e,c)$ and $S_{struct}(e,c)$ are the content-based and structural similarity of page e and c respectively. In addition, the size ratio of two pages is used to filter invalid page pairs.

### 3.3 The External and Enhanced Translation Similarity

As described above, the external translation similarity of two pages depends on their neighbors:

$$S_{ext}(e,c) = Sim\big(PG(e), PG(c)\big) \quad (3)$$

Where, *PG(x)*, a set of pages, is the neighbors of page *x*. Obviously, the similarity of two sets relies on the similarity of the elements in the two sets. Here, the elements are namely web pages. So, $S_{ext}(e,c)$ equals to $Sim\big(PG(e), PG(c)\big)$, and $Sim\big(PG(e), PG(c)\big)$ depends on $ETS\big(e_i, c_j\big)$ ($e_i, c_j$ belongs to $PG(e)$, $PG(c)$, respectively) and $ETS(e,c)$. According to Equation (1), $ETS(e,c)$ depends on $S_{in}(e,c)$ and $S_{ext}(e,c)$. Therefore, it is a process of iteration. $ETS(e,c)$ will converge after a certain number of iterations. Thus, $ETS^i(e,c)$ is defined as the enhanced similarity of page *e* and *c* after the *i*-th iteration, and the same is for $S^i_{ext}(e,c)$ and $Sim^i\big(PG(e), PG(c)\big)$. $Sim^i\big(PG(e), PG(c)\big)$ is computed by the following algorithm:

**Algorithm 1: Estimating the external translation similarity**

**Input**: $PG(e), PG(c)$
**Output**: $S^i_{ext}(e,c)$

**Procedure:**

$sum \leftarrow 0$
$e\_set \leftarrow PG(e)$
$c\_set \leftarrow PG(c)$
While $e\_set$ and $c\_set$ are both not empty:
  $< x, y >$
  $\leftarrow arg\,max_{x \in e\_set,\ y \in c\_set}\ (ETS^{i-1}(x,y))$
  $sum \leftarrow sum +$ $\mathrm{ETS}^{i-1}(x,y)$
  Remove $x$ from $e\_set$
  Remove y from $c\_set$
$S^i_{ext}(e,c) = Sim^i\big(p(e), p(c)\big)$
$= 2 \cdot sum \,/\, (|PG(e)| + |PG(c)|)$

**Algorithm 2 Estimating the enhanced translation similarity**

**Input**: $P_e, P_c$, (*the English and Chinese page set*)
**Output**: ETS(e, c), $e \in P_e, c \in P_c$
**Initialization:** Set *ETS*(*e*, *c*) random value or small value

**Procedure:**

LOOP:
  For each $e$ in $P_e$ :
    For each $c$ in $P_c$:
      $ETS^i(e,c) = \alpha \cdot S^i_{ext}(e,c) + (1-\alpha) \cdot S_{in}(e,c)$
  Parameters normalization
UNTIL $ETS(e,c)$ is stable

Algorithm 1 tries to find the real parallel pairs from $PG(e)$ and $PG(c)$. The similarity of $PG(e)$ and $PG(c)$ is calculated based on the similarity

values of these pairs. Finally, $ETS(e,c)$ is calculated by the following algorithm 2.

In Algorithm 2, the input $P_e$ and $P_c$ are English and Chinese page sets in a certain bilingual website. We use algorithm 2 to estimate the enhanced translation similarity.

### 3.4 Find the Parallel Page Pairs

At last, the enhanced translation similarity of every pair is obtained, and the parallel page pairs can be extracted in terms of these similarities:

**Algorithm 3 Finding parallel page pairs**

**Input**: $P_e, P_c$
$ETS(x,y),\ \ x \in P_e, y \in P_c$
$MAX\_P$ (or $MIN\_SIM$)

**Output**: Parallel Page Pairs List : $PPL$

**Procedure:**

LOOP:
$< x,y > = \ arg\,max_{x \in P_e, y \in P_c}\,(ETS(x,y))$
Add $< x,y >$ to $PPL$
Remove $x$ from $P_e$
Remove $y$ from $P_c$
UNTIL size of $PPL$ > $MAX\_P$ (or $ETS(x,y)$ < $MIN\_SIM$)

This algorithm is similar to Algorithm 1 in each bilingual website. The input $MAX\_P$ is an integer threshold which means that only top $MAX\_P$ page pairs will be extracted in a certain website. It needs to be noted that $MAX\_P$ is always less than $|P_e|$ and $|P_c|$. While the input $MIN\_SIM$ is another kind of threshold that is used for extracting page pairs with high translation similarity.

## 4 Experiments and Analysis

### 4.1 Experimental setup

Our experiments focus on six bilingual websites. Most of them are selected from HK government websites. All the web pages were retrieved by using a web site download tool: HTTrack[1]. We notice that a small amount of pages doesn't always contain valuable contents. So, we put a threshold (100 bytes in our experiment) on the web pages' content to filter meaningless pages. In order to evaluate our method, the bilingual page pairs of each website are annotated by a human annotator. Finally, we got 23109 pages and 11684 bilingual page pairs in total for testing. The basic information of these websites is listed in Table 1.

It's time-consuming to annotate whether two pages is parallel or not. Note that if a website contains *N* English pages and *M* Chinese pages, an annotator has to label *N*M* page pairs. To the best of our knowledge, there is no large scale and public parallel page pair dataset with human annotation. So we try to build a reliable and large-scale dataset.

In our experiments, *URL* similarity is used to reduce the workload for annotation. For a certain website, firstly, we obtain its *URL* pattern between English and Chinese pages manually. For example, in the website "www.gov.hk", the *URL* pairs like:

*http://www.gov.hk/en/about/govdirectory/ (English)*

*http://www.gov.hk/sc/about/govdirectory/ (Chinese)*

The *URL* pairs always point to a pair of parallel pages. So <"/en/","/sc/"> is considered as a *URL* pattern that was used to find parallel pages. For the other *URLs* that can't match the pattern, we have to label them by hand. The column "No pattern pairs" in Table 1 shows that the number of parallel page pairs which mismatch any patterns.

Table 1 Number of pages and bilingual page pairs of each websites

| Site ID | En/Ch pages | Total pairs | No pattern pairs | URL |
|---|---|---|---|---|
| S1 | 1101/1098 | 1092 | 20 | www.gov.hk |
| S2 | 501/497 | 487 | 7 | www.customs.gov.hk |
| S3 | 995/775 | 768 | 12 | www.sbc.edu.sg |
| S4 | 4085/3838 | 3648 | 4 | www.swd.gov.hk |
| S5 | 660/637 | 637 | 0 | www.landsd.gov.hk |
| S6 | 4733/4626 | 4615 | 8 | www.td.gov.hk |
| total | 12075/11471 | 11684 | 51 | |

Each website listed in Table 1 has a *URL* pattern for most parallel web pages. Some previous researches used the *URL* similarity or patterns to find parallel page pairs. However, due to the diversity of web page styles and website maintenance mechanisms, bilingual websites adopt varied naming schemes for parallel documents (Shi, et al, 2006). The effect of *URL* pattern-based mining always depends on the style of website. In order to build a large dataset, the *URL* pattern is not used in our method. Our method is able to handle bilingual websites without *URL* pattern rules.

In addition, an English-Chinese dictionary with 64K words pairs is used in our experiments. Algorithm 3 needs a threshold $MAX\_P$ or

[1] http://www.httrack.com/

$MIN\_SIM$. It is very hard to tune the $MIN\_SIM$ because it varies a lot in different websites and language pairs. However, Table 1 shows that the number of parallel pages is smaller than that of English and Chinese pages. Here, for each website, the $MAX\_P$ is set to the number of Chinese pages (which is always smaller than that of English pages). In this way, the precision will never reach 100%, but it is more practical in a real application. As a result, in some experiments, we only report the F-score, and the precision and recall can be calculated as follows:

$$Precision = \frac{F_{score} \cdot (N_{Pairs} + MAX\_P)}{2 \cdot MAX\_P} \quad (4)$$

$$Recall = \frac{F_{score} \cdot (N_{Pairs} + MAX\_P)}{2 \cdot N_{Pairs}} \quad (5)$$

Where, $N_{Pairs}$ for each website is listed in the "Total pairs" column of Table 1.

## 4.2 Results and Analysis

### Performance of the Baseline

Let's start by presenting the performance of a baseline method as follows. The baseline only employs the internal translation similarity for parallel web pages mining. Algorithm 3 is also used to get the page pairs in baseline system. Here, the input $ETS(x, y)$ is replaced by $S_{in}(x, y)$. The parameter $\beta$ in Equation 2 is a discount factor. For different $\beta$ values, the performance of baseline system on six websites is shown in Figure 2. In the Figure 2, it shows that when $\beta$ is set to 0.6, the baseline system achieves the best performance. The precision, recall and F-score are 85.84%, 87.55% and 86.69% respectively. So in the following experiments, we always set $\beta$ to 0.6.

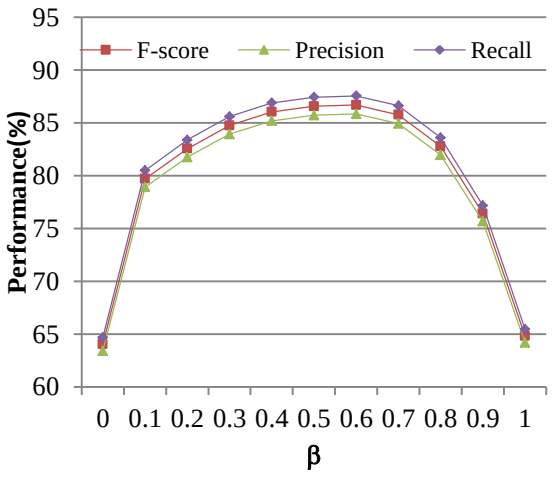



Figure 2 Performances of baseline system with different $\beta$ value

### Performance of Our Method

As described in Section 3, our method combines the internal with external translation similarity in estimating the final translation similarity (i.e., ETS) of two pages. So, the discount factor α in Equation (1) is important in our method. Besides, as shown in Algorithm 2, the iterative algorithm is used to calculate the similarity. Then, one question is that how many iterations are required in our algorithm. Figure 3 shows the performance of our method on each website. Its horizontal axis represents the number of iterations and the vertical axis represents the F-score. And for each website, the F-scores with different α (range from 0.2 to 0.8) are also reported in this figure. From Figure 3, it is very easy to find that the best iteration number is 3. For almost all the websites, the performance of our method achieves the maximal values and converges after the third iteration. In addition, Figure 3 also indicates that our method is robust for different websites. In the following experiments, the iteration number is set to 3.

Next, let's turn to the discount factor $\alpha$. Figure 4 reports the experimental results on the whole dataset. Here, the horizontal axis represents the discount factor α and the vertical axis represents the F-score. $\alpha = 0$ means that only the internal similarity is used in the algorithm, so the F-score equals to that in Figure 2 when $\beta = 0.6$. On the contrary, $\alpha = 1$ means that only the external similarity is used in the method, and the F-score is 80.20%. The performance is lower than the baseline system when only the external link information is used, but it is much better than the performance of the content-based method and structure-based method whose F-scores are 64.82% and 64.0% respectively. Besides, it is shown from Figure 4, the performance is improved significantly when the internal and external similarity measures are combined together. Furthermore, it is somewhat surprising that the discount factor α is not important as we previously expected. In fact, if we discard the cases that α equals to 0 or 1, the difference between the maximum and minimum F-score will be 0.76% which is very small. This finding indicates that the internal and external similarity can easily be combined and we don't need to make many efforts to tune this parameter when our method is applied to other websites. The reason of this phenomenon is that, no matter how much weight (i.e., 1- $\alpha$) was assigned to the internal similarity, the internal similarity always provides a relatively good initial

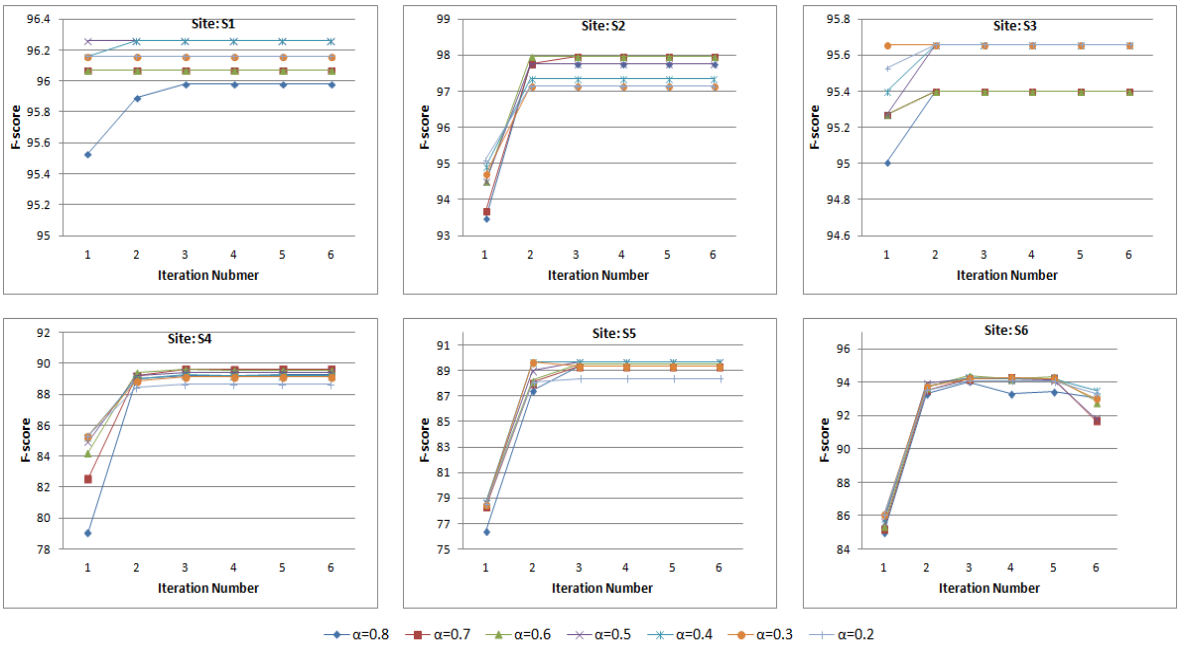



Figure 3 Experiment results of our method on each website

iterative direction. In the following experiments, the parameter α is set to 0.6.

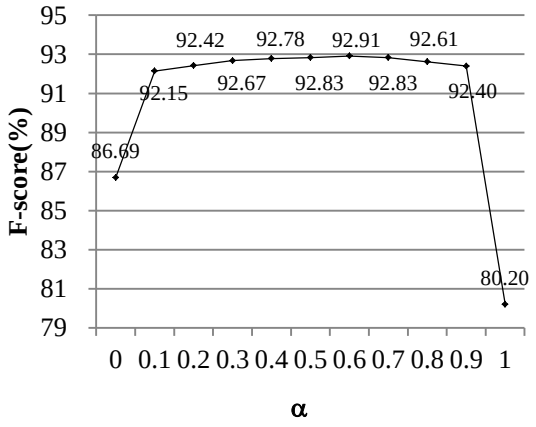



Figure 4 The F-scores of our method with different the value of $\alpha$

**The weight of pages**

The weight of the neighbor pages should also be considered. For example, in the most websites, it is very common that most of the web pages contain a hyperlink which points to the homepage of the website. While in most of the English/Chinese websites, almost every English page will link to the English homepage and each Chinese page will point to Chinese homepage. The English and Chinese homepages are probably parallel, but they will be helpless to find parallel web pages, because they are neighbors of almost every page in the site. On the contrary, sometimes the parallel homepages have negative effects on finding parallel pages They will increase the translation similarity of two pages which are not indeed mutual translations. So it is necessary to amend the Algorithm 1.

The weight of each page is calculated according to its popularity:

$$w(p) = log\frac{N+c}{Freq(p)+c} \quad (6)$$

where $w(p)$ indicates the weight of page $p$, $N$ is the number of all pages, $Freq(p)$ is the number of pages pointing to page $p$ and $c$ is a constant for smoothing.

In this paper, the weights of pages are used in two ways:

**Weight 1:** The 9th line of Algorithm 1 is amended by the page weight as follows:

$sum \leftarrow sum + ETS^{i-1}(x,y) \cdot (w(x)+w(y))/2$

**Weight 2:** The pages with low weight are removed from the input of Algorithm 1.

The experiment results are shown in Table 2.

Table 2 The effect of page weight

| Type | No Weight | Weight 1 | Weight 2 |
|---|---|---|---|
| F-score (%) | 92.91 | 92.78 | 92.75 |

Surprisingly, no big differences are found after the introduction of the page weight. The side effect of popular pages is not so large in our method. In the neighbor pages of a certain page, the popular pages are the minority. Besides, the iterative process makes our method more stable and robust.

**The impact of the size of bilingual lexicon**

The baseline system mainly combines the content-based similarity with structure similarity.

And two kinds of similarity measures are also used in our method. As Ma and Liberman (1999) pointed out, not all translators create translated pages that look like the original page which means that the structure similarity does not always work well. Compared to the structure similarity, the content-based is more reliable and has wider applicability. Furthermore, the bilingual lexicon is the only information that relates to the language pairs, and other features (such as structure and link information) are all language independent. So, it's important to investigate the effect of lexicon size in our method. We test the performance of our method with different size of the bilingual dictionary. The experiment results are shown in Figure 5. In this figure, the horizontal axis represents the bilingual lexicon size and the vertical axis represents the F-score. With the decline of the lexicon size, the performances of both the baseline method and our method are decreased. However, we can find that the descent rate of our method is smaller than that of the baseline. It indicates that our method does not need a big bilingual lexicon which is good news for the low-resource language pairs.

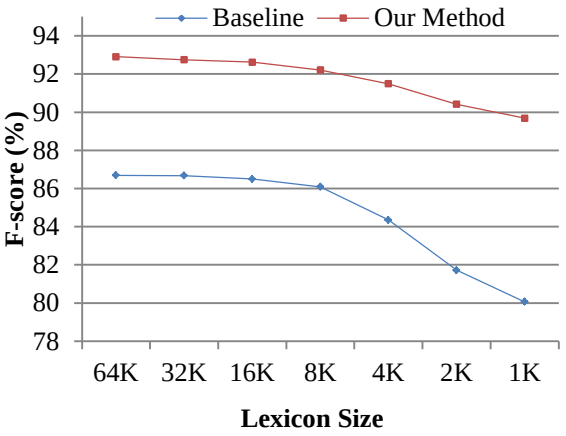



Figure 5 The impact of the size of bilingual lexicon

**Error analysis**

Errors occur when the two pages are similar in terms of structure, content and their neighbors. For example, Figure 6 illustrates a typical web page structure. There are 5 parts in the web page: $U$, $L$, $M$, $R$ and $B$. Part M always contains the main content of this page. While part $U$, $L$, $R$ and $B$ always contain some hyperlinks such as "home" in part $U$ and "About us" in part $B$. Links in $L$ and $R$ sometimes relate to the content of the page. For such a kind of non-parallel page pairs, let's assume that the two pages have the same structure (as shown in Figure 6). In addition, their content part $M$ is very short and contains the same or related topics. As a result, the links in other 4 parts are likely to be similar. In this case, our method is likely to regard the two pages as parallel.

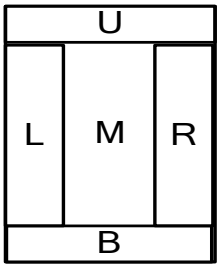



Figure 6 A typical web page structure

There are about 920 errors when our system obtains its best performance. By carefully investigating the error page pairs, we find that more than 90% errors fall into the category discussed above. The websites used in our experiments mainly come from Hong Kong government websites. Some government departments regularly publish quarterly or monthly work reports on one issue through their websites. These reports look very similar except the publish date and some data in them. The other 10% errors happen because of the particularity of the web pages, e.g. very short pages, broken pages and so on.

## 5 Conclusions and Future Work

Parallel corpora are valuable resources for a lot of NLP research problems and applications, such as MT and CLIR. This paper introduces an efficient and effective solution to bilingual language processing. We first explore how to extract parallel page pairs in bilingual websites with link information between web pages. Firstly, we hypothesize that the translation similarity of pages should be based on both internal and external translation similarity. Secondly, a novel iterative method is proposed to verify parallel page pairs. Experimental results show that our method is much more effective than the baseline system with 6.2% improvement on F-Score. Furthermore, our method has some significant contributions. For example, compared to previous work, our method does not depend on bilingual lexicons, and the parameters in our method have little effect on the final performance. These features improve the applicability of our method.

In the future work, we will study some method on extracting parallel resource from existing parallel page pairs, which are challenging tasks due to the diversity of page structures and styles. Besides, we will evaluate the effectiveness of our mined data on MT or other applications.

## Acknowledgments

This research work has been sponsored by National Natural Science Foundation of China (Grants No.61373097 and No.61272259), one National Natural Science Foundation of Jiangsu Province (Grants No.BK2011282), one Major Project of College Natural Science Foundation of Jiangsu Province (Grants No.11KJA520003) and one National Science Foundation of Suzhou City (Grants No.SH201212).

The corresponding author of this paper, according to the meaning given to this role by School of computer science and technology at Soochow University, is Yu Hong

## Reference

Chen, Jiang and Jianyun Nie. 2000. Automatic construction of parallel English-Chinese corpus for cross-language information retrieval. Proceedings of the sixth conference on Applied Natural Language Processing, 21–28.

Resnik, Philip and Noah A. Smith. 2003. The Web as a Parallel Corpus. Meeting of the Association for Computational Linguistics 29(3). 349–380.

Kit, Chunyu and Jessica Yee Ha Ng. 2007. An Intelligent Web Agent to Mine Bilingual Parallel Pages via Automatic Discovery of URL Pairing Patterns. Web Intelligence and Intelligent Agent Technology Workshops, 526–529.

Zhang, Ying, Ke Wu, Jianfeng Gao and Phil Vines. 2006. Automatic Acquisition of Chinese-English Parallel Corpus from the Web. Joint Proceedings of the Association for Computational Linguistics and the International Conference on Computational Linguistics, 420–431.

Nie, Jianyun, Michel Simard, Pierre Isabelle and Richard Durand. 1999. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the Web. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, 74–81.

Ma, Xiaoyi and Mark Y. Liberman. 1999. BITS: A Method for Bilingual Text Search over the Web. Machine Translation Summit VII.

Chen, Jisong, Rowena Chau and Chung-Hsing Yeh. 2004. Discovering Parallel Text from the World Wide Web. The Australasian Workshop on Data Mining and Web Intelligence, vol. 32, 157–161. Dunedin, New Zealand.

Jiang, Long, Shiquan Yang, Ming Zhou, Xiaohua Liu and Qingsheng Zhu. 2009. Mining Bilingual Data from the Web with Adaptively Learnt Patterns. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, vol. 2, 870–878.

Yanhui Feng, Yu Hong, Zhenxiang Yan, Jianmin Yao and Qiaoming Zhu. 2010. A novel method for bilingual web page acquisition from search engine web records. Proceedings of the 23rd International Conference on Computational Linguistics: Posters, 294–302.

Zhao, Bing and Stephan Vogel. 2002. Adaptive Parallel Sentences Mining from Web Bilingual News Collection. IEEE International Conference on Data Mining, 745–748.

Smith, Jason R., Chris Quirk and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 403–411.

Bharadwaj, Rohit G. and Vasudeva Varma. 2011. Language independent identification of parallel sentences using wikipedia. Proceedings of the 20th International Conference Companion on World Wide Web, 11–12. Hyderabad, India.

Gusfield, Dan. 1997. Algorithms on Strings, Trees and Sequences: Computerss Science and Computational Biology. Cambridge University Press

Shi, Lei, Cheng Niu, Ming Zhou and Jianfeng Gao. 2006. A DOM Tree Alignment Model for Mining Parallel Data from the Web. Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, 489–496.

# Effective Selection of Translation Model Training Data

**Le Liu Yu Hong* Hao Liu Xing Wang Jianmin Yao**
School of Computer Science & Technology, Soochow University, China
{20124227052, hongy, 20134227035, 20114227047, jyao}@suda.edu.cn

## Abstract

Data selection has been demonstrated to be an effective approach to addressing the lack of high-quality bitext for statistical machine translation in the domain of interest. Most current data selection methods solely use language models trained on a small scale in-domain data to select domain-relevant sentence pairs from general-domain parallel corpus. By contrast, we argue that the relevance between a sentence pair and target domain can be better evaluated by the combination of language model and translation model. In this paper, we study and experiment with novel methods that apply translation models into domain-relevant data selection. The results show that our methods outperform previous methods. When the selected sentence pairs are evaluated on an end-to-end MT task, our methods can increase the translation performance by 3 BLEU points.

## 1 Introduction

Statistical machine translation depends heavily on large scale parallel corpora. The corpora are necessary priori knowledge for training effective translation model. However, domain-specific machine translation has few parallel corpora for translation model training in the domain of interest. For this, an effective approach is to automatically select and expand domain-specific sentence pairs from large scale general-domain parallel corpus. The approach is named Data Selection. Current data selection methods mostly use language models trained on small scale in-domain data to measure domain relevance and select domain-relevant parallel sentence pairs to expand training corpora. Related work in literature has proven that the expanded corpora can substantially improve the performance of machine translation (Duh et al., 2010; Haddow and Koehn, 2012).

However, the methods are still far from satisfactory for real application for the following reasons:

- There isn't ready-made domain-specific parallel bitext. So it's necessary for data selection to have significant capability in mining parallel bitext in those assorted free texts. But the existing methods seldom ensure parallelism in the target domain while selecting domain-relevant bitext.
- Available domain-relevant bitext needs keep high domain-relevance at both the sides of source and target language. But it's difficult for current method to maintain two-sided domain-relevance when we aim at enhancing parallelism of bitext.

In a word, current data selection methods can't well maintain both parallelism and domain-relevance of bitext. To overcome the problem, we first propose the method combining translation model with language model in data selection. The language model measures the domain-specific generation probability of sentences, being used to select domain-relevant sentences at both sides of source and target language. Meanwhile, the translation model measures the translation probability of sentence pair, being used to verify the parallelism of the selected domain-relevant bitext.

## 2 Related Work

The existing data selection methods are mostly based on language model. Yasuda et al. (2008) and Foster et al. (2010) ranked the sentence pairs in the general-domain corpus according to the perplexity scores of sentences, which are computed with respect to in-domain language models. Axelrod et al. (2011) improved the perplexity-based approach and proposed bilingual cross-entropy difference as a ranking function with in- and general- domain language models. Duh et al. (2013) employed the method of (Axelrod et al.,

* Corresponding author



*Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 569–573,
Baltimore, Maryland, USA, June 23-25 2014. 

2011) and further explored neural language model for data selection rather than the conventional n-gram language model. Although previous works in data selection (Duh et al., 2013; Koehn and Haddow, 2012; Axelrod et al., 2011; Foster et al., 2010; Yasuda et al., 2008) have gained good performance, the methods which only adopt language models to score the sentence pairs are sub-optimal. The reason is that a sentence pair contains a source language sentence and a target language sentence, while the existing methods are incapable of evaluating the mutual translation probability of sentence pair in the target domain. Thus, we propose novel methods which are based on translation model and language model for data selection.

## 3 Training Data Selection Methods

We present three data selection methods for ranking and selecting domain-relevant sentence pairs from general-domain corpus, with an eye towards improving domain-specific translation model performance. These methods are based on language model and translation model, which are trained on small in-domain parallel data.

### 3.1 Data Selection with Translation Model

Translation model is a key component in statistical machine translation. It is commonly used to translate the source language sentence into the target language sentence. However, in this paper, we adopt the translation model to evaluate the translation probability of sentence pair and develop a simple but effective variant of translation model to rank the sentence pairs in the general-domain corpus. The formulations are detailed as below:

$$P(e|f) = \frac{\epsilon}{(l_f+1)^{l_e}} \prod_{j=1}^{l_e} \sum_{i=0}^{l_f} t(e_j|f_i) \quad (1)$$

$$R = \sqrt[l_e]{P(e|f)} \quad (2)$$

Where $P(e|f)$ is the translation model, which is IBM Model 1 in this paper, it represents the translation probability of target language sentence $e$ conditioned on source language sentence $f$. $l_e$ and $l_f$ are the number of words in sentence $e$ and $f$ respectively. $t(e_j|f_i)$ is the translation probability of word $e_j$ conditioned on word $f_i$and is estimated from the small in-domain parallel data. The parameter $\epsilon$ is a constant and is assigned with the value of 1.0. $R$ is the length-normalized IBM Model 1, which is used to score general-domain sentence pairs. The sentence pair with higher score is more likely to be generated by in-domain translation model, thus, it is more relevant to the in-domain corpus and will be remained to expand the training data.

### 3.2 Data Selection by Combining Translation and Language model

As described in section 1, the existing data selection methods which only adopt language model to score sentence pairs are unable to measure the mutual translation probability of sentence pairs. To solve the problem, we develop the second data selection method, which is based on the combination of translation model and language model. Our method and ranking function are formulated as follows:

$$P(e,f) = P(e|f) \times P(f) \quad (3)$$

$$R = \sqrt[l_e]{P(e|f)} \times \sqrt[l_f]{P(f)} \quad (4)$$

Where $P(e,f)$ is a joint probability of sentence $e$ and $f$ according to the translation model $P(e|f)$ and language model $P(f)$, whose parameters are estimated from the small in-domain text. $R$ is the improved ranking function and used to score the sentence pairs with the length-normalized translation model $P(e|f)$ and language model $P(f)$. The sentence pair with higher score is more similar to in-domain corpus, and will be picked out.

### 3.3 Data Selection by Bidirectionally Combining Translation and Language Models

As presented in subsection 3.2, the method combines translation model and language model to rank the sentence pairs in the general-domain corpus. However, it does not evaluate the inverse translation probability of sentence pair and the probability of target language sentence. Thus, we take bidirectional scores into account and simply sum the scores in both directions.

$$R = \sqrt[l_e]{P(e|f)} \times \sqrt[l_f]{P(f)} + \sqrt[l_f]{P(f|e)} \times \sqrt[l_e]{P(e)} \quad (5)$$

Again, the sentence pairs with higher scores are presumed to be better and will be selected to incorporate into the domain-specific training data. This approach makes full use of two translation models and two language models for sentence pairs ranking.

## 4 Experiments

### 4.1 Corpora

We conduct our experiments on the Spoken Language Translation English-to-Chinese task. Two corpora are needed for the data selection. The in-domain data is collected from CWMT09, which consists of spoken dialogues in a travel setting, containing approximately 50,000 parallel sentence pairs in English and Chinese. Our general-domain corpus mined from the Internet contains 16 million sentence pairs. Both the in- and general- domain corpora are identically tokenized (in English) and segmented (in Chinese)[1]. The details of corpora are listed in Table 1. Additionally, we evaluate our work on the 2004 test set of "863" Spoken Language Translation task ("863" SLT), which consists of 400 English sentences with 4 Chinese reference translations for each. Meanwhile, the 2005 test set of "863" SLT task, which contains 456 English sentences with 4 references each, is used as the development set to tune our systems.

| Bilingual Corpus | #sentence | | #token | |
|---|---|---|---|---|
| | Eng | Chn | Eng | Chn |
| In-domain | 50K | 50K | 360K | 310K |
| General-domain | 16M | 16M | 3933M | 3602M |

Table 1. Data statics

### 4.2 System settings

We use the NiuTrans [2] toolkit which adopts GIZA++ (Och and Ney, 2003) and MERT (Och, 2003) to train and tune the machine translation system. As NiuTrans integrates the mainstream translation engine, we select hierarchical phrase-based engine (Chiang, 2007) to extract the translation rules and carry out our experiments. Moreover, in the decoding process, we use the NiuTrans decoder to produce the best outputs, and score them with the widely used NIST mteval131a[3] tool. This tool scores the outputs in several criterions, while the case-insensitive BLEU-4 (Papineni et al., 2002) is used as the evaluation for the machine translation system.

### 4.3 Translation and Language models

Our work relies on the use of in-domain language models and translation models to rank the sentence pairs from the general-domain bilingual training set. Here, we employ ngram language model and IBM Model 1 for data selection. Thus, we use the SRI Language Modeling Toolkit (Stolcke, 2002) to train the in-domain 4-gram language model with interpolated modified Kneser-Ney discounting (Chen and Goodman, 1998). The language model is only used to score the general-domain sentences. Meanwhile, we use the language model training scripts integrated in the NiuTrans toolkit to train another 4-gram language model, which is used in MT tuning and decoding. Additionally, we adopt GIZA++ to get the word alignment of in-domain parallel data and form the word translation probability table. This table will be used to compute the translation probability of general-domain sentence pairs.

### 4.4 Baseline Systems

As described above, by using the NiuTrans toolkit, we have built two baseline systems to fulfill "863" SLT task in our experiments. The In-domain baseline trained on spoken language corpus has 1.05 million rules in its hierarchical-phrase table. While, the General-domain baseline trained on 16 million sentence pairs has a hierarchical phrase table containing 1.7 billion translation rules. These two baseline systems are equipped with the same language model which is trained on large-scale monolingual target language corpus. The BLEU scores of the In-domain and General-domain baseline system are listed in Table 2.

| Corpus | Hierarchical phrase | Dev | Test |
|---|---|---|---|
| In-domain | 1.05M | 15.01 | **21.99** |
| General-domain | 1747M | 27.72 | **34.62** |

Table 2. Translation performances of In-domain and General-domain baseline systems

The results show that General-domain system trained on a larger amount of bilingual resources outperforms the system trained on the in-domain corpus by over 12 BLEU points. The reason is that large scale parallel corpus maintains more bilingual knowledge and language phenomenon, while small in-domain corpus encounters data sparse problem, which degrades the translation performance. However, the performance of General-domain baseline can be improved further. We use our three methods to refine the general-domain corpus and improve the translation performance in the domain of interest. Thus, we build several contrasting systems trained on refined training data selected by the following different methods.

1http://www.nlplab.com/NiuPlan/NiuTrans.YourData.ch.html
2http://www.nlplab.com/NiuPlan/NiuTrans.ch.html#download
3 http://ww.itl.nist.gov/iad/mig/tools

- **Ngram**: Data selection by 4-gram LMs with Kneser-Ney smoothing. (Axelrod et al., 2011)
- **Neural net**: Data selection by Recurrent Neural LM, with the RNNLM Tookit. (Duh et al., 2013)
- **Translation Model (TM)**: Data selection with translation model: IBM Model 1.
- **Translation model and Language Model (TM+LM)**: Data selection by combining 4-gram LMs with Kneser-Ney smoothing and IBM model 1(equal weight).
- **Bidirectional TM+LM**: Data selection by bidirectionally combining translation and language models (equal weight).

### 4.5 Results of Training Data Selection

We adopt five methods for extracting domain-relevant parallel data from general-domain corpus. Using the scoring methods, we rank the sentence pairs of the general-domain corpus and select only the top N = {50k, 100k, 200k, 400k, 600k, 800k, 1000k} sentence pairs as refined training data. New MT systems are then trained on these small refined training data. Figure 1 shows the performances of systems trained on selected corpora from the general-domain corpus. The horizontal coordinate represents the number of selected sentence pairs and vertical coordinate is the BLEU scores of MT systems.

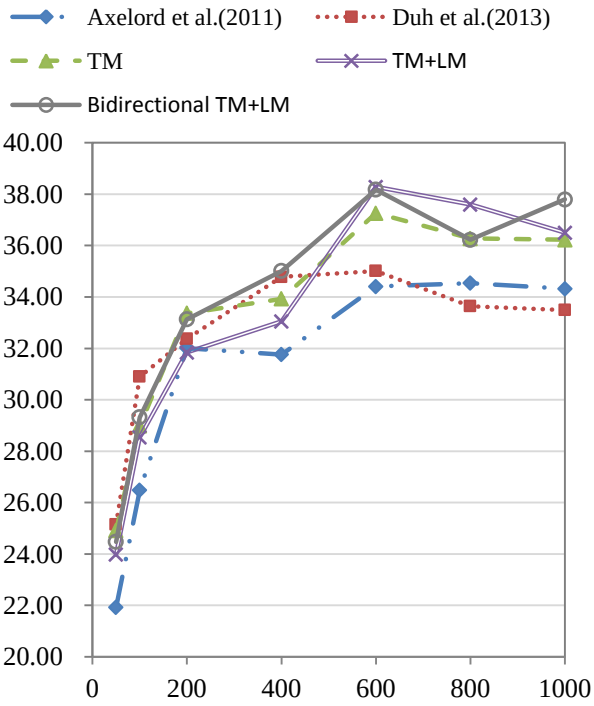



Figure 1. Results of the systems trained on only a subset of the general-domain parallel corpus.

From Figure 1, we conclude that these five data selection methods are effective for domain-specific translation. When top 600k sentence pairs are picked out from general-domain corpus to train machine translation systems, the systems perform higher than the General-domain baseline trained on 16 million parallel data. The results indicate that more training data for translation model is not always better. When the domain-specific bilingual resources are deficient, the domain-relevant sentence pairs will play an important role in improving the translation performance.

Additionally, it turns out that our methods (**TM**, **TM+LM** and **Bidirectional TM+LM)** are indeed more effective in selecting domain-relevant sentence pairs. In the end-to-end SMT evaluation, **TM** selects top 600k sentence pairs of general-domain corpus, but increases the translation performance by 2.7 BLEU points. Meanwhile, the **TM+LM** and **Bidirectional TM+LM** have gained 3.66 and 3.56 BLEU point improvements compared against the general-domain baseline system. Compared with the mainstream methods (**Ngram** and **Neural net)**, our methods increase translation performance by nearly 3 BLEU points, when the top 600k sentence pairs are picked out. Although, in the figure 1, our three methods are not performing better than the existing methods in all cases, their overall performances are relatively higher. We therefore believe that combining in-domain translation model and language model to score the sentence pairs is well-suited for domain-relevant sentence pair selection. Furthermore, we observe that the overall performance of our methods is gradually improved. This is because our methods are combining more statistical characteristics of in-domain data in ranking and selecting sentence pairs. The results have proven the effectiveness of our methods again.

## 5 Conclusion

We present three novel methods for translation model training data selection, which are based on the translation model and language model. Compared with the methods which only employ language model for data selection, we observe that our methods are able to select high-quality domain-relevant sentence pairs and improve the translation performance by nearly 3 BLEU points. In addition, our methods make full use of the limited in-domain data and are easily implemented. In the future, we are interested in applying

our methods into domain adaptation task of statistical machine translation in model level.

## Acknowledgments

This research work has been sponsored by two NSFC grants, No.61373097 and No.61272259, and one National Science Foundation of Suzhou (Grants No. SH201212).

## Reference

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK, July. Association for Computational Linguistics.

Peter F.Brown, Vincent J. Della Pietra, Stephen A. Della Pietra and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 1993, 19(2): 263-311.

Stanley Chen and Joshua Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. *Technical Report 10-98, Computer Science Group, Harvard University.*

Moore Robert C, Lewis William. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*. Association for Computational Linguistics, 2010: 220-224.

Chiang David. A hierarchical phrase-based model for statistical machine translation. 2005. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages: 263-270. Association for Computational Linguistics.

Kevin Duh, Graham Neubig, Katsuhito Sudoh and Hajime Tsukada. Adaptation Data Selection using Neural Language Models: Experiments in Machine Translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 678-683, Sofia, Bulgaria, August 4-9 2013.

Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010.Analysis of translation model adaptation for statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT) - Technical Papers Track*.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. *Empirical Methods in Natural Language Processing*.

Barry Haddow and Philipp Koehn. 2012. Analysing the effect of out-of-domain data on smt systems. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 422–432, Montreal, Canada, June. Association for Computational Linguistics.

Och, Franz Josef, and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics* 29.1 (2003): 19-51.

Och, Franz Josef. Minimum error rate training in statistical machine translation. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*-Volume 1. Association for Computational Linguistics, 2003.

Philipp Koehn and Barry Haddow. 2012. Towards effective use of training data in statistical machine translation. In *WMT*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *ACL*.

Andreas Stolcke. 2002. SRILM - An extensible language modeling toolkit. *Spoken Language Processing*.

Tong Xiao, Jingbo Zhu, Hao Zhang and Qiang Li. NiuTrans: an open source toolkit for phrase-based and syntax-based machine translation. In *Proceedings of the ACL 2012 System Demonstrations*. Association for Computational Linguistics, 2012: 19-24.

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumita. 2008. Method of selecting training data to build a compact and efficient translation model. *International Joint Conference on Natural Language Processing.*

# Skill Inference with Personal and Skill Connections

**Zhongqing Wang[†], Shoushan Li[†*], Hanxiao Shi[‡], and Guodong Zhou[†]**
[†] Natural Language Processing Lab, School of Computer Science and Technology,
Soochow University, China
[‡] School of Computer Science and Information Engineering,
Zhejiang Gongshang University, China
{wangzq.antony, shoushan.li}@gmail.com
hxshi@mail.zjgsu.edu.cn, gdzhou@suda.edu.cn

## Abstract

Personal skill information on social media is at the core of many interesting applications. In this paper, we propose a factor graph based approach to automatically infer skills from personal profile incorporated with both personal and skill connections. We first extract personal connections with similar academic and business background (e.g. co-major, co-university, and co-corporation). We then extract skill connections between skills from the same person. To well integrate various kinds of connections, we propose a joint prediction factor graph (JPFG) model to collectively infer personal skills with help of personal connection factor, skill connection factor, besides the normal textual attributes. Evaluation on a large-scale dataset from LinkedIn.com validates the effectiveness of our approach.

## 1 Introduction

With the large amount of user-generated content (UGC) published online every day in the context of social networks (Tan et al., 2011; Luo et al., 2013), such online social networks (e.g., Twitter, Facebook, and LinkedIn) have significantly enlarged our social circles and much affected our everyday life. One popular and important type of UGC is the personal profile, where people post their detailed information, such as education, experience and other personal information, on online portals. Social websites like Facebook.com and LinkedIn.com have created a viable business as profile portals, with the popularity and success largely attributed to their comprehensive personal profiles.

Obviously, online personal profiles can help people connect with others of similar backgrounds and provide valuable resources for businesses, especially for personnel resource managers to find talents (Yang et al., 2011a; Guy et al., 2010). In the profiles, the personal skill information is the most important aspect to reflect the expertise of a person. However, few social platforms allow users to manually attach such personal skill information into their personal profiles. For example, in our collected dataset, 91.8% skills appear less than 10 times. Even the distribution of the top 10 frequently occurring skills is asymmetric, and only 43.1% people attach skills on their profiles. For this regard, it is highly desirable to develop reliable methods to automatically infer personal skills for personal profiles.

Although it is straightforward to recast skill inference as a standard text classification problem, i.e., predicting the skills with the profile text alone, personal profiles usually are poorly organized, even with critical information missing. Thus, it is challenging to infer skills given the limited information from the profile texts. We propose two assumptions to address above challenges by incorporating additional connection information between persons and skills:

- People are always connected to others with similar academic and business backgrounds (e.g. co-major, co-corporation). For example if there is co-major, co-university, or co-corporation relationship between two persons, it is very likely that they may share similar skills. Therefore, it is reasonable to resort to personal connection information to improve the performance of skill inference.

*corresponding author





*Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 520–529, Dublin, Ireland, August 23-29 2014.

- One person tends to have some related skills. For example, it is very likely that C++, C, and Python programming languages may co-occur in the one's profile, i.e., if a person has skill C++, it is highly possible that he would have the skills such as C or Python. Thus, it is useful to integrate skill connection information when inferring personal skills.

Based on these assumptions, we propose a Joint Prediction Factor Graph (JPFG) model, which collectively predicts personal skills with help of both personal and skill connections. In particular, the JPFG model provides a general framework to integrate three kinds of knowledge, i.e. local textual attribute functions of an individual person, personal connection factors between persons, and skill connection factors between skills, in collectively inferring personal skills. Specially, we extract personal connections with similar academic and business background (e.g. co-major, co-corporation). We then extract skill connections between skills from same person. Evaluation on a large-scale data set from LinkedIn.com indicates that our JPFG model can significantly improve the performance of personal skill inference.

The remainder of this paper is structured as follows. We review the related work in Section 2. In Section 3, we introduce the data collection. In Section 4, we give the problem definition and some analysis on the task of personal skill reference. In Section 5, we propose the JPFG model and corresponding algorithms for parameter estimation and prediction. In Section 6, we present our experimental results. In Section 7, we summarize our work and discuss future directions.

## 2 Related Works

In this section, we briefly review related studies in expert finding, social tag suggestion and factor graph model.

### 2.1 Expert Finding

Expert finding aims to find right persons with appropriate skills or knowledge, i.e. "Who are the experts on topic X?" TREC-2005 and TREC-2006 have provided a common platform for researchers to empirically evaluate methods and techniques on expert finding (Soboroff et al, 2006; Zhang et al., 2007a).

In the literature, expert finding tends to consider each skill individually and seeks the most authority experts for each skill. Thus, expert finding is always considered as a ranking process, i.e., ranking the experts from the candidates who are most suitable for the skill (Balog and Rijke, 2007). For example, Campbell et al. (2003) investigated the issue of expert finding in an email network. They utilized the link between email authors and receivers to improve the expert finding performance.

Besides that link structure-based algorithms, such as PageRank and HITS, are employed to analyze the relationship of the link-relationship graph, social networks are utilized to improve the performance of expert finding. Zhang et al. (2007a) proposed a unified propagation-based approach to address the issue of expert finding in a social network, considering both personal local and network information (e.g. the relationship between persons).

Expert finding is in nature different from skill inference. Our study predicts various skills attachable to a person collectively with both personal and skill connections among people. One distinguishing characteristics of our study is that several skills from a person are simultaneously modeled and the relationship among these skills is fully leveraged in the inference.

### 2.2 Social Tag Suggestion

Social tag suggestion aims to extract proper tags from social media and can thus help people organize their information in an unconstrained manner (Ohkura et al., 2006; Si et al., 2010). Ohkura et al. (2006) created a multi-tagger to determine whether a particular tag from a candidate tag list should be attached to a weblog. Lappas et al. (2011) proposed a social endorsement-based approach to generate social tags from Twitter.com and Flickr.com where various kinds of information in recommendations and comments are used. Liu et al. (2012) propose a probabilistic model to connect the semantic relations between words and tags of microblog, and takes the social network structure as regularization. Li et al., (2012) propose to model context-aware relations of tags for suggestion by regarding resource content as context of tags.

Different from above researches, our study is forced on skill inference instead of traditional tag suggestion. Basically, the social connections in skill inference are much different from those in social tagging. In our study, we use co-major, co-title and other academic and business relationships to build the social connections. Meanwhile, there are also few researches concern to propose a joint model to leverage both personal and skill connections.

### 2.3 Factor Graph Model

Among various approaches investigated in social networks in the last several years (Leskovec et al., 2010; Lu et al., 2010; Lampos et al., 2013; Guo et al., 2013), Factor Graph Model (FGM) becomes an effective way to represent and optimize the relationship in social networks (Dong et al., 2012; Yang et al., 2012b) via a graph structure. Tang et al. (2011a) and Zhuang et al. (2012) formalized the problem of social relationship learning as a semi-supervised framework, and proposed Partially-labeled Pairwise Factor Graph Model (PLP-FGM) for inferring the types of social ties. Tang et al. (2013) further proposed a factor graph based distributed learning method to construct a conformity influence model and formalize the effects of social conformity in a probabilistic way.

Different from previous studies, this paper proposes a pairwise factor graph model to collectively infer personal skills with both social connection factor and skill connection factor.

## 3 Data Construction

We collect our data set from LinkedIn.com. It contains a large number of personal profiles generated by users, containing various kinds of information, such as personal Summary, Experience, Education, and Skills & Expertise. We do not collect personal names in public profiles to protect people's privacy.

The dataset contains 7,381 personal profiles, among which only 3,182 profiles (43.1% of all the profiles) show the Skills & Expertise field. In this study, we adopt only these profiles in all our experiments. As a result, we get 6,863 skills in total, among which 6,299 skills (91.8% of them) appear less than 10 times. Among the remaining 564 skills, we select top 10 frequently occurring skills as the candidate personal skills in this study (Since the remaining 554 skills only appear less than 250 times in total, it is difficult to build an effective classifier for them). Table 1 illustrates the statistics.

| Skill | Number | Ratio |
|---|---|---|
| Semiconductors | 948 | 0.298 |
| IC | 369 | 0.116 |
| Thin Films | 328 | 0.103 |
| Characterization | 326 | 0.102 |
| CMOS | 311 | 0.098 |
| Matlab | 287 | 0.090 |
| Microsoft Office | 283 | 0.089 |
| Manufacturing | 278 | 0.087 |
| Design of Experiments | 262 | 0.082 |
| Semiconductor Industry | 250 | 0.079 |

Table 1: The distribution of the candidate personal skills

From Table 1, we can see that the skill distribution in the personal profiles is asymmetric. For example, the Semiconductor skill occurs about 1,000 times, taking 29.8%, while the Semiconductor Industry skill occurs 250 times only, taking 7.9%.

## 4 Problem Definition and Analysis

Before presenting our approach for skill inference, we first give the definition of the problem, and convey a series of discoveries we observed from the data.

### 4.1 Problem Definition

We first introduce some necessary definitions and then formulate of the problem.

**Definition 1:** *Skill inference*. In principle, we cast skill inference as a skill prediction problem. Since one person might have several skills, we build several vectors for a person and each vector is designed to determine whether the corresponding skill is appropriate for the person or not ("Positive" means that the person has the target skill, whereas "Negative" stands for the opposite). Note that the number of vectors for a person is equal to the number of candidate skills. For example, suppose we have m persons and n candidate skills in the dataset, we totally build vectors to represent if these skills are attached in these persons' profiles.

**Definition 2:** *Textual information*. We use texts of Summary and Experience as the textual information for our research. Texts of Summary and Experience are unstructured information, while texts of Skills & Expertise are structured information. However, some skills in the Skill & Expertise fields may not be mentioned in the Summary and Experience fields.

**Definition 3:** *Personal connections*. We can explicitly extract four kinds of personal relationships between two persons from the Education and Experience fields, as follows:

- *co_major*, which denotes that two persons have the same major at school
- *co_univ*, which denotes that two persons graduated from the same university
- *co_title*, which denotes that two persons have the same title in a corporation.
- *co_corp*, which denotes that two persons work in the same corporation.

**Definition 4:** *Skill connections*. We extract skill connections from same person. That is, if two vectors are from the same person with different skills, we consider these two vectors share skill connections (e.g. John has IC and Thin Films skills).

**Learn task:** Given the textual information of each profile, the personal connections between profiles, and skill connections of skill from same persons, the goal is to infer the skill through the above information.

To learn the skill inference model, there are several requirements. First, the skills of persons are related to multiple factors, e.g., network structure, personal connections, and skill connections, it is important to find a unified model which is able to incorporate all the information together. Second, the algorithm to learn the inference model should be efficient. In practice, the scale of the social network might be very large.

### 4.2 Statistics and Observations

In the following, we give some statistics and observations on personal and skill connections.

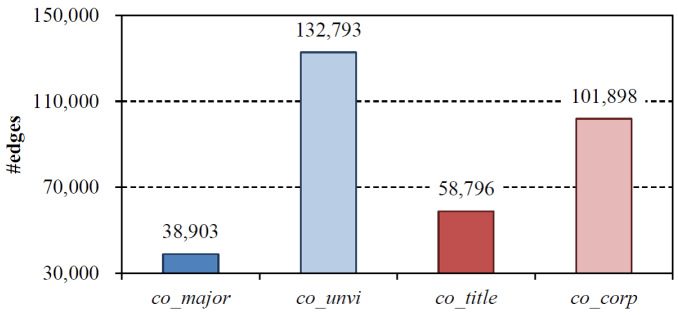



Figure 1: The statistic of personal connection edges in our dataset

**Statistics of personal connections:** Figure 1 gives the statistics of personal connection edges. It shows that with 3,182 profiles, there exist 332,390 personal connection edges. Besides, among all the

four relations, co_major, co_unvi, co_title, and co_corp occupy 11.7%, 40.0%, 17.7% and 30.6% respectively.

**Observations of skills connections:** To validate the tendency of a person sharing similar skills, we use PMI (Point-wise Mutual Information) to measure the co-occurrence between two skills. As a popular way to measure the co-occurrence between a pair (Turney, 2002), PMI is calculated as follows:

$$PMI(i,j) = \log\left(N\frac{P(i\&j)}{P(i)P(j)}\right) \tag{1}$$

$N$ is the number of profiles, $P(i\&j)$ denotes the probability of the skills (i.e., $i$ and $j$) co-occurrence in a person's profile, while $P(i)$ denotes the probability of the skill $i$ appearing in a person's profile.

| **Skill** $i$ | **Skill** $j$ | **PMI** |
|---|---|---|
| C | COMS | 1.711 |
| Thin Films | Characterization | 1.624 |
| Thin Films | Design of Experiments | 1.543 |
| Semiconductor Industry | IC | 1.345 |
| Semiconductor Industry | Design of Experiments | 1.345 |
| IC | Microsoft Office | -2.390 |
| CMOS | Microsoft Office | -2.627 |
| Semiconductor Industry | Matlab | -3.112 |
| Average PMI score | | 0.190 |

Table 2: The top-5 and bottom-3 co-occurred skill pairs with their PMI scores

Table 2 lists the top-5 and bottom-3 co-occurred skill pairs with their PMI scores, together with the average PMI score. From this table, we can see that if two skills are related, e.g., "IC" and "CMOS", these two skills tend to co-occur in one person's profile, vice versa.

## 5 Joint Prediction Factor Graph Model

In this section, we propose a Joint Prediction Factor Graph (JPFG) model for learning and predicting the skills with personal and skill connection information besides local textual information.

### 5.1 Model

We formalize the problem of skill prediction using a pairwise factor graph model, and our basic idea of defining the correlations is to use different types of factor functions (i.e., personal connection factor, and skill connection factor). Here, the objective function $P_\theta(Y|X,G)$ is defined based on the joint probability of the factor functions, and the problem of collective skill inference model learning is cast as learning model parameters $\theta$ that maximizes the joint probability of skills based on the input continuous dynamic network.

Since directly maximizing the conditional probability $P_\theta(Y|X,G)$ is often intractable, we factorize the "global" probability as a product of "local" factor functions, each of which depends on a subset of the variables in the graph (Tang et al., 2013). In particular, we use three kinds of functions to represent the local textual information of the vector (local textual attribute function), personal connection information between vectors (personal connection factor) and skill connection information between skills (skill connection factor), respectively. We now briefly introduce the ways to define the above three functions.

**Local textual attribute functions** $f(x_{ij},y_i)_j$ **:** It denotes the attribute value associated with each person $i$. Here, we define the local textual attribute as a feature (Lafferty et al., 2001) and accumulate all the attribute functions to obtain local entropy for a person:

$$\frac{1}{Z_1}\exp\left(\sum_i\sum_k \alpha_k f_k(x_{ik},y_i)\right) \tag{2}$$

Where $\alpha_k$ is the function weight, representing the influence degree of the attribute $k$. For simplicity, we use word unigrams of a text as the basic textual attributes.

**Personal connection factor function** $g(y_i, y_j)$ **:** For the personal correlation factor function, we define it through the pairwise network structure. That is, if a person $i$ and another person $j$ have a personal relationship, we define a personal connection factor function as follows:

$$g(y_i, y_j) = \exp\left\{\beta_{ij}(y_i - y_j)^2\right\} \tag{3}$$

The personal connections are defined Section 4, i.e., co_major, co_univ, co_title, and co_corp. We define that if two persons have at least one personal connection edge, they have a personal relationship. In addition, $\beta_{ij}$ is the weight of the function, representing the influence degree of $i$ on $j$.

**Skill connection factor function** $h(y_i, y_j)$**:** For the skill connection factor function, we define it through the pairwise network structure. That is, if vector $i$ and vector $j$ are from the same person with different skills, we define their skill connection influence factor function as follows:

$$h(y_i, y_j) = \exp\left\{\gamma_{ij}(y_i - y_j)^2\right\} \tag{4}$$

Where $\gamma_{ij}$ is the function weight, representing the influence degree of $i$ on $j$.

By the above defined correlations, we can construct the graphical structure in the factor model. According to the Hammersley-Clifford theorem (Hammersley and Clifford, 1971), we integrate all the factor functions and obtain the following log-likelihood objective function:

$$\begin{aligned} L(\theta) &= \log_\theta P(Y|X,G) \\ &= \frac{1}{Z_1}\sum_i\sum_k \alpha_k f_k(x_{ik}, y_i) \\ &+ \frac{1}{Z_2}\sum_i \sum_{j\in NB(i)} \exp\left\{\beta_{ij}(y_i - y_j)^2\right\} \\ &+ \frac{1}{Z_3}\sum_i \sum_{k\in SAME(i)} \exp\left\{\gamma_{ik}(y_i - y_k)^2\right\} - \log Z \end{aligned} \tag{5}$$

Where $(i, j)$ is a pair derived from the input network, $Z = Z_1 Z_2 Z_3$ is a normalization factor and $\theta = (\{\alpha\}, \{\beta\}, \{\gamma\})$ indicates a parameter configuration, $NB(i)$ denotes the set of social relationship neighbors nodes of i (personal connection), and $SAME(i)$ denotes the set of the node with the same person of $i$ (skill connection).

### 5.2 Learning and Prediction

**Model Learning:** Learning of the factor model is to find the best configuration for free parameters $\theta = (\{\alpha\}, \{\beta\}, \{\gamma\})$ that maximizes the log likelihood objective function $L(\theta)$.

$$\theta^* = \arg\max L(\theta) \tag{6}$$

As the network structure in a social network can be arbitrary (e.g. possible of containing cycles), we use the Loopy Belief Propagation (LBP) algorithm (Tang et al., 2011a) to approximate the marginal distribution. To explain how we learn the parameters, we can get the gradient of each $\beta_k$ with regard to the objective function (Eq. 5), taking $\beta$ (the weight of the personal connection factor function $g(y_i, y_j)$) as an example:

$$\frac{L(\theta)}{\beta_k} = E[g(i,j)] + E_{\beta_k P(Y|X,G)}[g(i,j)] \tag{7}$$

Where $E[g(i,j)]$ is the expectation of factor function $g(i,j)$ given the data distribution in the input network and $E_{\beta_k P(Y|X,G)}[g(i,j)]$ represents the expectation under the distribution learned by the model, i.e., $P(y_i|X,G)$ .

With the marginal probabilities, the gradient is obtained by summing up all triads (similar gradients can be derived for parameter $\alpha_k$ and $\gamma_{ij}$). It is worth noting that we need to perform the LBP process

twice in each iteration. The first run to estimate the marginal distribution of unknown variables $y_i =?$ and the second one is to estimate the marginal distribution over all pairs. Finally, with the obtained gradient, we update each parameter with a learning rate $\eta$.

**Skill Prediction:** We can see that in the learning process, additional loopy belief propagation is used to infer the label of unknown relationships. After learning, all unknown skills are assigned with labels that maximize the marginal probabilities (Tang et al., 2011b), i.e.,

$$Y^* = \arg\max L(Y|X, G, \theta) \tag{8}$$

## 6 Experimentation

In this section, we first introduce the experimental setting, and then evaluate the performance of our proposed JPFG model with both personal and skill connection information.

### 6.1 Experimental Setting

As described in Section 3, the experimental data are collected from LinkedIn.com. With top 10 frequently used skills as candidate skills in all our experiments, we randomly select 2,000 profiles as training data and 1,000 profiles as testing data.

Though positive and negative samples of each skill are imbalanced (In this paper, the number of the negative samples is much larger than that of the positive samples), we select balanced testing and training samples for each skill. Following models are implemented and compared.

- *Keyword*, for each profile, we consider the profile attached with the skill, only if the text of the skill appears on the profile article with textual information.

- *MaxEnt*, which first uses local textual information as features to train a maximum entropy (ME) classification model, and then employs the classification model to predict the skills in the testing data set. The ME algorithm is implemented with the mallet toolkit [1].

- *JPFG*, exactly our proposed model, which jointly predicts personal skills with local textual information, personal connection and skill connection.

For performance evaluation, we adopt Precision (P.), Recall (R.) and F1-Measure (F1.).

### 6.2 Comparison with Baselines

Our first group of experiments is to investigate whether the JPFG model is able to improve skill inference and whether the personal and skill connections are useful. The experimental results are shown in Table 3. From the table we can find that as some skills may not be mentioned on the Summary and Experience fields directly, the performance of the Keyword approach is far from satisfaction. As incorporating personal and skill connections, the JPFG model yields a much higher F1-measure, which improves the performance with about 6.8% gain than the MaxEnt model.

### 6.3 Performance of JPFG with Different Training Data Sizes

After we evaluate the effective of the JPFG model with the large-scale training data, we carry out experiments to test the effect of the JPFG model with different training data sizes. Experiment results are shown in Figure 3. It shows that the JPFG model with both personal and skill connections always outperform the two baseline models. Impressively, our JPFG model using 20% training data outperforms MaxEnt using 100% training data.

[1] http://mallet.cs.umass.edu/

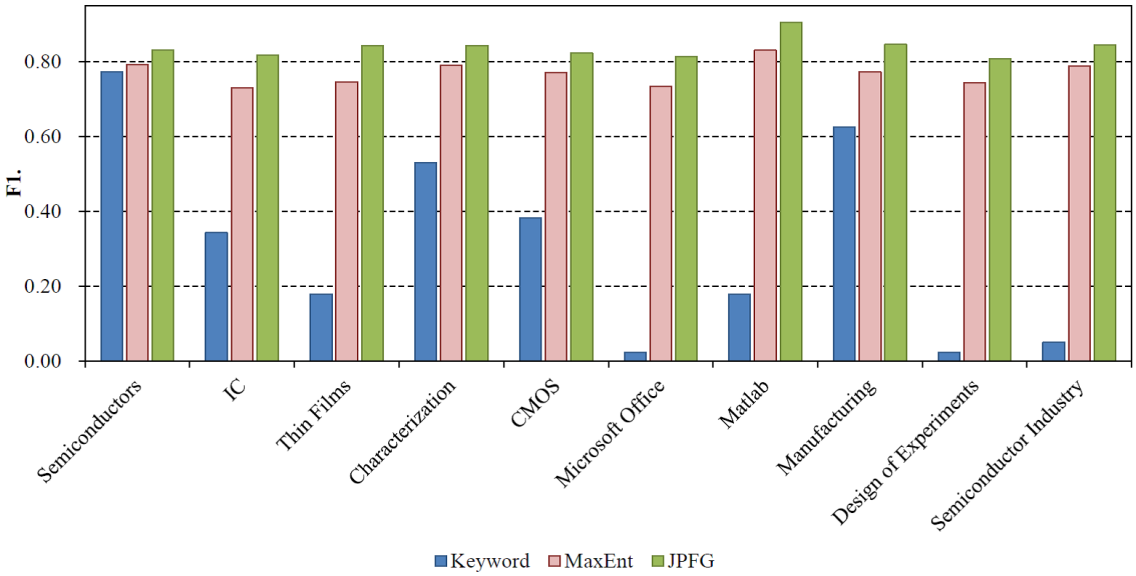



Figure 2: The performance of different methods for skill inference

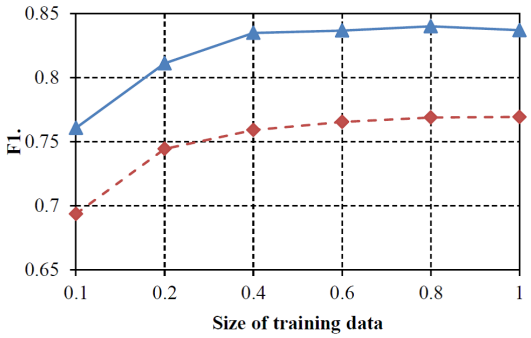



Figure 3: The performance of JPFG with different training data sizes

### 6.4 Connections Contribution Analysis

Personal connections and skill connections can be also used to build the factor graph models to infer the skills. We therefore want to compare our JPFG model with the factor graph model with only consider the personal connections or skill connections, and analysis the contribution of each kinds of connection. Specifically, MaxEnt-Personal employs the personal connections as additional features incorporated with textual features to build the maximum entropy classification. FGM-Personal is a simplified version of the JPFG model, which only employs textual attribute functions and personal connection factor functions to build the factor graph model. Likewise, FGM-Skill only employs textual attribute functions and skill connection factor functions to build the factor graph model. Table 3 shows the experiment results.

| System | P. | R. | F1. |
|---|---|---|---|
| MaxEnt | 0.744 | 0.797 | 0.769 |
| MaxEnt-Personal | 0.758 | 0.812 | 0.783 |
| FGM-Personal | 0.765 | 0.817 | 0.790 |
| FGM-Skill | 0.704 | **0.967** | 0.815 |
| JPFG | **0.780** | 0.905 | **0.837** |

Table 3: The contribution of connections

From Table 3, we can observe that, 1) Both FGM-Personal and FGM-Skill outperform the baseline

MaxEnt approach. It shows that both personal connections and skill connections are helpful for skill inference; 2) MaxEnt-Personal and FGM-Personal outperform the baseline MaxEnt approach, it show that personal connections are helpful for inferring skills, and as considering the global optimization, FGM-Personal is more effective; 3) FGM-Skill built on the skill connections is more effective than MaxEnt-Personal and FGM-Personal, it show that skill connections are more useful than personal connections; 4) JPFG model outperforms both FGM-Personal and FGM-Skill, it suggests that we should incorporate both personal and skill connections to the factor graph model when we infer the skills from profile.

## 7 Conclusion

In this study, we propose a novel task named personal skill inference, which aims to determine whether a person takes a specific skill or not. To address this task, we propose a joint prediction factor graph model with help of both personal and skill connections besides local textual information. Evaluation on a large-scale dataset shows that our joint model performs much better than several baselines. In particular, it shows that the performance on personal skill inference can be greatly improved by incorporating skill connection information.

The general idea of exploring personal and skill connections to help predict people's skills represents an interesting research direction in social networking, which has many potential applications. Besides, as skill information of a person is normally incomplete and fuzzy, how to better infer personal skills with weakly labeled information is challenging.

## Acknowledgements

This research work is supported by the National Natural Science Foundation of China (No. 61273320, No. 61331011, and No. 61375073), National High-tech Research and Development Program of China (No. 2012AA011102), Zhejiang Provincial Natural Science Foundation of China (No. LY13F020007), the Humanity and Social Science on Young Fund of the Ministry of Education (No. 12YJC630170).

We thank Dr. Jie Tang and Honglei Zhuang for providing their software and useful suggestions about PGM. We thank Prof. Deyi Xiong for helpful discussions, and we acknowledge Dr. Xinfang Liu, and Yunxia Xue for corpus construction and insightful comments. We also thank anonymous reviewers for their valuable suggestions and comments.

## References

Balog K and M. Rijke. 2007. Determining Expert Profiles (With an Application to Expert Finding). In *Proceedings of IJCAI-07*.

Campbell C, P. Maglio, A. Cozzi, and B. Dom. 2003. Expertise Identification Using Email Communications. In *Proceedings of CIKM-03*.

Dong Y., J. Tang, S. Wu, J. Tian, N. Chawla, J. Rao, and H. Cao. 2012. Link Prediction and Recommendation across Heterogeneous Social Networks. In *Proceedings of ICDM-12*.

Guo W., H. Li, H. Ji, and M. Diab. 2013. Linking Tweets to News: A Framework to Enrich Short Text Data in Social Media. In *Proceedings of ACL-13* .

Guy I., N. Zwerdling, I. Ronen, D. Carmel, and E. Uziel. 2010. Social Media Recommendation based on People and Tags. In *Proceedings of SIGIR-10* .

Hammersley J. and P. Clifford. 1971. Markov Field on Finite Graphs and Lattices, *Unpublished manuscript*.

Helic D. and M. Strohmaier. 2011. Building Directories for Social Tagging Systems. In *Proceedings of CIKM-2011*.

Lafferty J, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML-01*.

Lampos V., D. Preo?iuc-Pietro, and T. Cohn. 2013. A User-centric Model of Voting Intention from Social Media. In *Proceedings of ACL-13*.

Lappas T., K. Punera, and T. Sarlos. 2011. Mining Tags Using Social Endorsement Networks. In *Proceedings of SIGIR-11*.

Li H., Z. Liu, and M. Sun. 2012. Random Walks on Context-Aware Relation Graphs for Ranking Social Tags. In *Proceedings of COLING-12*.

Liu Z., X. Chen, and M. Sun. 2011. A Simple Word Trigger Method for Social Tag Suggestion. In *Proceedings of EMNLP-2011*.

Liu Z., C. Tu, and M. Sun. 2012. Tag Dispatch Model with Social Network Regularization for Microblog User Tag Suggestion. In *Proceedings of COLING-12*.

Lu Y., and P. Tsaparas, A. 2010. Ntoulas and L. Polanyi. 2010. Exploiting Social Context for Review Quality Prediction. In *Proceedings of WWW-10*.

Luo T., J. Tang, J. Hopcroft, Z. Fang, and X. Ding. 2013. Learning to Predict Reciprocity and Triadic Closure in Social Networks. *ACM Transactions on Knowledge Discovery from Data*. vol.7(2), Article No. 5.

Murphy K., Y. Weiss, and M. Jordan. 1999. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In *Proceedings of UAI-99* .

Ohkura T., Y. Kiyota and H. Nakagawa. 2006. Browsing System for Weblog Articles based on Automated Folksonomy. In *Proceedings of WWW-06*.

Si X., Z. Liu, and M. Sun. 2010. Explore the Structure of Social Tags by Subsumption Relations. In *Proceedings of COLING-10*.

Soboroff I., A. Vries and N. Craswell. 2006. Overview of the TREC 2006 Enterprise Track In *Proceedings of TREC-06*.

Turney P. 2002. Thumbs up or Thumbs down? Semantic Orientation Applied to Unsupervised Classification of reviews. In *Proceedings of ACL-02*.

Tan C., L. Lee, J. Tang, L. Jiang, M. Zhou, and P. Li. 2011. User-Level Sentiment Analysis Incorporating Social Networks. In *Proceedings of KDD-11*.

Tang W., H. Zhuang, and J. Tang. 2011a. Learning to Infer Social Ties in Large Networks. In *Proceedings of ECML/PKDD-11*.

Tang J., Y. Zhang, J. Sun, J. Rao, W. Yu, Y. Chen, and A. Fong. 2011b. Quantitative Study of Individual Emotional States in Social Networks. *IEEE Transactions on Affective Computing*. vol.3(2), Pages 132-144.

Tang J., S. Wu, J. Sun, and H. Su. 2012. Cross-domain Collaboration Recommendation. In *Proceedings of KDD-12*.

Tang J., S. Wu, and J. Sun. 2013. Confluence: Conformity Influence in Large Social Networks. In *Proceedings of KDD-13*.

Xing E, M. Jordan, and S. Russell. 2003. A Generalized Mean Field Algorithm for Variational Inference in Exponential Families. In *Proceedings of UAI-03*.

Yang S., B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. 2011a. Like like alike - Joint Friendship and Interest Propagation in Social Networks. In *Proceedings of WWW-11*.

Yang Z., K. Cai, J. Tang, L. Zhang, Z. Su, and J. Li. 2011b. Social Context Summarization. In *Proceedings of SIGIR-11*.

Zhang J., J. Tang, and J. Li. 2007a. Expert Finding in A Social Network. In *Proceedings of the Twelfth Database Systems for Advanced Applications (DASFAA-2007)*.

Zhang J., M. Ackerman, and L. Adamic. 2007b. Expertise Networks in Online Communities: Structure and Algorithms. In *Proceedings of TREC-07*.

Zhuang H, J. Tang, W. Tang, T. Lou, A. Chin, and X. Wang. 2012. Actively Learning to Infer Social Ties. In *Proceedings of Data Mining and Knowledge Discovery (DMKD-12)*, vol.25 (2), pages 270-297.

# Bilingual Event Extraction: a Case Study on Trigger Type Determination

**Zhu Zhu**[†] **Shoushan Li**[†*] **Guodong Zhou**[†] **Rui Xia**[‡]

[†]Natural Language Processing Lab
Soochow University, China
{zhuzhu0020, shoushan.li}@gmail.com, gdzhou@suda.edu.cn

[‡]Department of Computer Science
Nanjing University of Science and Technology
rxia@njust.edu.cn

## Abstract

Event extraction generally suffers from the data sparseness problem. In this paper, we address this problem by utilizing the labeled data from two different languages. As a preliminary study, we mainly focus on the subtask of trigger type determination in event extraction. To make the training data in different languages help each other, we propose a uniform text representation with bilingual features to represent the samples and handle the difficulty of locating the triggers in the translated text from both monolingual and bilingual perspectives. Empirical studies demonstrate the effectiveness of the proposed approach to bilingual classification on trigger type determination.

## 1 Introduction

Event extraction is an increasingly hot and challenging research topic in the natural language processing (NLP) community (Ahn, 2006; Saun et al. 2006; Zhao et al. 2008). It aims to automatically extract certain types of events with the arguments to present the texts under a structured form. In event extraction, there are four primary subtasks, named trigger identification, trigger type determination, argument identification, and argument role determination (Chen and NG, 2012). As an important technology in information extraction, event extraction could be applied to many fields such as information retrieval, summarization, text mining, and question answering.

Recently, the dominative approach to event extraction is based on supervised learning where a set of labeled samples are exploited to train a model to extract the events. However, the available labeled data are rather sparse due to various kinds of event categories. For example, the event taxonomy in ACE 2005 [1] (Automatic Content Extraction) includes 8 types of events, with 33 subtypes, such as "*Marry/Life*" (subtype/type), and "*Transport/Movement*". Moreover, some subtypes such as "*Nominate/Personnel*" and "*Convict/Justice*" contain less than 10 labeled samples in the English and Chinese corpus respectively. Apparently, such a small scale of training data is difficult to yield a satisfying performance.

One possible way to alleviate the data sparseness problem in event extraction is to conduct bilingual event extraction with training data from two different languages. This is motivated by the fact that labeled data from a language is highly possible to convey similar information in another language. For example, **E1** is an event sample from the English corpus and **E2** is another one in the Chinese corpus. Apparently, **E1** and the English translation text of **E2**, share some important clues such as *meet* and *Iraq* which highly indicates the event type of "*Meet/Contact*".

**E1**: *Bush arrived in Saint Petersburg on Saturday, when he also briefly met German chancellor Gerhard Schroeder, whose opposition to the Iraq war had soured his relationship with Washington, at a dinner hosted by Putin.*

**E2**: *美国总统布什将于2月访问德国并与施罗德会谈，伊朗和伊拉克问题将是双方会谈的重点。(U.S. president George W. Bush will visit Germany in February and meet with Schroeder, Iran and Iraq will be the focus of the talks the two sides.)*

In this paper, we address the data sparseness problem in event extraction with a bilingual pro-

* Corresponding author

[1] http://www.nist.gov/speech/tests/ace/2005



*Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 842–847, Baltimore, Maryland, USA, June 23-25 2014. 

cessing approach which aims to exploit bilingual training data to enhance the extraction performance in each language. As a preliminary work, we mainly focus on the subtask of trigger type determination. Accordingly, our goal is to design a classifier which is trained with labeled data from two different languages and is capable of classifying the test data from both languages. Generally, this task possesses two main challenges.

The first challenge is text representation, namely, how to eliminate the language gap between the two languages. To tackle this, we first employ Google Translate[2], a state-of-the-art machine translation system, to gain the translation of an event instance, similar to what has been widely done by previous studies in bilingual classification tasks e.g., Wan (2008); Then, we uniformly represent each text with bilingual word features. That is, we augment each original feature vector into a novel one which contains the translated features.

The second challenge is the translation for some specific features. It is well-known that some specific features, such as the triggers and their context features, are extremely important for determining the event types. For example, in **E3**, both trigger "***left***" and named entity "***Saddam***" are important features to tell the event type, i.e., "*Transport/Movement*". When it is translated to Chinese, it is also required to know trigger "*离开*"(***left***) and named entity "*萨达姆*" (***Saddam***) in **E4,** the Chinese translation of **E3.**

**E3**: ***Saddam****'s clan is said to have* ***left*** *for a small village in the desert.*

**E4**: Chinese translation: *据说* ***萨达姆(Saddam)*** *家族已经* ***离开(left)*** *沙漠中的一个小村庄。*

However, it is normally difficult to know which words are the triggers and surrounding entities in the translated sentence. To tackle this issue, we propose to locate the trigger from both monolingual and bilingual perspectives in the translation text. Empirical studies demonstrate that adding the translation of these specific features substantially improves the classification performance.

The remainder of this paper is organized as follows. Section 2 overviews the related work on event extraction. Section 3 proposes our approach to bilingual event extraction. Section 4 gives the experimental studies. In Section 5, we conclude our work and give some future work.

## 2 Related Work

In the NLP community, event extraction has been mainly studied in both English and Chinese.

In English, various supervised learning approaches have been explored recently. Bethard and Martin (2006) formulate the event identification as a classification problem in a word-chunking paradigm, introducing a variety of linguistically motivated features. Ahn (2006) proposes a trigger-based method. It first identifies the trigger in an event, and then uses a multi-classifier to implement trigger type determination. Ji and Grishman (2008) employ an approach to propagate consistent event arguments across sentences and documents. Liao and Grishman (2010) apply document level information to improve the performance of event extraction. Hong et al. (2011) leverage cross-entity information to improve traditional event extraction, regarding entity type consistency as a key feature. More recently, Li et al. (2013) propose a joint framework based on structured prediction which extracts triggers and arguments together.

In Chinese, relevant studies in event extraction are in a relatively primary stage with focus on more special characteristics and challenges. Tan et al. (2008) employ local feature selection and explicit discrimination of positive and negative features to ensure the performance of trigger type determination. Chen and Ji (2009) apply lexical, syntactic and semantic features in trigger labeling and argument labeling to improve the performance. More recently, Li et al. (2012) and Li et al. (2013) introduce two inference mechanisms to infer unknown triggers and recover trigger mentions respectively with morphological structures.

In comparison with above studies, we focus on bilingual event extraction. Although bilingual classification has been paid lots of attention in other fields (Wan 2008; Haghighi et al., 2008; Ismail et al., 2010; Lu et al., 2011； Li et al., 2013), there is few related work in event extraction. The only one related work we find is Ji (2009) which proposes an inductive learning approach to exploit cross-lingual predicate clusters to improve the event extraction task with the main goal to get the event taggers from extra resources, i.e., an English and Chinese parallel corpus. Differently, our goal is to make the la-

[2] www.google.com

beled data from two languages help each other without any other extra resources, which is original in the study of event extraction.

## 3 The Proposed Approach

Trigger type determination aims to determine the event type of a trigger given the trigger and its context (e.g., a sentence). Existing approaches to trigger type determination mainly focus on monolingual classification. Figure 1 illustrates the framework for Chinese and English.

In comparison, our approach exploits the corpora from two different languages. Figure 2 illustrates the framework. As shown in the figure, we first get the translated corpora of Chinese and English origin corpora through machine translation. Then, we represent each text with bilingual features, which enables us to merge the training data from both languages so as to make them help each other.

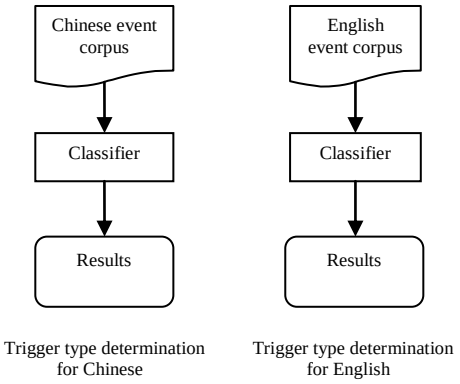



Figure 1: The framework of monolingual classification for trigger type determination

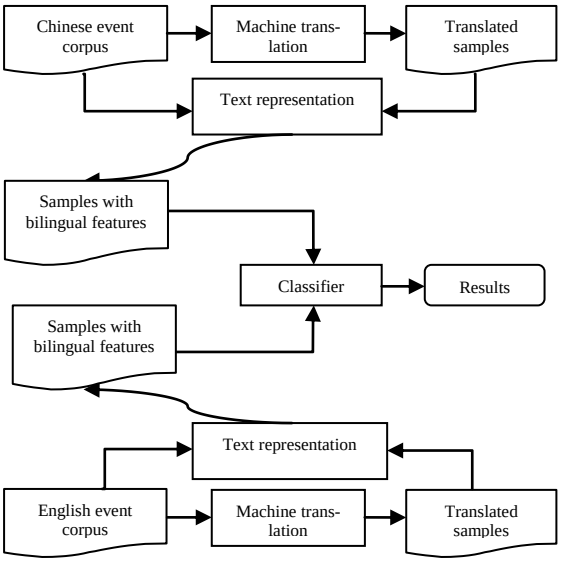



Figure 2: The framework of bilingual classification for trigger type determination

### 3.1 Text Representation

In a supervised learning approach, labeled data is trained to obtain a classifier. In this approach, the extracted features are the key components to make a successful classifier. Table 1 shows some typical kinds of features in a monolingual classification task for trigger type determination. To better understand these features, the real feature examples in **E3** are given in the table.

Given the feature definition, a monolingual sample $x$ is represented as the combination of all the features, i.e.,

$$x = \begin{pmatrix} e_1, e_2 \ldots, e_n, Tri, POS\_Tri, Tri\_con, \\ POS\_con, Ent, Ent\_type, Ent\_subtype \end{pmatrix} \quad (1)$$

| Features | Feature examples in **E3** |
|---|---|
| All words ($e_1, e_2 \ldots, e_n$) | *Saddam, clan, is, ... , desert* |
| Trigger (*Tri*) | *left* |
| POS of the trigger (*POS_Tri*) | *VBN* |
| Trigger's context words (*Tri_con*) | *...,have, for,...* |
| POS of trigger's context words (*POS_con*) | *...,VB,IN,...* |
| Entities around trigger (*Ent*) | *Saddam* |
| Entity type (*Ent_type*) | *PER* |
| Entity subtype (*Ent_subtype*) | *individual* |

Table 1: The features and some feature examples for trigger type determination

In bilingual classification, we represent a sample with bilingual features, which makes it possible to train with the data from two languages. To achieve this goal, we employ a single feature augmentation strategy to augment the monolingual features into bilingual features, i.e.,

$$x \Rightarrow x_{Chinese,}\, x_{English} \quad (2)$$

Specifically, a sample $x$ is represented as follows:

$$x = \begin{pmatrix} \begin{pmatrix} c_1, c_2 \ldots, c_m, Tri_c, POS\_Tri_c, Tri_c\_con, \\ POS\_con, Ent_c, Ent_c\_type, Ent_c\_subtype \end{pmatrix} \\ \begin{pmatrix} e_1, e_2, \ldots, e_n, Tri_e, POS\_Tri_e, Tri_e\_con, \\ POS\_con, Ent_e, Ent_e\_type, Ent_e\_subtype \end{pmatrix} \end{pmatrix} \quad (3)$$

Where the tokens with the '*c*'/'*e*' subscript mean the features generated from the Chinese/English text. From the features, we can see that some

features, such as *Tri_con* and *Ent*, depend on the location of the trigger word. Therefore, locating the trigger in the translated text becomes crucial.

### 3.2 Locating Translated Trigger

Without loss of generality, we consider the case of translating a Chinese event sample into an English one. Formally, the word sequence of a Chinese event sample is denoted as $s_c = (c_1, c_2, \ldots, c_n)$, while the sequence of the translated one is denoted as $s_e = (e_1, e_2 \ldots, e_m)$. Then, the objective is to get the English trigger $Tri_e$ in $s_e$, given the Chinese trigger word $Tri_c$ in $s_c$. The objective function is given as follows:

$$\arg\max_{1 \le k,l \le m} P\left(e_{k\_l} = Tri_e\right) \quad (4)$$

Where $e_{k\_l}$ denotes the substring $(e_k, e_{k+1} \ldots, e_l)$ in $s_e$ and $1 \le k,l \le m$.

In this paper, the above function could be solved in two perspectives: monolingual and bilingual ones. The former uses the English training data alone to locate the trigger while the latter exploit the bilingual information to get the translated counterpart of the Chinese trigger.

**The monolingual perspective**: The objective is to locate the trigger with the monolingual information. That is,

$$\arg\max_{1 \le k,l \le m} P\left(e_{k\_l} = Tri_e \mid s_e, R_e\right) \quad (5)$$

Where $R_e$ denotes the training resource in English. In fact, this task is exactly the first subtask in event extraction named trigger identification, as mentioned in Introduction. For a simplified implementation, we first estimate the probabilities of $P\left(e_{k\_l} = Tri_e\right)$ in $R_e$ with maximum likelihood estimation when $e_{k\_l} \in s_e$.

**The bilingual perspective:** The objective is to locate the trigger with the bilingual information. That is,

$$\arg\max_{1 \le k,l \le m} P\left(e_{k\_l} = Tri_e \mid s_e, s_c, Tri_c\right) \quad (6)$$

Where $Tri_c$ is the trigger word in Chinese and $s_e$ is the translated text towards $s_c$. More generally, this can be solved from a standard word alignment model in machine translation (Och et al, 1999; Koehn et al, 2003). However, training a word alignment requires a huge parallel corpus which is not available here.

For a simplified implementation, we first get the $Tri_c$ 's translation, denoted as $trans_{Tri_c}$, with Google Translate. Then, we estimate $P\left(e_{k\_l} = Tri_e\right)$ as follows:

$$P\left(e_{k\_l} = Tri_e\right) = \begin{cases} 0.9 & if\ e_{k\_l} = trans_{Tri_c} \\ \alpha & others \end{cases} \quad (7)$$

Where 0.9 is an empirical value which makes the translation probability become a dominative factor when the translation of the trigger is found in the translated sentence. $\alpha$ is a small value which makes the sum of all probabilities equals 1.

The final decision is made according to both the monolingual and bilingual perspectives, i.e.,

$$\arg\max_{1 \le k,l \le m} P\left(e_{k\_l} = Tri_e \mid s_e, R_e\right) \cdot P\left(e_{k\_l} = Tri_e \mid s_e, s_c, Tri_c\right) \quad (8)$$

Note that we reduce the computational cost by make the word length of the trigger less than 3, i.e., $l - k \le 3$.

## 4 Experimentation

### 4.1 Experimental Setting

**Data sets**: The Chinese and English corpus for even extraction are from ACE2005, which involves 8 types and 33 subtypes. All our experiments are conducted on the subtype case. Due to the space limit, we only report the statistics for each type, as shown in Table 2. For each subtype, 80% samples are used as training data while the rest are as test data.

| # | Chinese | English | *total* |
|---|---|---|---|
| Life | 389 | 902 | *1291* |
| Movement | 593 | 679 | *1272* |
| Transaction | 147 | 379 | *526* |
| Business | 144 | 137 | *281* |
| Conflict | 514 | 1629 | *2143* |
| Contact | 263 | 373 | *636* |
| Personnel | 203 | 514 | *717* |
| Justice | 457 | 672 | *1129* |
| *total* | *2710* | *5285* | *7995* |

Table 2: Statistics in each event type in both Chinese and English data sets

**Features**: The features have been illustrated in Table 1 in Section 3.2.

**Classification algorithm**: The maximum entropy (ME) classifier is implemented with the public tool, Mallet Toolkits[3].

**Evaluation metric**: The performance of event type recognition is evaluated with F-score.

### 4.2 Experimental Results

In this section, we evaluate the performance of our approach to bilingual classification on trigger type determination. For comparison, following approaches are implemented:

- **Monolingual**: perform monolingual classification on the Chinese and English corpus individually, as shown in Figure 1.
- **Bilingual**: perform bilingual classification with partial bilingual features, ignoring the context features (e.g., context words, context entities) under the assumption that the trigger location task is not done.
- **Bilingual_location**: perform bilingual classification by translating each sample into another language and using a uniform representation with all bilingual features as shown in Section 3.2. This is exactly our approach. The number of the context words and entities before or after the trigger words is set as 3.

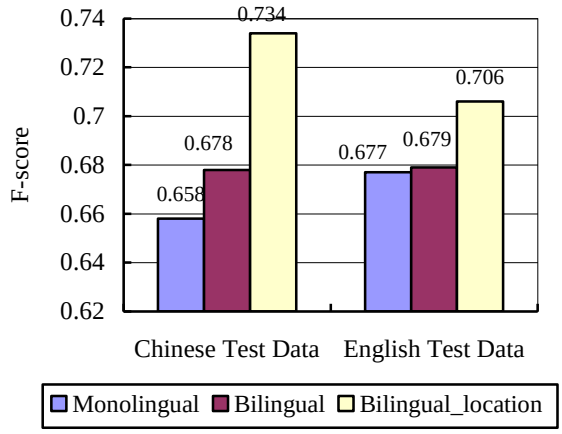



Figure 3: Performance comparison of the three approaches on the Chinese and English test data

Figure 3 shows the classification results of the three approaches on the Chinese and English test data. From this figure, we can see that **Bilingual_location** apparently outperform **Monolingual**, which verifies the effectiveness of using bilingual corpus. Specifically, the improvement by our approach in Chinese is impressive, reaching 7.6%. The results also demonstrate the importance of the operation of the trigger location, without which, bilingual classification can only slightly improve the performance, as shown in the English test data.

The results demonstrate that our bilingual classification approaches are more effective for the Chinese data. This is understandable because the size of English data is much larger than that of Chinese data, 5285 vs. 2710, as shown in Table 2. Specifically, after checking the results in each subtype, we find that some subtypes in Chinese have very few samples while corresponding subtypes in English have a certain number samples. For example, the subtype of "*Elect/Personnel*" only contains 30 samples in the Chinese data while 161 samples can be found in the English data, which leads a very high improvement (15.4%) for the Chinese test data. In summary, our bilingual classification approach provides an effective way to handle the data sparseness problem in even extraction.

## 5 Conclusion and Future Work

This paper addresses the data sparseness problem in event extraction by proposing a bilingual classification approach. In this approach, we use a uniform text representation with bilingual features and merge the training samples from both languages to enlarge the size of the labeled data. Furthermore, we handle the difficulty of locating the trigger from both the monolingual and bilingual perspectives. Empirical studies show that our approach is effective in using bilingual corpus to improve monolingual classification in trigger type determination.

Bilingual event extraction is still in its early stage and many related research issues need to be investigated in the future work. For example, it is required to propose novel approaches to the bilingual processing tasks in other subtasks of event extraction. Moreover, it is rather challenging to consider a whole bilingual processing framework when all these subtasks are involved together.

## Acknowledgments

This research work has been partially supported by two NSFC grants, No.61375073, and No.61273320, one National High-tech Research and Development Program of China No.2012AA011102, one General Research Fund (GRF) project No.543810 and one Early Career Scheme (ECS) project No.559313 sponsored by the Research Grants Council of Hong Kong, the NSF grant of Zhejiang Province No.Z1110551.

[3] http://mallet.cs.umass.edu/

## References


Ahn D. 2006. The Stages of Event Extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pp.1~8.

Bethard S. and J. Martin. 2006. Identification of Event Mentions and Their Semantic Class. In *Proceedings of EMNLP-2006*, pp.146-154.

Chen C. and V. NG. 2012. Joint Modeling for Chinese Event Extraction with Rich Linguistic Features. In *Proceedings of COLING-2012*, pp. 529-544.

Chen Z. and H. Ji. 2009. Language Specific Issue and Feature Exploration in Chinese Event Extraction. In *Proceedings of NAACL-2009*, pp. 209-212.

Haghighi A., P. Liang, T. Berg-Kirkpatrick and D. Klein. 2008. Learning Bilingual Lexicons from Monolingual Corpora. In *Proceedings of ACL-2008*, pp. 771-779.

Hong Y., J. Zhang., B. Ma., J. Yao., and G. Zhou. 2011. Using Cross-Entity Inference to Improve Event Extraction. In *Proceedings of ACL-2011*, pp. 1127–1136.

Ismail A., and S. Manandhar. 2010. Bilingual Lexicon Extraction from Comparable Corpora Using In-domain Terms. In *Proceedings of COLING-2010*, pp.481-489.

Ji H. 2009. Cross-lingual Predicate Cluster Acquisition to Improve Bilingual Event Extraction by Inductive Learning. In *Proceedings of the Workshop on Unsupervised and Minimally Supervised Learning of Lexical Semantics*, pp. 27-35.

Ji H, and R. Grishman. 2008. Refining Event Extraction through Cross-Document Inference. In *Proceedings of ACL-2008*, pp. 254-262.

Koehn P., F. Och, and D. Marcu. 2003. Statistical Phrase-based Translation. In *Proceedings of HTL-NAACL-2003*, pp. 127-133.

Li P., and G. Zhou. 2012. Employing Morphological Structures and Sememes for Chinese Event Extraction. In *Proceedings of COLING-2012*, pp. 1619-1634.

Li P., Q. Zhu and G. Zhou. 2013. Using Compositional Semantics and Discourse Consistency to Improve Chinese Trigger Identification. In *Proceedings of COLING-2013*, pp. 399-415.

Li Q, H Ji, and H. Liang. 2013. Joint Event Extraction via Structured Prediction with Global Features. In *Proceedings of ACL-2013*, pp. 73-82.

Li S, R Wang, H Liu, and CR Huang. 2013. Active Learning for Cross-Lingual Sentiment Classification. In *Proceedings of Natural Language Processing and Chinese Computing*, pp. 236-246.

Liao S and R. Grishman. 2010. Using Document Level Cross-event Inference to Improve Event Extraction. In *Proceedings of ACL-2010*, pp. 789-797.

Lu B., C. Tan, C. Cardie and B. K. Tsou. 2011. Joint Bilingual Sentiment Classification with Unlabeled Parallel Corpora. In *Proceedings of ACL-2011*, pp. 320-330.

Och F., C. Tillmann, and H. Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In *Proceedings of EMNLP-1999*, pp.20-28.

Tan H., T. Zhao, and J. Zheng. 2008. Identification of Chinese Event and Their Argument Roles. In *Proceedings of CITWORKSHOPS-2008*, pp. 14-19.

Wan X. 2008. Using Bilingual Knowledge and Ensemble Techniques for Unsupervised Chinese Sentiment Analysis. In *Proceedings of EMNLP-2008*, pp. 553-561.

Zhao Y., Y. Wang, B. Qin, et al. 2008. Research on Chinese Event Extraction. In *Proceedings of Journal of Chinese Information, 22(01)*, pp. 3-8.

# Ranking Based Activity Trajectory Search

Wei Chen[1], Lei Zhao[1,2], Xu Jiajie[1,2], Kai Zheng[3], and Xiaofang Zhou[1,3]

[1] School of Computer Science and Technology, Soochow University, China
[2] Jiangsu Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, China
[3] School of ITEE, The University of Queensland, Brisbane, Australia
wchzhg@gmail.com, {zhaol,xujj}@suda.edu.cn, {kevinz,zxf}@itee.uq.edu.au

**Abstract.** With the proliferation of the GPS-enabled devices and mobile techniques, there has been a lot of work on trajectory search in the last decade. Previous trajectory search has focused on spatio-temporal features and text descriptions. Different from them, we study a novel problem of searching trajectories with activities and corresponding ranking information. Given a query $q$, which is attached with a set of activities and a threshold of distance, the results of ranking based activity trajectory search (RTS) are $k$ trajectories such that the given activities are performed with the highest ranking within the threshold of distance. In addition, we also extend the query with an order, i.e., order-sensitive ranking based activity trajectory search (ORTS), which takes both the order of activities in a query $q$ and the order of trajectories into account. It is challenging to answer RTS and ORTS efficiently due to the structural complexity of trajectory data with ranking information. In this paper, a hybrid index AC-tree and its optimized variant RAC-tree are proposed to achieve higher efficiency. Extensive experiments verify the high efficiency and scalability of the proposed algorithms.

**Keywords:** Trajectory Search, Ranking, Activity Trajectory.

## 1 Introduction

As the ubiquitousness of devices with GPS and the rapid development of wireless sensor technology, more and more people log their locations nowadays. In addition, the share of trajectories become available on more and more web sites, such as Twitter, Four-square, Facebook and Bikely, which leads to an incredible increasing of trajectory number. Trajectory search has become a popular concern for industry and academic community in the last decade. Besides, many trajectories based applications are also becoming increasingly popular, such as trip planning and trajectory recommendation.

In recent work on trajectory search, a trajectory is usually modeled as a sequence of geo-locations with keywords information. These work focuses on minimizing the distance while all keywords of a query are covered. However, this is not always reasonable. Assuming that a user wants to do some shopping and

B. Benatallah et al. (Eds.): WISE 2014, Part I, LNCS 8786, pp. 170–185, 2014.

have a haircut after work, conventional query will return $k$ trajectories in which the user can fulfill his plan. However the barbershops and shopping malls in these trajectories may have a bad reputation. Having observed the weakness of previous studies, we propose a new query of searching activity trajectories with ranking information.

Consider the example demonstrated in Fig. 1. A tourist wants to conduct activities $(a, c, d)$ on her/his trip and the total distance of trip cannot exceed 20. If the ranking information is beyond consideration, $\tau_1$ will be the best result since the trajectory $(q, p_{1,2}, p_{1,3}, p_{1,4})$ has the minimum distance. However, $\tau_2$ is a better choice in a sense because the rankings of activities $(a, c, d)$ in $\tau_2$ are higher than that in $\tau_1$.

Ranking Information

| *trajectory* | *poi* | *activity* | *ranking* |
|---|---|---|---|
| $\tau_1$ | $p_{1,1}$ | $d$ | $d.r:9$ |
| | $p_{1,2}$ | $a$ | $a.r:7$ |
| | $p_{1,3}$ | $c$ | $c.r:6$ |
| | $p_{1,4}$ | $d$ | $d.r:4$ |
| $\tau_2$ | $p_{2,1}$ | $d$ | $d.r:7$ |
| | $p_{2,2}$ | $c$ | $c.r:8$ |
| | $p_{2,3}$ | $a$ | $a.r:8$ |
| | $p_{2,4}$ | $d$ | $d.r:9$ |

Euclidean Distance

| | $p_{1,1}$ | $p_{1,2}$ | $p_{1,3}$ | $p_{1,4}$ | $p_{2,1}$ | $p_{2,2}$ | $p_{2,3}$ | $p_{2,4}$ |
|---|---|---|---|---|---|---|---|---|
| $q$ | 2 | 2 | 9 | 12 | 4 | 4 | 11 | 14 |

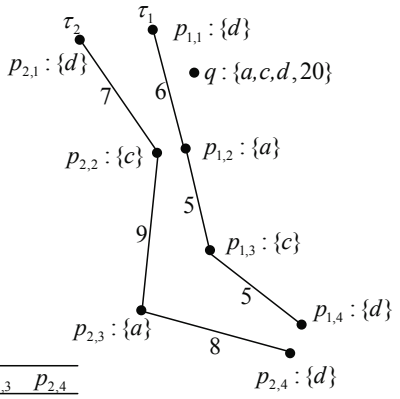



**Fig. 1.** An Example of Trajectory Matching

The query proposed in this paper is more challenging compared with the previous work. On one hand, indexing trajectory data becomes more difficult since extra ranking information should be taken into account. On the other hand, computing is becoming increasingly complex since more factors should be taken into consideration. Obviously, existing methods cannot be applied to this work directly.

In order to tackle the problem efficiently, we propose new indexes and algorithms in this paper. Firstly, a new hybrid index called AC-tree is proposed. We prune the search space fast and obtain a candidate set by retrieving AC-tree. Secondly, new algorithms are proposed to compute the maximum ranking within a threshold of distance for each candidate trajectory. Thirdly, optimized index and algorithms are proposed to make computing faster. To sum up, the main contribution of this work is to take ranking information into account while recommending trajectories. It makes the results more reasonable.

The rest of this paper is organized as follows. In Section 2, we briefly view existing work related to trajectory search. Section 3 presents the problem statement

and necessary notations in this work. We introduce the index tree in Section 4 and algorithms in Section 5, which is followed by the optimization of index and algorithms in Section 6. In Section 7 we report the experimental results. This paper is concluded in Section 8.

## 2 Related Work

In the past decade, trajectory search has received significant attentions. Existing work [7,6,4,5] focus on searching trajectories in spatial-temporal domain without any other important features, such as, textual descriptions and activity information. [3] tackles the problem of searching trajectories by locations in spatial domain, in which multiple locations are used as the query and Euclidean distance is the only restriction. Chen et al. [5] investigate the problem of discovering popular routes from historical trajectories by given a location to a destination, in which a transfer network is established to derive the transfer probability for transfer nodes.

In existing work, [21,15,20,10,8] study the problem of predicting destinations based on trajectory data. [8] handle the problem of destination prediction by sub-trajectory synthesis, where historical trajectories are decomposed into sub-trajectories to avoid the data sparsity problem. A grid graph is proposed to represent the trajectories and a Bayesian inference framework is proposed to compute the probability in this work.

Meanwhile, with the prevalence of spatial web objects on the Internet, a lot of work appears in spatial keyword search. Existing work [14,16,18,19,1,13,22] address the problem of spatial keywords search in spatial-temporal and textual domain. Especially, [1,13] consider the fusion of keywords and trajectories. In [1], keywords are associated with the whole trajectory rather than each individual point. In addition, privacy protection is another important issue while searching trajectories [12,11,9,24,23,8] due to the increasing importance of protecting users from information leak.

[13] is the most similar work to our query, where each point in a trajectory is attached with keywords. Multiple locations with keywords are used as the query to search for trajectories that has the minimum match distance with respect to the query. A novel grid index is developed to organize trajectory data and new algorithms are also proposed to the tackle the problem efficiently. Despite the great contributions made by [13], it does not take the ranking information into account during processing. Hence, both the index structure and the algorithms of [13] are not suitable to our problem.

In order to address spatial keywords search efficiently, some hybrid index structures are proposed. In [17], a new index called $IR^2$-tree is proposed to prune search space in spatial and textual domain. Cong et al. [2] propose an IR-tree, which is an integration of R-tree and inverted files. By traversing IR-tree, the search space can be pruned fast with location and textual information. Zheng et al. [13] propose a novel Grid index called GAT, which utilizes both spatial information and query keywords to reduce search space.

In spite of the significant contributions of the aforementioned work, none of them take ranking information into consideration, which is an important new feature of trajectories. Meanwhile, the hybrid indexes and the corresponding algorithms are also not suitable to our search. As a result, we propose novel indexes and algorithms in this paper.

## 3 Problem Statement

In this section, we present all the definitions used throughout the paper. Before that, the notations used in this paper are summarized in Table 1.

**Table 1.** Definitions of notations

| Notation | Definition |
|---|---|
| $\alpha$ | Activity |
| $\alpha.r$ | Ranking of $\alpha$ and the value is in the range of [1-10] |
| $\tau$ | Trajectory |
| $\mathcal{D}$ | Set of $\tau$ |
| $\widehat{d}$ | The threshold of distance |
| $p.\varphi$ | The set of activities of POI $p$ |
| $\omega$ | Trajectory matching in the form of $(\tau, q)$ |
| $\omega^o$ | Order-sensitive trajectory matching |
| $r(\omega)$ | Ranking of a trajectory matching $\omega$ |
| $d(\omega)$ | Distance of a trajectory matching $\omega$ |

**Definition 1.** ***Trajectory.*** *Let $p = (x, y, \varphi, r)$ be a POI where $x$ is the longitude, $y$ is the latitude, $\varphi$ denotes the activities that can be performed in the location $(x, y)$, and $r$ is a set of ranking information of $\varphi$. A trajectory is a sequence of POIs, denoted as $\tau = (p_1, p_2, \ldots, p_n)$.*

**Definition 2.** ***Sub-trajectory.*** *Given a trajectory $\tau = (p_1, p_2, \ldots, p_n)$, a sub-trajectory of $\tau$ is $\tau_s^e = (p_s, p_{s+1}, \ldots, p_e)$, where $1 \leq s \leq e \leq n$, denoted as $\tau_s^e \subseteq \tau$.*

**Definition 3.** ***Trajectory matching.*** *Let $q = (x, y, \varphi)$ be a query, where $(x, y)$ is the start point of a trip and $\varphi$ is a set of activities needed to be performed. Given a trajectory $\tau$, a tuple $\omega = (\tau, q)$ is a trajectory matching if there exists a sub-trajectory $\tau_s^e \subseteq \tau$, such that $q.\varphi \subseteq \bigcup_{p_i \in \tau_s^e} p_i.\varphi$.*

**Definition 4.** ***Distance between trajectory and query.*** *Given a trajectory $\tau = (p_1, p_2, \ldots, p_n)$ and a query $q$, the distance between $\tau$ and $q$ is*

$$d(\tau, q) = dis(q, p_1) + \sum_{i=1}^{n-1} dis(p_i, p_{i+1}), \tag{1}$$

*where $dis$ is to get the Euclidean distance between two locations.*

**Definition 5.** ***Ranking of trajectory matching.*** *Given a trajectory matching* $\omega = (\tau, q)$ *and a threshold of distance* $\widehat{d}$*, suppose* $\tau$ *has* $n$ *sub-trajectories* $\tau_1$*,* $\tau_2$*, ...,* $\tau_n$*, such that each* $\omega_i = (\tau_i, q)(1 \leqslant i \leqslant n)$ *is a trajectory matching, then*

$$r(\omega, \widehat{d}) = \max_{d(\omega_i) \leqslant \widehat{d}} ( \sum_{\alpha \in q.\varphi} \max_{p_j \in \tau_i} (p_j.\alpha.r)), \tag{2}$$

*is the ranking of the trajectory matching* $\omega$*.*

In Fig. 1, $r((\tau_1, q), \widehat{d}) = r(((p_{1,1}, p_{1,2}, p_{1,3}), q), 20) = 22$. Although the sub-trajectory $(p_{1,2}, p_{1,3}, p_{1,4})$ also covers the activity set of $q$, its ranking is less than 22. Meanwhile, $r((\tau_2, q), \widehat{d}) = r(((p_{2,1}, p_{2,2}, p_{2,3}), q), 20) = 23$. Although the ranking of sub-trajectory $(p_{1,2}, p_{1,3}, p_{1,4})$ is higher, its distance is larger than 20.

Given a set of trajectories $\mathcal{D}$ and a query $q$, let $\mathcal{C}(\mathcal{C} \subseteq \mathcal{D})$ be a candidate which means for any trajectory $\tau \in \mathcal{C}$, it follows that $(\tau, q)$ is a trajectory matching. Given a positive integer $k$ and a threshold of distance $\widehat{d}$, the *Ranking Based Activity Trajectory Search( RTS )* returns a set $\mathcal{R}(\mathcal{R} \subseteq \mathcal{C}, |\mathcal{R}| = k)$, such that $\forall \tau \in \mathcal{R}$ and $\forall \tau' \in \mathcal{C} - \mathcal{R}$ it follows that $r((\tau, q), \widehat{d}) \geq r((\tau', q), \widehat{d})$.

## 4 AC-Tree

Due to the uneven distribution of POIs in practice, the search regions are partitioned into different grid cells. If the number of POIs attached to a cell exceeds a threshold $\theta$, the cell should be partitioned into four components. An AC-tree is proposed to organize all cells. Each cell corresponds to a leaf node of the AC-tree.

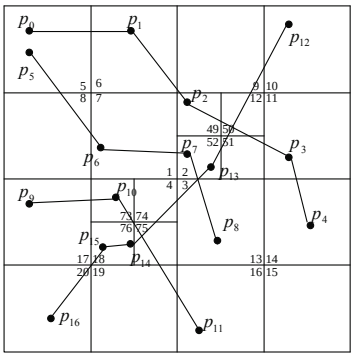
**Non-Leaf Nodes.** Each non-leaf node of the AC-tree is used to store the coordinates of four vertices of the cell and the pointers to their child nodes. If one cell is divided and the number of POIs of its $i$-th is 0, then the pointer to $i$-th child is null.

**Leaf Nodes.** Each leaf node stores the activities of some POIs which belong to some certain trajectories and fall in the range of the cell corresponding to the leaf node. In Fig. 2(a), trajectory $\tau_1$ contains $c$ and $d$, $\tau_3$ contains $a$ and $c$ in cell 52. As a result, leaf node 52 should maintain a list to store this information.

Fig. 2(c) shows an example of an AC-tree for which the threshold $\theta = 2$. For the sake of convenience, the cell_id of the $i$-th sub-cell of current cell is defined as $4 \times current\_cell\_id + i$, $dis(p, q)$ is used to denotes the minimum distance between query $q$ and any location $(x, y)$ in cell $p$.

## 5 Algorithms of RTS and ORTS

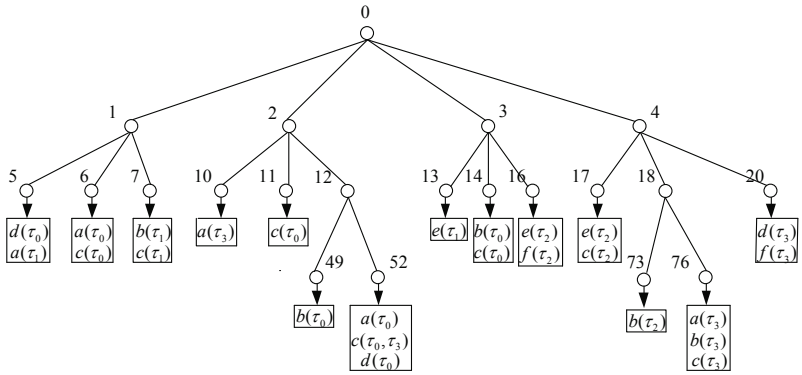
To the best of our knowledge, there is no previous work on trajectory search considering activities, spatial and ranking information simultaneously. Given a query $q$ and a threshold of distance $\widehat{d}$, the method of RTS consists of two steps.




(a)Search Regions

| *trajectory* | *poi* | *activity* | *ranking* |
|---|---|---|---|
| $\tau_0$ | $p_0$ | $d$ | $d.r:9$ |
| | $p_1$ | $(a,c)$ | $a.r:7\ c.r:6$ |
| | $p_2$ | $b$ | $b.r:2$ |
| | $p_3$ | $c$ | $c.r:6$ |
| | $p_4$ | $(d,e)$ | $d.r:5\ e.r:4$ |
| $\tau_1$ | $p_5$ | $a$ | $a.r:8$ |
| | $p_6$ | $(b,c)$ | $b.r:3\ c.r:2$ |
| | $p_7$ | $(c,d)$ | $c.r:9\ d.r:9$ |
| | $p_8$ | $e$ | $e.r:8$ |
| $\tau_2$ | $p_9$ | $(c,e)$ | $c.r:7\ e.r:8$ |
| | $p_{10}$ | $b$ | $b.r:5$ |
| | $p_{11}$ | $(e,f)$ | $e.r:3\ f.r:6$ |
| $\tau 3$ | $p_{12}$ | $a$ | $a.r:5$ |
| | $p_{13}$ | $(a,c)$ | $a.r:4\ c.r:6$ |
| | $p_{14}$ | $(b,c)$ | $b.r:7\ c.r:6$ |
| | $p_{15}$ | $(a,c)$ | $a.r:4\ c.r:8$ |
| | $p_{16}$ | $(d,f)$ | $d.r:2\ f.r:5$ |

(b)Trajectory Information




(c)AC-tree

**Fig. 2.** Details of AC-tree

Firstly, traversing the AC-tree to get a candidate set $\mathcal{C}$, in which all trajectories match query $q$. Secondly, computing the ranking of trajectory matching for each candidate trajectory. Baseline algorithm is proposed to prune the search space and get a candidate set, the detail information is presented as follows.

### 5.1 Traversing Index Tree

In this section, a new algorithm is developed to prune search space and get a candidate set $\mathcal{C}$. We commence this part by traversing AC-tree beginning at the root node with breadth-first strategy. The details are illustrated in algorithm 1.

Given a query $q$ and a threshold $\widehat{d}$, assuming $q.\varphi = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$. The retrieval of candidate trajectory consists of two steps.

**Algorithm 1.** Traversing Index Tree

**Input:** query $q$, $\widehat{d}$, AC-tree $tr$
**Output:** candidate set $\mathcal{C}$
1: Inserting the root node of $tr$ into $l$
2: $\mathcal{M}[*,*] \leftarrow 0$
3: **while** $l \neq \phi$ **do**
4: $p \leftarrow$ the first entry of $l$
5: **if** $dis(p,q) \leq \widehat{d}$ **then**
6: **if** $p$ *is a non − leaf node* **then**
7: Insert all child nodes of $p$ into $l$;
8: **else**
9: Update matrix $\mathcal{M}$;
10: **end if**
11: **end if**
12: Remove the first entry of $l$;
13: **end while**
14: **for** $j \leftarrow 1$ *to* $|\mathcal{D}|$ **do**
15: **if** *all* $\mathcal{M}[i,j] = 1$ $(i \in [1, |q.\varphi|])$ **then**
16: put trajectory $\tau_j$ into $\mathcal{C}$;
17: **end if**
18: **end for**
19: **return** $\mathcal{C}$;

*Step* 1: Compute the matrix $\mathcal{M}$:

$$\mathcal{M} = \begin{pmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & & \vdots \\ p_{m,1} & p_{m,2} & & p_{m,n} \end{pmatrix}$$

where

$$p_{i,j} = \begin{cases} 1, & if\ trajectory\ j\ contains\ query\ activity\ \alpha_i \\ 0, & otherwise \end{cases}$$

In step 1, the algorithm maintains a FIFO queue $l$ to store the nodes that should be visited. At the beginning, inserting the root node of AC-tree into $l$ and matrix $\mathcal{M}$ is initialized to be zero. For each loop, if the minimum distance between a node $p$ and query $q$ is smaller than $\widehat{d}$, we insert the child nodes of $p$ into list $l$, or update matrix $\mathcal{M}$ based on the list attached to $p$. In addition, if the distance is larger than $\widehat{d}$, there is no need to retrieve this region.

*Step* 2: Line 14 to 19 get a candidate set by processing matrix $\mathcal{M}$. For each trajectory $\tau$ in $\mathcal{D}$, it will be a candidate if $\tau$ contains all activities of $q.\varphi$.

### 5.2 Computing the Ranking of Trajectory Matching

For each trajectory $\tau$ in $\mathcal{C}$, a straightforward way to compute $r(\omega, \widehat{d})$ is to find out all possible sub-trajectories that match query $q$ and compute $r(\omega_i, \widehat{d})$, then

the algorithm returns the maximum $r(\omega_i, \widehat{d})$ as the result. However, the computation complexity of this method is too high. In the next paper, a more efficient algorithm is proposed to resolve the problem.

Given a query $q$, a threshold $\widehat{d}$ and a candidate trajectory $\tau = (p_1, p_2, \ldots, p_n)$, the details of computing $r(\omega, \widehat{d})$ is illustrated in algorithm 2 which has two steps.

**Algorithm 2.** Computing the Ranking of Trajectory Matching

**Input:** query $q$, trajectory $\tau$
**Output:** $r(\omega, \widehat{d})$

```
1: create a link list l;
2: vec ← 0, r(ω, d̂) ← 0, s ← 1, e ← 1;
3: while e ≤ l.length do
4:     if d(τ_s^e, q) ≤ d̂ then
5:         for each α ∈ p_e.φ ∧ q.φ do
6:             vec[α] + +;
7:         end for
8:     else
9:         if all vec[α] > 0(α ∈ q.φ) then
10:            compute r((τ_s^{e−1}, q), d̂) based on Eq.(2);
11:            if r((τ_s^{e−1}, q), d̂) > r(ω, d̂) then
12:                r(ω, d̂) ← r((τ_s^{e−1}, q), d̂);
13:            end if
14:        end if
15:        for each α ∈ p_s.φ ∧ q.φ do
16:            vec[α] − −;
17:        end for
18:        s ← s + 1;
19:        continue;
20:    end if
21:    e ← e + 1;
22: end while
23: return r(ω, d̂);
```

*Step* 1: For each $p_i$ in trajectory $\tau$, $p_i$ is inserted into a link list $l$ if $p_i.\varphi \wedge q.\varphi \neq \phi$ and $dis(q, p_i) \leq \widehat{d}$. Compared with using the whole trajectory $\tau$, computation cost is reduced with $l$ since less POIs are taken into account.

*Step* 2: In this section, a vector *vec* is used to keep tracking the number of occurrences of activity $\alpha(\alpha \in q.\varphi)$ in current sub-trajectory. For each loop, the algorithm updates *vec* or computes $r((\tau_s^{e-1}, q), \widehat{d})$ if necessary according to the distance between $\tau_s^e$ and $q$.

### 5.3 Computing the Ranking in Order-Sensitive Situation

In many real applications, users may want to perform their activities with an order. For instance, a worker plans to go shopping after having a haircut, i.e.,

the order of activities is *haircut* → *shopping*. Obviously, RTS is not applicable in this scenario.

**Definition 6.** ***Order-Sensitive Trajectory Matching.*** *Given a trajectory matching* $\omega = (\tau, q)$ *and a threshold of distance* $\widehat{d}$, $\omega = (\tau, q)$ *is called an order-sensitive trajectory matching, denoted as* $\omega^o = (\tau, q)^o$, *on condition that existing* $\omega_i = (\tau_i, q)$ *and for any pair of query activities*$(q_i,\ q_j,\ i < j)$ *of* $q.\varphi$ *there exists* $p_m$ *and* $p_n (m \leq n)$ *of* $\tau_i$, *such that* $q_i \in p_m.\varphi,\ q_j \in p_n.\varphi$. *The ranking of* $\omega^o = (\tau, q)^o$ *is defined as:*

$$r(\omega^o, \widehat{d}) = \max_{d(\omega_i^o) \leqslant \widehat{d}} ( \sum_{\alpha \in q.\varphi} \max_{p_j \in \tau_i} (p_j.\alpha.r)),$$

where $d(\omega_i^o) = d(\omega_i)$.

The *Order-sensitive Ranking Based Activity Trajectory Search( ORTS )* returns $k$ distinct trajectories which have the maximum $r(\omega^o, \widehat{d})$.

Consider the example in Fig. 1, assuming the orders of $\tau_1$ and $\tau_2$ are $p_{1,1} \rightarrow p_{1,2} \rightarrow p_{1,3} \rightarrow p_{1,4}$ and $p_{2,1} \rightarrow p_{2,2} \rightarrow p_{2,3} \rightarrow p_{2,4}$. If the order of $q.\varphi$ is $a \rightarrow c \rightarrow d$, the sub-trajectory $(p_{1,2}, p_{1,3}, p_{1,4})$ is the only result. This is because the order of performing $(a, c, d)$ in sub-trajectory $(p_{1,1}, p_{1,2}, p_{1,3})$ and $(p_{2,1}, p_{2,2}, p_{2,3})$ do not keep the order of $\tau_1$ and $\tau_2$.

The same with RTS, ORTS consists of two steps of traversing AC-tree and computing the ranking of trajectory matching. The algorithm 1 is still adopted in this part to prune search space. However, computing the ranking of trajectory matching in order-sensitive case is more challenging, as it needs to make the order of performing activities in a trajectory consistent with the query and try to maximize the ranking within $\widehat{d}$. Given a candidate trajectory, a naive method of ORTS is to enumerate all possible sub-trajectory matches and find the maximum $r(\omega_i^o, \widehat{d})$. Clearly, this is not efficient. A new algorithm is illustrated in the rest of this paper.

Given a trajectory $\tau = (p_1, p_2, \ldots, p_n)$ and a query $q$, let $q.\varphi = (\alpha_1, \alpha_2, \ldots, \alpha_m)$. We define an $m \times n$ matrix $\mathcal{R}$ such that its element $\mathcal{R}[i, j] (1 \leq i \leq m,\ 1 \leq j \leq n)$ denotes the maximum ranking between the sub-query $q_1^i.\varphi = (\alpha_1, \alpha_2, \ldots, \alpha_i)$ and the sub-trajectory $\tau_1^j = (p_1, p_2, \ldots, p_j)$. The element of $\mathcal{R}$ is given as follows:

$$\mathcal{R}[i, j] = \max_{1 \leq k \leq j} \{\mathcal{R}[i-1, k] + mr(\alpha_i, \tau_k^j)\} \tag{3}$$

where $mr(\alpha_i, \tau_k^j)$ is the maximum ranking of $\alpha_i$ in sub-trajectory $\tau_k^j$.

Computing the ranking of trajectory matching in order-sensitive is illustrated in algorithm 3, which consists of two steps. Firstly, creating a link list $l$. Secondly, computing $r(\omega^o, \widehat{d})$. Different from algorithm 2, this algorithm updates matrix $\mathcal{R}(*, *)$ according to Eq.(3).

## 6 Optimization

As described in Section 5, there is no need to search regions which are beyond $\widehat{d}$. However, it needs to compute the ranking of trajectory matching for each

**Algorithm 3.** Computing the Ranking in Order-sensitive Case

**Input:** query $q$,trajectory $\tau$
**Output:** $r(\omega^o, \widehat{d})$

create a link list $l$;
$\mathcal{R}(*,*) \leftarrow 0$, $vec \leftarrow 0$, $s \leftarrow 1$, $e \leftarrow 1$, $r(\omega^o, \widehat{d}) \leftarrow 0$;
**while** $e \leq l.length$ **do**
  **if** $d(\omega_s^e, q) \leq \widehat{d}$ **then**
    **for** *each* $\alpha \in p_e.\varphi \wedge q.\varphi$ **do**
      $vec[\alpha] + +$;
    **end for**
  **else**
    **if** *all* $vec[\alpha] > 0 (\alpha \in q.\varphi)$ **then**
      $\mathcal{R}(*,*) \leftarrow 0$;
      Update each element $\mathcal{R}[i,j]$ based on Eq.(3)
      **if** $\mathcal{R}[|q.\varphi|, |\tau|] > r(\omega^o, \widehat{d})$ **then**
        $r(\omega^o, \widehat{d}) \leftarrow \mathcal{R}[|q.\varphi|, |\tau|]$;
      **end if**
    **end if**
    **for** *each* $\alpha \in p_s.\varphi \wedge q.\varphi$ **do**
      $vec[\alpha] - -$;
    **end for**
    $s \leftarrow s + 1$;
    continue;
  **end if**
  $e \leftarrow e + 1$;
**end while**
**return** $r(\omega^o, \widehat{d})$;

trajectory in $\mathcal{C}$. It is costly especially when the cardinality of $\mathcal{C}$ is large. In order to improve the efficiency of query, we optimize the AC-tree and the algorithm of computing the ranking of trajectory matching. Two components of the optimization are illustrated as follows.

*Componet* 1: In this section, a RAC-tree is proposed, which is the variant of AC-tree. As depicted in Fig. 3, each non-leaf node of RAC-tree contains all activities that can be fulfilled in its child nodes. Shown in Fig.2(a), the activities that can be performed in child cells of 3 are $(b, c, e, f)$, then this information is inserted into node 3.

The leaf nodes of RAC-tree not only contain activity information but also ranking information. Seen from Fig. 3, the entry of a list attached to a leaf node is a tuple in the form of $\alpha(\tau_i : \alpha.r)$, where $\tau_i$ is the trajectory that contains $\alpha$ and $\alpha.r$ represents the maximum ranking of $\alpha$ in current leaf node in trajectory $\tau_i$. Besides, the definition of each element of matrix $\mathcal{M}$ is changed:

$$p_{i,j} = \begin{cases} \alpha_i.r, & if\ trajectory\ j\ contains\ query\ activity\ \alpha_i \\ 0, & otherwise \end{cases}$$

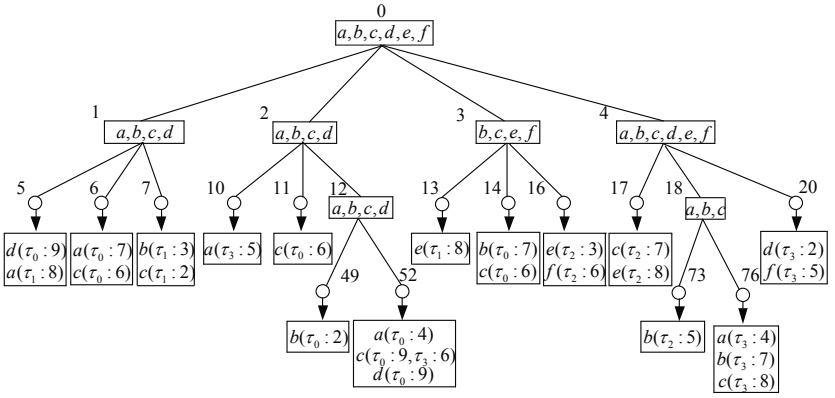



**Fig. 3.** RAC-tree

where $\alpha_i.r$ is the maximum ranking of query activity $\alpha_i$ in trajectory $j$ within $\widehat{d}$. As shown in Fig 2(a) and Fig 2(b), cell 76 contains activities $a$, $b$ and $c$, and the maximum rankings of them are: (4, 7, 8), so the entry attached to the leaf node 76 is $(a(\tau_3 : 4), b(\tau_3 : 7), c(\tau_3 : 8))$.

Compared with using AC-tree, the query is more efficient with the RAC-tree. On one hand, the search regions are pruned with spatial and activity information, as there is no need to consider the tree node $p$ which contains no activity of $q.\varphi$ even if $dis(q, p) \leq \widehat{d}$. On the other hand, the candidate set $\mathcal{C}$ can be pruned fast by the RAC-tree. Given a candidate trajectory $\tau_j$, if $\tau_j$ matches query $q$, it follows that $r((\tau_j, q), \widehat{d}) \leq \sum_{i=1}^{|q.\varphi|} p_{i,j}$ according to the definition of matrix $\mathcal{M}$. As a result, if $\sum_{i=1}^{|q.\varphi|} p_{i,j} < res[k]$, $\tau_j$ can be pruned from $\mathcal{C}$, where $res[k]$ denotes the current $k$th maximum result.

*Component* 2: In this section, algorithm 2 and 3 are optimized. Different from taking all $p_i(p_i.\varphi \wedge q.\varphi \neq \phi)$ of $l$ into consideration, a new list $l = \{(p_1, p_1.r), (p_2, p_2.r), \ldots, (p_n, p_n.r)\}$ is created, where $p_i$ is the POI such that $p_i.\varphi \wedge q.\varphi \neq \phi$ and $dis(p_i, q) \leq \widehat{d}$, and $p_i.r = \sum_{j=1}^{|q.\varphi|} \alpha_j.r$, where $\alpha_j.r$ denotes the maximum ranking of query activity $\alpha_j$ from $p_i$ to the last node of $l$.

Given a candidate trajectory $\tau$ and its corresponding $l$, we have $p_s.r \geq p_{s'}.r (s' > s)$. As a result, if there exists $(p_s, p_s.r) \in l$ such that $p_s.r < res[k]$ or $p_s.r < r((\tau, q), \widehat{d})$, then there is no need to compute any $r((\tau_{s'}^{e'}, q), \widehat{d})$ since $r((\tau_{s'}^{e'}, q), \widehat{d}) \leq p_{s'}.r$.

## 7 Experimental Study

In this section, we conduct extensive experiments on half-real datasets to demonstrate the performance of proposed indexes and algorithms. The settings of experiment are presented in table 2.

**Table 2.** Settings of Experiment

| Parameter | Range | Default Value |
|---|---|---|
| $\|\mathcal{D}\|$ | 10k-50k | 50k |
| $\|q.\varphi\|$ | 3-7 | 5 |
| results $k$ | 5-25 | 10 |
| activity | 30-50 | 40 |
| $\widehat{d}$ | 4km-8km | 6km |

We study the query time of different algorithms. 1) Using hybrid index AC-tree to organize trajectory data and unoptimized algorithm to compute the ranking of trajectory matching, denoted as B-RTS and B-ORTS for RTS and ORTS respectively. 2) RAC-tree and upoptimized algorithm based method, denoted as R-RTS and R-ORTS. 3) We use C-RTS and C-ORTS to denote organizing trajectory data with RAC-tree and using optimized algorithm to compute the ranking. Especially, we set $\theta = 200$ in this study, which means that if the number of POIs in each grid cell exceeds 200, it should be partitioned into four components.

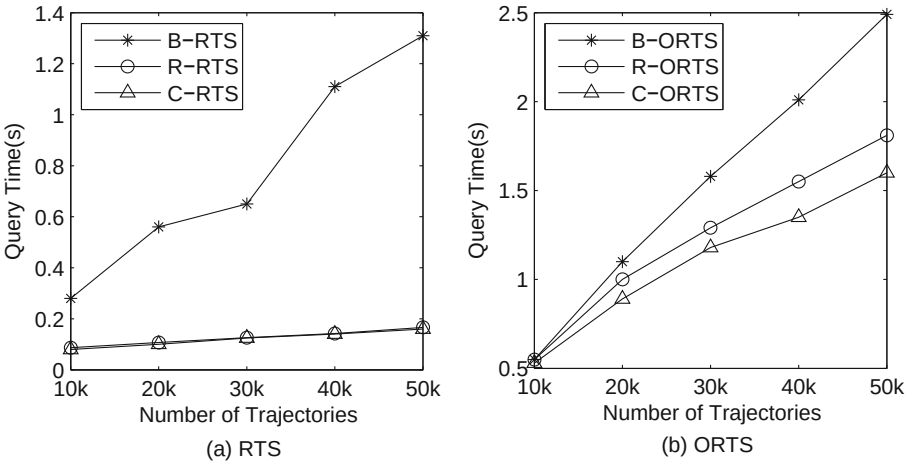



**Fig. 4.** Effectiveness of $|\mathcal{D}|$

**Effectiveness of $|\mathcal{D}|$.** First of all, we study the scalability of the three approaches by comparing the time cost of them. In Fig. 4(a) and 4(b), the cardinality of $\mathcal{D}$ varies from 10k to 50k. With no surprise, it needs more time in query with the increasing of the number of trajectories. Besides, from Fig. 4 we know that using RAC-tree to organize trajectory data outperforms AC-tree. The optimization of computing the ranking of trajectory matching is not sensitive in disorder case. However, this optimization is sensitive in order-sensitive case.

**Effectiveness of Number of Activities.** We also study the performance of proposed indexes and algorithms while varying the number of activities from 30 to 50. Seen from Fig. 5(a) and 5(b), with the increasing of the number of activities the query time becomes less, which is expected. For each trajectory, the number of different kinds of activities contained by which becomes smaller with the increasing of the number of activities. As a result, a trajectory is less likely to be a result and the cardinality of candidate set also becomes smaller, which results in the decreasing of query time.

**Effectiveness of $\widehat{d}$.** Another important concern of query is the threshold of distance. Shown in Fig.5(c) and 5(d), we set $\widehat{d}$ from 4km to 8km. With no surprise, the query spends more time with the increasing of $\widehat{d}$, even though it is

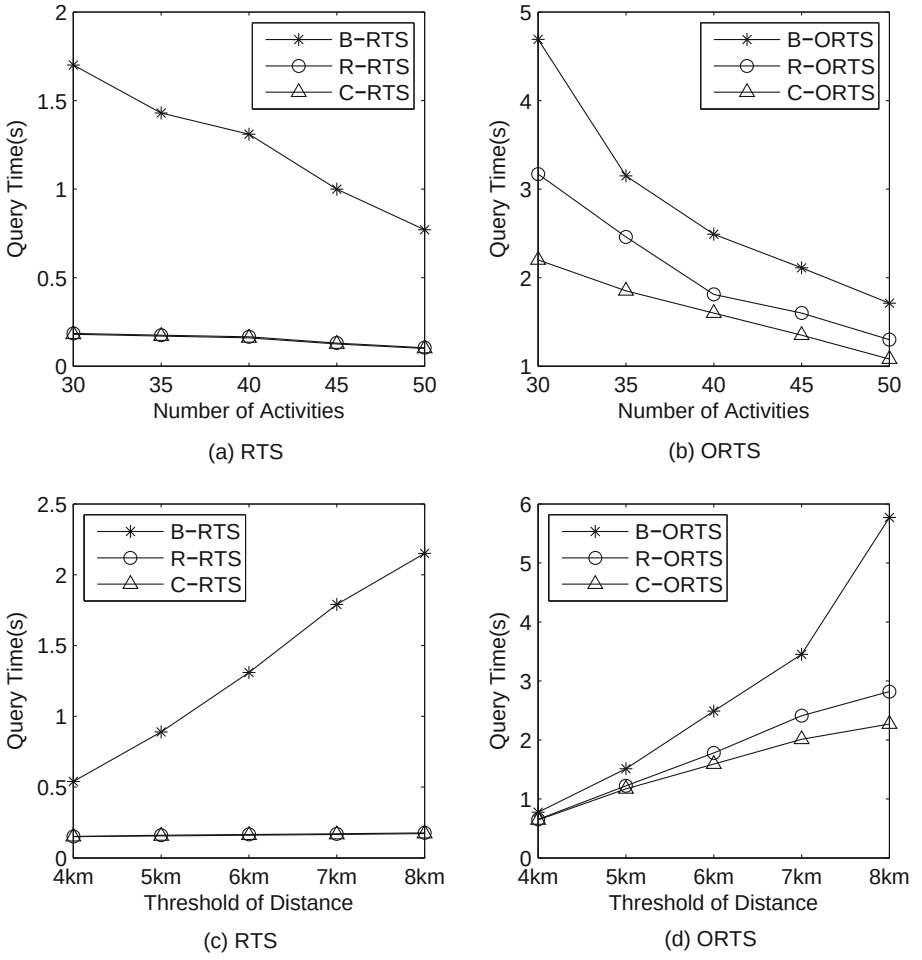



**Fig. 5.** Effectiveness of $\widehat{d}$ and the number of activities

not obvious for B-RTS and R-RTS. A trajectory is more likely to be a candidate and the number of candidate trajectories becomes lager, with a greater $\widehat{d}$.

**Effectiveness of $k$.** Fig.6(a) and 6(b) show the query time while varying $k$ from 5 to 25. Different from R-RTS, C-RTS, R-ORTS and C-ORTS the time cost of B-RTS and B-ORTS almost has no significant change for different $k$. This is because it needs to compute $r(\omega, \widehat{d})$ for each trajectory in candidate set $\mathcal{C}$, which has the same cardinality for different $k$. For R-RTS, C-RTS, R-ORTS and C-ORTS the current $k$th maximum result tends to be smaller as $k$ increases, which means that more candidate trajectories should be taken into account. Hence, the query time of them monotonously increases with the increasing of $k$.

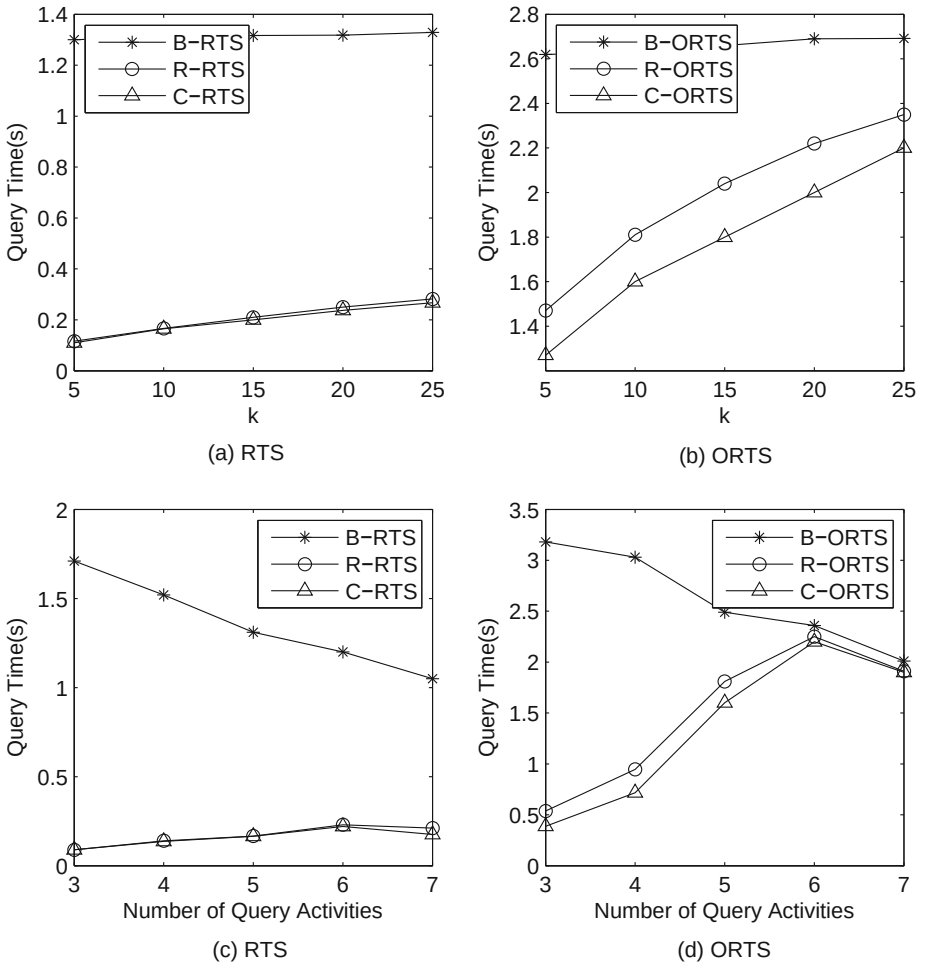



**Fig. 6.** Effectiveness of $k$ and $q.\varphi$

**Effectiveness of** $|q.\varphi|$. Finally, we investigate the performance while changing the number of query activities. As depicted in Fig. 6(c) and 6(d), query time of B-RTS and B-ORTS monotonously decreases with the increasing of $q.\varphi$. This is because a trajectory is less likely to be a candidate and the size of the candidate set becomes smaller as $q.\varphi$ increases.

## 8 Conclusions

This paper studies a novel problem of searching trajectories with activities, spatial and ranking information. In order to tackle the problem efficiently, we propose an AC-tree to organize trajectory data and develop novel algorithms, named RTS and ORTS, to compute the ranking of trajectory matching. As a progress, we optimize the AC-tree by developing a RAC-tree to prune the search space, and optimized algorithms are also proposed to improve the efficiency of query. Experimental results show that the optimization of index structure and algorithms achieves high efficiency and scalability.

**Acknowledgements.** This work was supported by NSFC grant 61073061, 61303019, 61003044 and 61232006, Doctoral Fund of Ministry of Education of China 20133201120012, and Jiangsu Provincial Department of Education grant 12KJB520017.

## References

1. Shang, S., Ding, R.G., Yuan, B., Xie, K.X., Zheng, K., Kalnis, P.: User Oriented Trajectory Search for Trip Recommendation. In: Proceedings of the 15th International Conference on Extending Database Technology, pp. 156–167 (2012)
2. Cong, G., Jensen, C.S., Wu, D.M.: Efficient Retrieval of The Top-k Most Relevant Spatial Web Objects. In: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, pp. 337–348 (2009)
3. Chen, Z.B., Shen, H.T., Zhou, X.F., Zheng, Y., Xie, X.: Searching Trajectories by Locations-An Efficiency Study. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, pp. 255–266 (2010)
4. Reza, S., Davood, R.: On Efficiently Searching Trajectories and Archival Data for Historical Similarities. In: Proceedings of the 2008 VLDB Endowent, pp. 896–908 (2008)
5. Chen, Z.B., Shen, H.T., Zhou, X.F.: Discovering Popular Routes from Trajectories. In: Proceedings of the 2011 ICDE International Conference on Data Engineering, pp. 900–911 (2011)
6. Chen, L., Ozsu, M.T., Oria, V.: Robust and Fast Similarity Search for Moving Object Trajectories. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 491–502 (2005)
7. Vlachos, M., Kollios, G., Gunopulos, D.: Discovering Similar Multidimensional trajectories. In: Proceedings of the 2002 ICDE International Conference on Data Engineering, pp. 673–684 (2002)

8. Xue, A.Y., Zhang, R., Zheng, Y., Xie, X., Huang, J., Xu, Z.H.: Destination Prediction by Sub-Trajectory Synthesis and Privacy Protection Against Such Prediction. In: Proceedings of the 2013 ICDE International Conference on Data Engineering, pp. 254–265 (2013)
9. Nergiz, M.E., Atzoir, M., Sayqin, Y.: Towards Trajectory Anonymization: a Generalization-based approach. In: Proceedings of the 2008 Internatioal Workshop on Security and Privacy in GIS and LBS, pp. 52–61 (2008)
10. Jeung, H.Y., Liu, Q., Shen, H.T., Zhou, X.F.: A Hybrid Prediction Model for Moving Objects. In: Proceedings of the 2008 ICDE International Conference on Data Engineering, pp. 70–79 (2008)
11. Terrovitis, M., Manoulis, N.: Privacy Preservation in the Publication of Trajectories. In: Proceedings of the 2008 MDM International Conference on Mobile Data Management, pp. 65–72 (2008)
12. Gidofalvi, G., Huang, X.G., Perdersen, T.B.: Privacy-Preserving Data Mining on Moving Object Trajectories. In: Proceedings of the 2007 MDM International Conference on Mobile Data Management, pp. 60–68 (2007)
13. Zheng, K., Shang, S., Yuan, N.J., Yang, Y.: Towards Efficient Search for Activity Trajectories. In: Proceedings of the 2013 ICDE International Conference on Data Engineering, pp. 230–241 (2013)
14. Zhou, Y.H., Xie, X., Wang, C., Gong, Y.C., Ma, W.Y.: Hybrid Index Structures for Location-based Web Search. In: Proceedings of the 14th ACM International Conference on Informaiton and Knowledge Management, pp. 155–162 (2005)
15. Tao, Y.F., Faloutsos, C., Papadis, D., Liu, B.: Prediction and indexing of moving objects with unknown motion patterns. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, pp. 611–622 (2004)
16. Chen, Y.Y., Suel, T., Markowetz, A.: Efficient Query Processingin Geographic Web Search Engines. In: Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, pp. 277–288 (2006)
17. Felipe, I.D., Hristidis, V., Rishe, N.: Keyword Search on Spatial Databases. In: Proceedings of the 2008 ICDE International Conference on Data Engineering, pp. 656–665 (2008)
18. Hariharan, R., Hore, B., Li, C., Mehrotra, S.: Processing Spatial-Keyword (SK) Queries in Geographic Information Retrieval (GIR) Systems. In: Proceedings of the 2007 SSDBM international conference on Scientific and Statical Database Management, p. 16 (2007)
19. Cao, X., Cong, G., Jensen, C.S., Ooi, B.C.: Collective Spatial Keyword Querying. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pp. 373–384 (2011)
20. Simmons, R., Browning, B., Zhang, Y., Sadekar, V.: Learning to Predict Driver Route and Destination Intent. In: ITSC, pp. 127–132 (2006)
21. Patterson, D.J., Liao, L., Fox, D., Kautz, H.: Inferring High-Level Behavior from Low-Level Sensors. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) UbiComp 2003. LNCS, vol. 2864, pp. 73–89. Springer, Heidelberg (2003)
22. Long, C., Wong, R., Wang, K., Fu, A.: Collective Spatial Keyword Queries: A Distance Owner-Driven Approach. In: SIGMOD, pp. 689–700 (2013)
23. Hashem, T., Kulik, L., Zhang, R.: Privacy Preserving Group Nearest Neighbor Queries. In: Proceedings of the 13th International Conference on Extending Database Technology, pp. 489–500 (2010)
24. Abul, O., Bonchi, F., Nanni, M.: Never Walk Alone: Uncertainty for Anonymity in Moving Objects Databases. In: Proceedings of the 2008 ICDE International Conference on Data Engineering, pp. 376–385 (2008)

# A Multicriterion Query-Based Batch Mode Active Learning Technique

**Yang Jiao, Pengpeng Zhao, Jian Wu, Yujie Shi and Zhiming Cui**

**Abstract** Active learning is well-motivated in many modern machine learning problems, where unlabeled data may be abundant or easily obtained, but labels are difficult, time-consuming, or expensive to obtain. The selection and use of sampling strategy is core of active learning. Most active learning methods select those uncertain or representative unlabeled samples to query their labels. And some of the active learning methods consider both in the query selection. However, the uncertainty sampling methods rely on the relative correctness or confidence of the current model and suffer from a lack of the feature space. The representative sampling methods avoid the drawbacks associated with uncertainty sampling, but tend not to improve the learning model very efficiently. The combining methods are lack of the consideration of sample redundancy. This paper proposes a multicriterion active learning technique for solving multiclass problems. First, use the Best-versus-Second-Best (BvSB) method to calculate the sample's uncertainty and then select the most valuable component to constitute the uncertain set; further, use the kernel *k*-means clustering algorithm and the resulting sample set is divided into *h* different clusters; finally, use Gaussian process to select the most informative sample in each cluster and submit to human experts for annotation. The results show that the labeling cost can be reduced without degrading the performance.

**Keywords** Active learning · Sampling strategy · Uncertainty · Representative · Diversity

Y. Jiao · P. Zhao (✉) · J. Wu · Y. Shi · Z. Cui
Department of Computer Science and Technology, Soochow University, Suzhou 215006, China
e-mail: ppzhao@vip.sina.com

Z. Wen and T. Li (eds.), *Foundations of Intelligent Systems*,
Advances in Intelligent Systems and Computing 277,
DOI: 10.1007/978-3-642-54924-3_63, 

## 1 Introduction

Many supervised learning algorithms of the machine learning area have been widely used in pattern recognition tasks. In all of these methods, the accuracy of classifier depends heavily on the labeled sample set. However, in reality, to obtain the training samples is very easy while the labeled samples are scarce, so to get labeled samples requires a high price. Moreover, redundant-labeled samples may slowdown the training speed, while they are not helpful to the classifier. In order to reduce the time and cost of labeling [1], which requires that the sample selected during training, not only has the information content but be diversed from each other. Active learning [2] is an effective method to solve these problems. Active learning algorithms select high-information content unlabeled examples to be labeled by experts [3, 4], several loops make the correct classification accuracy gradually increased, and thus the classifier obtains the strong generalization ability in the case of minimum labeling cost. Compared with the traditional supervised learning methods, active learning can significantly reduce time and cost of labeling. How to choose samples, as few as possible to get the higher classification accuracy, are the core issue of active learning algorithm. Therefore, the sampling strategy [5] naturally becomes a concern of active learning algorithms.

The traditional sampling methods are generally divided into two categories: one is based on uncertainty [6], merely use the uncertainty to measure sample information content, select the most uncertain samples of classification for labeling; although this method has a wide range of applications and achieved good results in practical tasks, but it only considers the relationship between current sample and the labeled samples, ignoring the distribution information of unlabeled sample set. Thus the important issue is that the unavoidable choice of outliers in the training process may reduce the classification accuracy. Another method surmounts the difficulty of uncertainty sampling, considering the sample of uncertainty and representativeness [7]. But for different data, it is difficult to measure the importance level of uncertainty and representativeness, i.e., the respective weights of uncertainty and representativeness; moreover these methods did not consider redundancy between selected samples. Some batch mode active learning methods will suffer from the same problem. In order to accelerate the learning process, it is necessary to speed up the learning process by selecting more than one sample at each iteration for batch mode active learning methods. So it needs to consider the diversity of the selected samples. To solve the above problems, this chapter proposes a multicriterion query-based active learning method. Experimental results show that our proposed method has achieved good performance.

## 2 Related Work

A general active learner can be modeled as a quintuple (G, Q, S, L, U) [8]. Initially, the training set L has few labeled samples to train the classifier G. After that, the query function Q is used to select a set of most informative samples from the unlabeled pool U and the supervisor S assigns a class label to each of them. Then, these new labeled samples are included into L and the classifier G is retrained using the updated training set. The closed loop of querying and retraining continues for some predefined iterations or until a stop criterion is satisfied.

There has been more research work in the sampling strategy of active learning. Uncertainty-based sampling methods are more commonly used. Entropy is most commonly used in uncertainty sampling method. Sample entropy can better represent the uncertainty of samples, i.e., the greater the entropy, the greater the uncertainty of the sample. However, in multiclass problems, the entropy does often not well reflect the uncertainty of the sample. Some may have larger classification uncertainty than the ones whose entropy may be higher. For the above problem, Joshi [9] proposed a more direct active learning sample selection criteria Best-versus-Second-Best (BvSB). The BvSB method considers the difference between the probability values of the two classes having the highest estimated probability value as a measure of uncertainty. The practical applications of the method get a better performance. Another common sampling strategy is based on the reduction of version space. Query-by-committee (QBC) algorithm is the most widely used famous algorithm. The committee, which is constituted by a set of group classifiers. Each committee member is then allowed to vote on the labelings of query candidates. The most informative query is considered to be the instance about which they most disagree. In essence, the QBC is based on uncertainty sampling. However, these methods only consider the impact on the labeled samples, without considering the distribution of unlabeled sample set, ignoring the relationship with unlabeled samples. The literature [7] shows that unlabeled samples have a great influence on classification accuracy. If the current sample can better represent the remaining unlabeled samples, then we say that the sample has a high-representative, meaning that the information content is higher. There have been some studies for a combination of uncertainty and representative aspects. Settles and Craven [7] proposed information density(ID) method, first with uncertainty methods measure the current sample basic information content, then use the sample feature vector cosine similarity method to calculate average similarity to all other samples in the input distribution, information is then multiplied by the density and set a fixed threshold to control the weight of density items. However, in many problems it is necessary to speed up the learning process by selecting more than one sample at each iteration. Li and Sethi [8], proposed that estimates the uncertainty level of each sample according to the output score of a classifier and selects only those samples whose output scores are within the uncertainty range. Patra and Bruzzone [10], proposed a fast clustering-based active learning method to solve the multiclass classification problems. However, these methods

select batch of samples at each iteration by considering only the uncertainty criterion. This will result in the selection of redundant samples which reduce the speed of the classifier without adding any additional information. To solve this problem, Brinker [11] presented a batch mode method of considering the diversity. In the literature [12, 13], the clustering method to measure the diversity was introduced into the design of active learning query function. For the combination of uncertainty, representation, and diversity, there is also some research. Lin and Bilmes [14] also studied batch mode active learning with submodular graph functions for the problem of training hidden Markov models for speech recognition, but this method is mainly designed for representativeness. Diversity, density, and relevance (analogous to uncertainty) are all incorporated in a query criterion by Xu [15] et al. but the approach is to simply interpolate three scores with two empirically-tuned weights. Tuning weights for active learning is more challenging in a real scenario than for classification accuracy.

Different from the above methods, we propose an active learning algorithm which integrates different selection criteria. The framework of our method consists of three key components: uncertainty, representativeness, and diversity criterion. Compared to the previous work this method has the advantages of (1) take into account the relationship with the labeled samples, but also make full use of the remaining unlabeled samples, which ensure the samples with higher uncertainty and better representation. In addition, considering the correlation between the selected samples, so that the selected samples are diverse; (2) without respecting to the weights of uncertainty, representativeness, and diversity. The first consideration of our method is the uncertainty of samples, and then with a combination of representativeness, and diversity. (3) Compared to the methods that directly deal with all the unlabeled samples, our method can greatly reduce the cost of sample selection, thereby improving efficiency.

## 3 MCQAL: Multicriterion Query-Based Active Learning

### *3.1 Problem Analysis*

A limitation of the uncertainty sampling strategy is that it relies on the relative correctness or confidence of the current model, which can be a difficulty, especially in the early stages. And the uncertainty sampling can also suffer from a lack of exploration of the feature space and may not work well in some scenarios. As shown in Fig. 1, the red triangles and green circles represent the instances which have been labeled, the remaining white diamonds represent unlabeled sample set. Since sample $x_A$ lies on the classification boundary, it must be selected by using uncertainty sampling strategy for human experts to label. In fact, $x_B, x_C, x_D$ are samples with higher information content, they can better reflect the

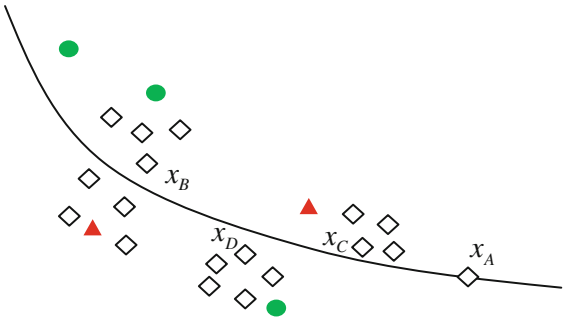



**Fig. 1** An illustration of when uncertainty sampling can be a poor strategy

distribution of the sample set. Therefore, we make a combination of representativeness and diversity criteria based on the uncertainty sampling strategy, thus this part of samples will be selected together.

## 3.2 Uncertainty of Sample Selection

In the uncertainty sampling strategy, we employ the BvSB [9] approach, which consider the difference between the probability values of the two classes having the highest estimated probability value as a measure of uncertainty. Say that our estimated probability distribution for a certain example is denoted by $P$, where $p_i$ denotes the membership probability for class $i$. Also suppose that the distribution $P$ has a maximum value for class $h$. Based on current knowledge, the most likely set of classifiers in contention is $C_h$. The classification confidence for the classifiers in this set is indicated by the difference in the estimated class probability values, $p_h - p_i$. This difference is an indicator of how informative the particular example is to a certain classifier. Minimizing the difference $p_h - p_i$, or equivalently, maximizing the confusion (uncertainty), we obtain the BvSB measure.

$$\begin{aligned} \text{BvSB}^* &= \arg\min_{x_i \in U}\Big(\min_{y \in Y, y \neq y_{\text{Best}}} (p(y_{\text{Best}}|x) - p(y|x))\Big) \\ &= \arg\min_{x_i \in U}(p(y_{\text{Best}}|x) - p(y_{\text{Second-Best}}|x)) \end{aligned} \tag{1}$$

## 3.3 Representativeness Measure

In addition to the most informative sample, we also prefer the most representative sample. The representativeness of a sample can be evaluated based on how many samples are similar or near to it. So, the samples with high-representative degree are less likely to be an outlier. Adding them to the training set will have an effect on a large number of unlabeled samples. In this section, we use the Gaussian

Process [16] model to measure the mutual information between the uncertain set and remaining unlabeled samples.

Using mutual information criterion, we define the mutual information-based representativeness measure for a candidate sample $x_i$ as below

$$\mathrm{rep}(x_i) = I(x_i, U_{x_i}) = H(x_i) - H(x_i|U_{x_i}) \tag{2}$$

where $U_{x_i}$ denotes the index set of unlabeled instances after removing $x_i$ from $U$.

We propose to compute the entropy terms in (2) within a Gaussian Process framework. A Gaussian Process is a joint distribution over a (possibly infinite) set of random variables, such that the marginal distribution over any finite subset of variables is multivariate Gaussian. For our problem, we associate a random variable $\chi(x)$ with each instance. A symmetric positive definite Kernel function $K(\cdot,\cdot)$ is then used to produce the covariance matrix, such that

$$\sigma_i^2 = K(x_i, x_i) \tag{3}$$

$$\sum_{U_i U_i} = \begin{pmatrix} K(x_1,x_1) & K(x_1,x_2) & \dots & K(x_1,x_u) \\ K(x_2,x_1) & K(x_2,x_2) & \dots & K(x_2,x_u) \\ \vdots & \vdots & \vdots & \vdots \\ K(x_u,x_1) & K(x_u,x_2) & \dots & K(x_u,x_u) \end{pmatrix} \tag{4}$$

where the covariance matrix $\sum_{U_i U_i}$ is actually a kernel matrix defined over all the unlabeled instances indexed by $U_i$, we assume $U_i = \{1, 2, \dots, u\}$. One commonly used kernel function is the Gaussian kernel $K(x_i, x_j) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\lambda^2}\right)$.

Closed-form solutions exist for the entropy of multivariate Gaussian distributions such that

$$H(x_i) = \frac{1}{2}\ln\left(2\pi e \sum_{ii}\right) \tag{5}$$

$$H(x_i|U_{x_i}) = \frac{1}{2}\ln\left(2\pi e \sum_{i|U_i}\right) \tag{6}$$

using (5) and (6), the representativeness definition given in (2) can finally be rewritten into the following form

$$\mathrm{rep}(x_i) = H(x_i) - H(x_i|U_{x_i}) = \frac{1}{2}\ln\left(\frac{\sum_{ii}}{\sum_{i|U_i}}\right) \tag{7}$$

### 3.4 Diversity Analysis

Diversity criterion is to maximize the training utility of a batch. We prefer the batch in which the examples have high variance to each other. In this step we cluster all samples in the uncertainty set based on the representativeness measure proposed in Sect. 3.3. The samples in the same cluster may be considered similar to each other, so we will select the most informative sample from different clusters at one time. We apply the kernel $k$-means clustering algorithm.

In greater detail, let us assume that the kernel $k$-means clustering algorithm divides the $m$ samples into $h$ clusters $C_1, C_2, \ldots, C_h$. After $C_1, C_2, \ldots, C_h$ are obtained, the $h$ most informative samples are selected as

$$x_k = \arg\min_{x \in C_k} \mathrm{rep}(x), k = 1, 2 \ldots, h. \tag{8}$$

### 3.5 A Combination Framework

In this section, we will study how to combine and strike a proper balance between these criteria, to reach the maximum effectiveness.

We first consider the uncertainty criterion. We choose $m$ samples with the most informativeness score from all of the pool. By this preselecting, we make the selection process faster in the later steps since the size of uncertainty set is much smaller than that of the pool. Then we cluster the samples in uncertainty set and choose the centroid of each cluster into a batch. The centroid of a cluster is the most representative sample in that cluster since it has the largest density. Furthermore, the samples in different clusters may be considered diverse to each other. By this means, we consider representativeness and diversity criteria at the same time. We will summarize our overall algorithm, shown in Table 1.

## 4 Experimental Results

### 4.1 Design of Experiments

In order to assess the effectiveness of our algorithm, four international standard UCI data sets were used in the experiments data. Table 2 shows the information of data sets. As a comparison, we choose (1) information density (ID) active learning [7], a representative approach which selects informative and representative instances (2) A cluster-assumption-based batch mode active learning technique [17], a representative approach which selects informative and diverse instances.

**Table 1** The pseudo-code of MCQAL is summarized in Algorithm 1

Algorithm 1:

Input: labeled data set $L$ unlabeled data set $U$
  Repeat
    Training on $L$ to get the probabilistic classification model $\Theta$
    for each $x_i$ in $U$
      Use (1) to measure the uncertainty of sample $x_i$
    end for
    Select the most uncertainty sample set MUSS;
    for each $x_j$ in MUSS
      Use (7) to measure the representativeness of sample $x_j$
    end for
    Apply kernel $k$-means clustering algorithm to the MUSS;
    Select one sample from each of the $h$ clusters using (8)
    Assign true labels to the $h$ selected samples
    $L = L \cup C^*$ $U = U \backslash C^*$
Until the stop criterion is satisfied

**Table 2** Experimental data information table

| Data set | Classes | Features | Unlabeled | Test |
|---|---|---|---|---|
| Ionosphere | 2 | 34 | 246 | 105 |
| Letters | 26 | 16 | 10000 | 10000 |
| Pen-Digits | 10 | 16 | 7494 | 3498 |
| Balance-scale | 3 | 4 | 166 | 459 |

A RBF kernel with default parameters is used (performances with linear kernel are not as stable as that with RBF kernel). LibSVM is used to train a SVM classifier for all active learning approaches in comparison. To reduce the classification error, for every data set, we run the experiment for 10 times, each with a random partition of the data set. In the next section, we present the results of three different experiments. In the first experiment, we compared the accuracy of the proposed technique with other two techniques by using four datasets. The second experiment shows the diversity of samples during selecting. The computational load of the different methods is analyzed in the third experiment.

## *4.2 Results*

Figure 2 shows the classification accuracy of different active learning approaches with varied numbers of queries.

For the two data sets Ionosphere and Letters, the proposed method is superior to the other two methods: When the number of labeled samples is small, our method

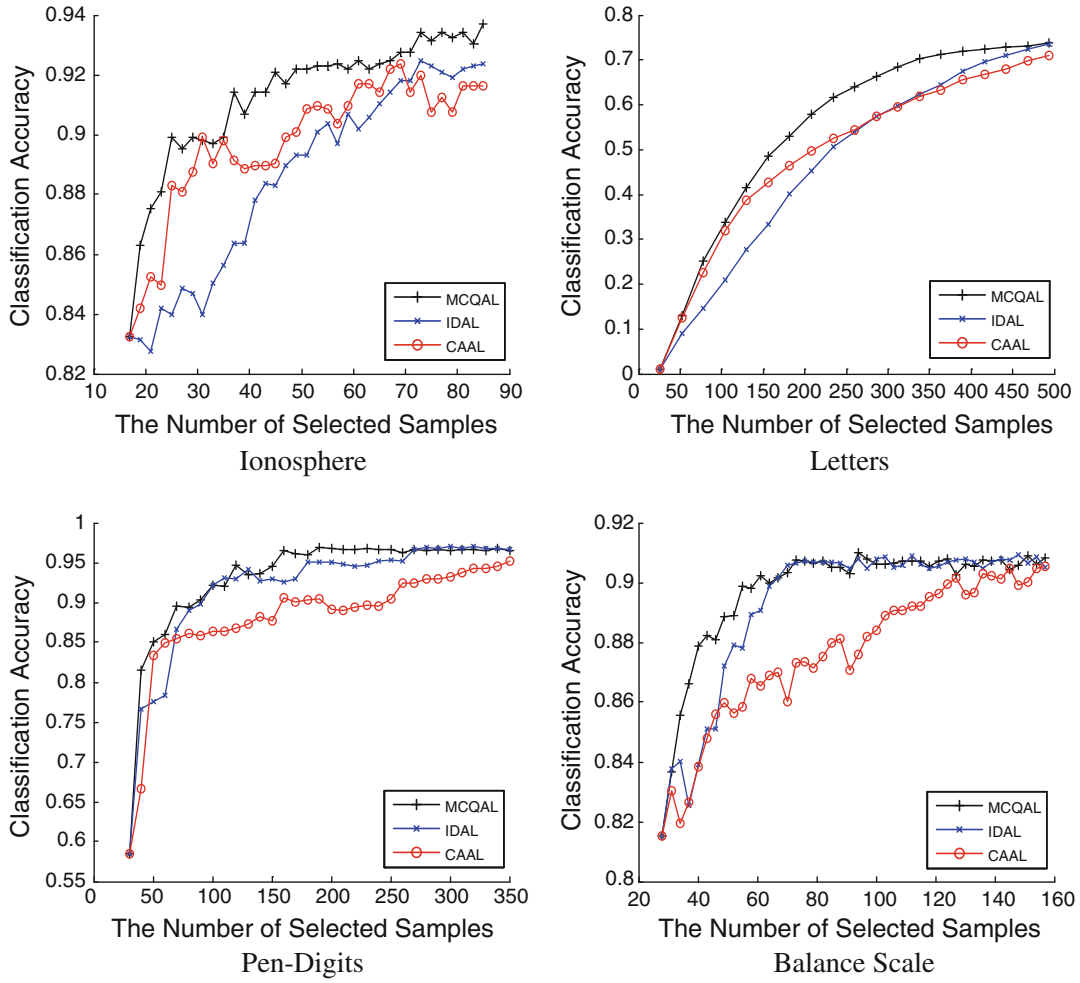



**Fig. 2** Classification accuracy of different active learning approaches in four datasets

and CAAL are similar, but significantly better than IDAL; With the increase number of labeled samples, the superiority of our method gradually reflected and significantly better than CAAL. So in each iteration, the samples selected by our method are more informative (representative). In the condition of same labeled samples, our method is more effective to increase the classification accuracy.

Pen-Digits dataset, since our method took a priority on the basis of uncertainty and then selected samples both of representativeness and diversity, so the performance will be influenced by the uncertainty sample set. As shown in the section of the curve, we can see that the sample selected is not optimal. Those samples with high-representative but low-uncertainty may be ignored. In contrast, samples of higher information content are selected by IDAL.

The Balance Scale data set is imbalanced of class distribution. For imbalanced data sets, certain categories of samples are extremely rare, but often this part samples have higher information content. Therefore, it needs to pick up them as possible submitted to human experts for annotation. The result can be seen that our approach also received great performance on imbalanced dataset. Initially,

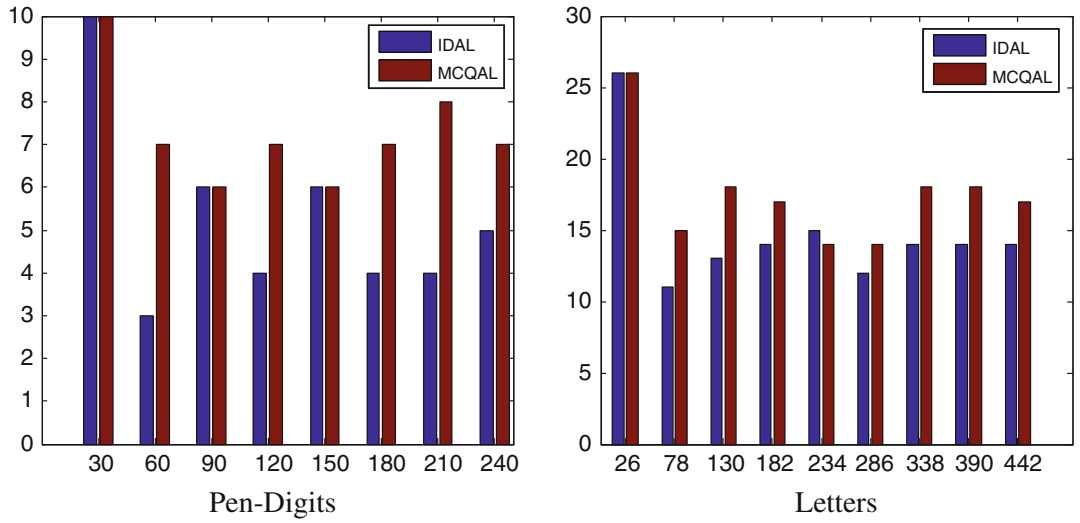



**Fig. 3** The class number of selected samples at each iteration

**Table 3** Computational time required by three techniques for all four data sets

| Data set | Train-set | Test-set | Features | MCQAL (s) | CAAL (s) | IDAL (s) |
|---|---|---|---|---|---|---|
| Ionosphere | 246 | 105 | 34 | **2.17** | 2.51 | 40.10 |
| Letters | 520 | 2080 | 16 | **12.57** | 19.57 | 44.19 |
| Pen-digits | 700 | 3498 | 16 | **18.40** | 73.58 | 200.86 |
| Balance-scale | 166 | 459 | 4 | 1.70 | 1.15 | 15.75 |

classification accuracy is much higher than the others as the selection of our approach is relatively more balanced. When the labeled samples reach a certain number, due to the high-valuable samples have been added to the training set, then the consideration of sample balance doesn't affect classification accuracy so obviously.

On the Pen-Digits and Letters datasets, each iteration we recorded the category number of selected samples. From the statistical data in the Fig. 3, in the condition of the same samples labeled at each iteration, the distribution of selected samples in our method is relatively more balanced. Reduce the selection redundancy while considering the diversity of samples, so the selected samples are more informative, quickly improving the classification performance.

The computational time required by the different techniques using the same experimental setting as described in the experiments. All the experiments were carried out on a PC (Pentium (R) Dual-Core CPU i5-2400@3.1 GHz, 4G RAM). Table 3 shows the computational time (in seconds) required by three techniques for all four data sets. From this table, compared with other two methods our method greatly reduced the running time in most cases.

## 5 Discussion and Conclusion

This paper presents a new active learning algorithm which combines uncertainty, representativeness, and diversity creation. On the basis of uncertainty sampling, we combined the measure of sample representativeness and analysis of sample diversity. This technique shortens the time required for training samples under the guarantee of classification accuracy. The labeling cost can be reduced without degrading the performance. For different data, our method needs to specify the size of the uncertain sample set according to the experiment. Therefore, according to the distribution of samples, how to dynamically determine the size of uncertainty set in order to ensure the optimal performance, is the focus of next study in the future.

**Acknowledgments** This work is partially supported by NSFC (No. 61003054, No. 61170020); College Natural Science Research project of Jiangsu Province (No. 10KJB520018); Science and Technology Support Program of Suzhou (No. SG201257); Science and Technology Support program of Jiangsu province (No. BE2012075); Open fund of Jiangsu Province Software Engineering R&D Center (SX201205). This work is also partially supported by the Natural Science Foundation of China under Grant No. 61003054 and 61170020, Jiangsu Province Colleges and Universities Natural Science Research Project under Grant No. 10KJB520018 and 13KJB520021, Jiangsu Province Science and Technology Support Program under Grant No. BE2012075, and Suzhou City Science and Technology Support Program under grant No. SG201257.

## References

1. Demir B, Minello L, Bruzzone L (2013) An effective strategy to reduce the labeling cost in the definition of training sets by active learning. IEEE Geoscience and Remote Sensing Letters 11(1):79–83
2. Settles Burr (2012) Synthesis lectures on artificial intelligence and machine learning. Act Learn 6(1):1–114
3. Settles B (2009) Active learning literature survey. Computer science technical report 1648, University of Wisconsin-Madison, USA, pp 3–4
4. Settles B (2010) Active learning literature survey. University of Wisconsin-Madison, USA
5. Fu Y, Zhu X, Li B (2013) A survey on instance selection for active learning. Knowl Info Syst 35(2):249–283
6. Patra S, Bruzzone L (2012) A batch-mode active learning technique based on multiple uncertainty for SVM classifier. IEEE Geosci Remote Sens Lett 9(3):497–501
7. Settles B, Craven M (2008) An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the conference on empirical methods in natural language processing. association for computational linguistics, pp 1070–1079
8. Li M, Sethi IK (2006) Confidence-based active learning. Pattern Anal Mach Intell IEEE Trans 28(8):1251–1261
9. Joshi AJ, Porikli F, Papanikolopoulos N (2009) Multi-class active learning for image classification. Comput Vis Pattern Recogn 2372–2379
10. Patra S, Bruzzone L (2011) A fast cluster-assumption based active-learning technique for classification of remote sensing images. Geosci Remote Sens IEEE Trans 49(5):1617–1626

11. Brinker K (2003) Incorporating diversity in active learning with support vector machines. ICML 3:59–66
12. Liu R, Wang Y, Baba T et al (2008) SVM-based active feedback in image retrieval using clustering and unlabeled data. Pattern Recogn 41(8):2645–2655
13. Demir B, Persello C, Bruzzone L (2011) Batch-mode active-learning methods for the interactive classification of remote sensing images. Geosci Remote Sens IEEE Trans 49(3):1014–1031
14. Lin H, Bilmes J (2009) How to select a good training-data subset for transcription: submodular active selection for sequences. Washington University Seattle Department of Electrical Engineering, Washington
15. Xu Z, Akella R, Zhang Y (2007) Incorporating diversity and density in active learning for relevance feedback. Advances in Information Retrieval. Springer, Berlin, Heidelberg, pp 246–257
16. Kapoor A, Grauman K, Urtasun R et al. (2007) Active learning with gaussian processes for object categorization. IEEE 11th international conference on computer vision, pp 1–8
17. Patra S, Bruzzone L (2012) A cluster-assumption based batch mode active learning technique. Pattern Recogn Lett 33(9):1042–1048

# A Multiple-Phase Stratification-Based Hierarchical Clustering Over a Deep Web Data Source

**Yuanliu Liu, Pengpeng Zhao, Xu Zhou and Zhiming Cui**

**Abstract** Compared with surface web, deep web stores more high-quality data, and data mining over deep web is more valuable. Nevertheless, in deep web, the entire data sets are stored in back-end databases and cannot be accessed directly, and data can only be retrieved over the Internet through query forms. The only particular method for mining a deep web data source is to sample the data set, which caused several unique challenges. In this paper, according to active learning, instead of traditional one-time sample allocation, we use multiple phases of sample allocation, which improves the representativeness of our gained samples. At the step of stratified sampling in each phase, we sample parts of representative samples for initial clustering. Using gained clusters, we can explore boundary points in them. A boundary point owns much uncertainty than others; for example, it contains more information. Sampling on a boundary point is useful to gain more representative samples. According to our experiments, our method performs better than random sampling and two-phase sampling in Liu and Agrawal (Int Conf Data Mining 70–81, 2012) at the same sampling costs.

**Keywords** Deep web · Active learning · Stratified sampling · Hierarchical clustering

## 1 Introduction

In recent years, deep web serves as a model of data dissemination and has become extremely popular. An early study conducted by Brightplanet in 2000 estimated that the public information in the deep web is 500 times larger than the surface

Y. Liu · P. Zhao (✉) · X. Zhou · Z. Cui
Department of Computer Science and Technology, Soochow University,
Suzhou 215006, China
e-mail: ppzhao@suda.edu.cn

Z. Wen and T. Li (eds.), *Foundations of Intelligent Systems*,
Advances in Intelligent Systems and Computing 277,
DOI: 10.1007/978-3-642-54924-3_37, 

web, approximately with 7,500 terabytes of data, across 200,000 deep web sites. UIUC conducted similar survey in 2004 and has shown that the scale of the deep web increased by 3–7 times between 2000 and 2004, and it was still growing.

The deep web has received much attention lately [2–8]. However, the problem about how many information contained in the deep web has not been resolved. It is desirable to obtain summary of key insights from one or more deep web data sources. Unfortunately, mining a deep web data source involves several unique challenges, which have not yet been adequately addressed. The back-end databases of the deep web cannot be accessed directly; it is not reasonable to gain whole database. Instead, the data can only be accessed through query interfaces, which are based on input attributes. As the queries are executed over a wide area network, acquiring data from deep web is time-consuming. Nearly 80 % of the execution time for deep web queries is spent on data delivery between the server and the clients [8]. The only particular way for mining a deep web data source is to sample the database. Thus, an efficient and effective sampling method is required.

## 2 Challenges and Overview of Our Strategy

This section summarizes the main challenges in clustering data in deep web [9–11] and presents our strategy in sample allocation.

As stated earlier, only submitting queries to the interface can retrieve a deep web data source. The key challenge in mining a deep web data source arises because the back-end databases cannot be accessed directly. It is difficult to discover clusters on output attributes since the distribution is unknown. A naive solution for this problem will be to run a simple random walk on the deep web [12–14] and then apply a traditional clustering algorithm, such as k-means algorithm [15] or a hierarchical clustering algorithm [16]. However, a simple random walk may not obtain appropriate samples to represent the entire population when the sample size is small; nonetheless, obtaining large samples can be extremely expensive.

Liu et al. [17, 18, 19] proposed stratified sampling and have achieved a good experimental result. Stratified sampling picks separate samples from H groups, which are also called strata or subpopulations. Stratification is performed on the query interface based on random samples, and the stratification will build a query tree named hierarchical tree. Accordingly, using this tree to obtain representative samples will have some uncertainty if the initial samples are not representative enough.

Thus, we draw on the idea of active learning [20], using multiple phases of sample allocation strategy to replace the traditional one-time allocation strategy, with representative samples gained at each phase added to the initial random sample to build hierarchical tree. This strategy will benefit stratified sampling in the following phase to obtain more representative samples. After building the hierarchical tree, we use Neymann allocation to determine the sample size at each stratum, and then, some representative samples can be used to execute the initial stratification-based

hierarchical clustering. Points near the boundary of clusters have large uncertainty. Resampling in boundary points can help to improve the representativeness of the samples collected, which in turn benefits the accuracy of clustering.

# 3 Multiple-Phase Stratification-Based Hierarchical Clustering

In active learning, representativeness and uncertainty are two important indicators to measure a sample. Sample with higher representativeness can help to improve the precision of clustering. When a sample has greater uncertainty, indicating that the sample contains richer information can improve the precision of clustering. This paper draws on the idea of active learning and proposed multiple phases of sample allocation strategy. In our method, the representative samples can be enhanced through several phases; then, a better hierarchical tree is generated. The probability of collecting representative samples will be increased using updated tree, so that the clustering accuracy will be improved.

We assume $N$ samples need to be sampled, and $\beta$ is a regulatory factor that represents how many phases should be executed. Then, at each phase, $N/\beta$ samples need to be achieved from the deep web data source. Here, a phase contains four steps: (1) stratifying input attributes, (2) performing sample allocation and sampling, (3) initial clustering, and (4) resampling in boundary points. When a phase finishes, the obtained representative samples will be added to the initial random sample to build a better hierarchical tree. When all phases are finished, perform stratification-based hierarchical clustering on entire obtained samples.

## *3.1 Representative Sampling*

In the context of deep web, the distribution of the output attributes is generally considered to be the statistical variables. Therefore, when the mean values of output attributes about a sample are close to their true values in real environment, the sample can be considered as a representative sample. Since the back-end database is not directly accessible, the true mean values of the set of output attributes are unknown. Thus, our goal is to find a good estimation of the mean values for the output attributes.

For an output attribute $O_j \in OS$, let the expression $x_j$ denotes the value of the output attribute $O_j$, and $\bar{x}_{j,i}$ denote the sample average of $O_j$ for the $i^{\text{th}}$ stratum. Then, the mean value of the output attribute can be estimated as follows:

$$\bar{x}_j = \sum_{i=1}^{H} \frac{N_i}{N} \bar{x}_{j,i} \tag{1}$$

where $N$ denotes the size of the entire population and $N_{\mathrm{i}}$ denotes the size of the sub-population in the $i^{\mathrm{th}}$ stratum. For a deep web data source, the values of $N$ and $N_{\mathrm{i}}, i = 1, 2$ are normally available.

In stratified sampling, variance will be minimized when sample size for the $i$th stratum is proportional both to the size of the stratum and to the variance of the target values in the $i$th stratum. In the context of the deep web, the target values are the value of output attributes. At each phase, for a particular output attribute $O_j \in OS$, if we want to draw n data records across the strata, in order to minimize the variance of the estimated mean value, the sample allocation for the $i$th stratum $n_i$ is computed as follows:

$$n_i = \frac{n}{\sum_r N_r \sigma_r} N_i \sigma_i \tag{2}$$

where $\sigma_r^2$ is the variance for the output attribute $O_j$ in the $r$th stratum; when performing sample allocation, we need to consider the entire set of output attributes $OS$ as a whole. Let $\mathrm{Var}(\bar{x}_j)$ denote the variance for estimated mean value; then, for the set of output attributes $OS$, the summation of variance for estimated mean values is computed as $\mathrm{Var}(OS) = \sum_j \mathrm{Var}(\bar{x}_j)$. In order to minimize the variance for the output attributes in $OS$, the sample allocation is computed as

$$n_i = \frac{n}{\sum_r N_r \sigma'_r} N_i \sigma'_i \tag{3}$$

where $\sigma'_r = \sqrt{\sum_j^n \delta_{j,r}^2}$ represents the square root of the summation of $\delta_{j,r}^2$, which represents the variance of output attribute $O_j \in OS$ in $r$th stratum. As stated above, this method tends to allocate more samples to the stratum where the variance for the output attribute is large, so that the integrated variance is minimized.

## 3.2 Stratification-Based Hierarchical Clustering

We choose stratification-based hierarchical clustering [1] to do our clustering work in this paper. Similar to traditional hierarchical clustering, two clusters with the minimum distance are merged into one cluster at each step. The process continues until there are $k$ clusters left, where $k$ is the predefined number of clusters. The distance between two data records is computed as the Euclidean distance on the output attributes. For two data records $D_1 \in NS, D_2 \in NS$ from the deep web data source with the set of output attributes $OS$, the distance is computed as

$$\mathrm{Dist}(D_1, D_2) = \sqrt{\sum_j \left(O_{1,j} - O_{2,j}\right)^2} \tag{4}$$

Different from traditional method, the distance between two clusters is the average of the weighted distances between all pairs of two data records $D_i \in C_1$, $D_j \in C_2$ in stratification-based hierarchical clustering, and $C_i$ represents a cluster:

$$\mathrm{Dist}(C_1, C_2) = \frac{\sum_i \sum_j w_i \times w_j \times \mathrm{Dist}\left(D_i, D_j\right)}{\sum_i \sum_j w_i \times w_j} \tag{5}$$

for $i$th cluster $C_i$; the center vector is computed as

$$\bar{m}_i = \frac{\sum_r w_r \times o_r}{\sum_r w_r}, \tag{6}$$

and $o_r$ corresponds to the vector of output attributes for data record $D_r \in C_i$. The associated radius $R_i$ for cluster $C_i$ is estimated as

$$\overline{R}_i = \frac{\sum_r w_r \times \mathrm{Dist}(o_r, \overline{m}_i)}{\sum_r w_r} \tag{7}$$

## 3.3 Resampling in Boundary Points

When a sample has greater uncertainty, indicating that the sample contains richer information can effectively improve the precision of clustering. Compared with passive learning, which selects training data randomly from the entire population, active learning selects certain types of data records, to help build a better model faster. Using active learning is beneficial to reduce the sample cost and improve the sample quality.

Specifically, the uncertainty here is with respect to the possibility for a data point belonging to one particular cluster. A data point has a high uncertainty if it is far from the center of a cluster or it is between two clusters. Performing clustering on representative samples obtained in first step is helpful to find boundary points, and resampling in boundary points can help to improve the representativeness of the samples collected, which in turn benefit the accuracy of clustering.

For a data point $p$, which belongs to cluster $i$, $d(p, m_i)$ is the smallest of distances to all cluster centers. A data point $p$ is considered to be a boundary data point if there exists cluster $l, l \neq i$, so that

$$0 \leq d(p, m_i) - d(p, m_l) \leq \left(\overline{R}_i + \overline{R}_l\right) \times \theta$$

where $\theta$ is a predefined parameter and $\overline{R}_i, \overline{R}_l$ are the radii of clusters $i$ and $l$.

For a stratum associated with the query space $Q_t \in QS$, the probability of containing boundary data points is $U_t = \frac{n_{t,b}}{n_t}$ and $n_{t,b}$ denotes the size of sample in the stratum. Both of the values can be estimated after initial clustering. Formally, the size we should sample in the stratum contains boundary points and can be computed as follows and $n$ represents the size of sample to be drawn in this step:

$$n'_t = \frac{p(Q_t) \times U_t}{\sum_r p(Q_r) \times U_r} \times n \tag{8}$$

### 3.4 Overall Algorithm

In this section, we will summarize our overall algorithm, which can be seen in Table 1. The input parameters of this algorithm comprise the following: the set of input attributes *IS*, the set of output attributes *OS,* and the set of potential attributes *PS*. The initial sample set is denoted by *SS* at each phase. Stratifying the query space (Line 5) and gaining representative samples (Line 6–10) are referring to the method mentioned in [1]. Stratification-based hierarchical clustering (Line 11) and resampling on boundary points (Line 12–17) are performed after them. While all phases are finished, we execute stratification-based hierarchical clustering on gained samples, and then, k centers can be computed.

## 4 Evaluation Study

This section presents experimental results using our multiple-phase stratification-based hierarchical clustering method. We evaluate three methods in this paper, which are the proposed method of ours, the idea mentioned in [1] by Tantan Liu, and the simple random sampling using traditional hierarchical clustering method. Throughout this section, our experiment has been conducted using a synthetic data set and a real data set crawled from Yahoo.

### 4.1 Design of Experiments

As follows, two data sets are chosen to do our experiments:

Synthetic data set: This data set is generated by Minitab. This data set contains 4,000 data records with five input attributes and two output attributes. There are four clusters on the two output attributes, which are generated by Gaussian distribution. Specially, the output attributes are generated to be dependent on the input attributes.

**Table 1** Multiple-phase stratification-based hierarchical clustering

| Multiphase Clustering $(IS, OS, PS, n_u, n_v, k, \beta)$ |
|---|
| 1: $IRS \leftarrow$ initial random sample |
| 2: $SS \leftarrow IRS$ |
| 3: for $i = 0; i < \beta; i++$ do |
| 4: $LF \leftarrow NULL$ |
| 5: $TreeBuild(IS, OS, SS, PS, LF)$ |
| 6: for all $l_i \in LF$ do |
| 7: $n_i = \frac{n_u/\beta}{\sum_r N_r \delta_r'} N_i \sigma_i'$ |
| 8: $S_i \leftarrow Sampling$ $n_i$ *data records in* $l_i$ |
| 9: $SS \leftarrow SS \cup S_i$ |
| 10: end for |
| 11: $CS \leftarrow StratifiedClustering(SS, k)$ |
| 12: for all $l_i \in LF$ do |
| 13: $Q_i \leftarrow$ query corresponding to $l_i$ |
| 14: $n\prime_t = \frac{p(Q_t) \times U_t}{\sum_r p(Q_r) \times U_r} \times \frac{n_v}{\beta}$ |
| 15: $S_i \leftarrow$ Sampling $n_t'$ data records in $l_i$ |
| 16: $SS \leftarrow SS \cup S_i$ |
| 17: end for |
| 18: end for |
| 19: $CS \leftarrow StratifiedClustering(SS, k)$ |
| 20: return $CE \leftarrow$ *center matrix of* $CS$ |

Yahoo data set: This data set is crawled from a real-world hidden database at http://autos.yahoo.com/. In this paper, we download 8,000 data records on used cars located within 50 miles of a zipcode address. The data record consists of four categorical input attributes and 1 numerical output attribute. The input attributes contain the age, mileage, brand, and the number of windows of the cars. And two clusters are assigned on the output attributes.

The following two criteria have been chosen to evaluate all methods:

(1) Average Distance: It is computed based on the distance between the estimated centers and true centers of the clusters. For k clusters with centers $m_1, \ldots, m_k$, the corresponding radius $R_1, \ldots, R_k$, and the estimated centers denoted by $\overline{m_1}, \ldots, \overline{m_k}$, then the average distance can be computed as $A\,\text{Dist} = \frac{1}{k} \sum_i \frac{|m_i - \overline{m_i}|}{R_i}$. Obviously, a smaller average distance value indicates a higher accuracy rate.

(2) Precision: It is the percentage of the real clusters correctly identified by the methods. An identified cluster is considered to be correct if the distance between the estimated center and the true center is within 10 % of the radius.

We compared three methods in different conditions; while reporting the results from our experiments, our algorithm is referred to MS and the method proposed in [1] is referred to TS. The clustering directly conducted on the simple random

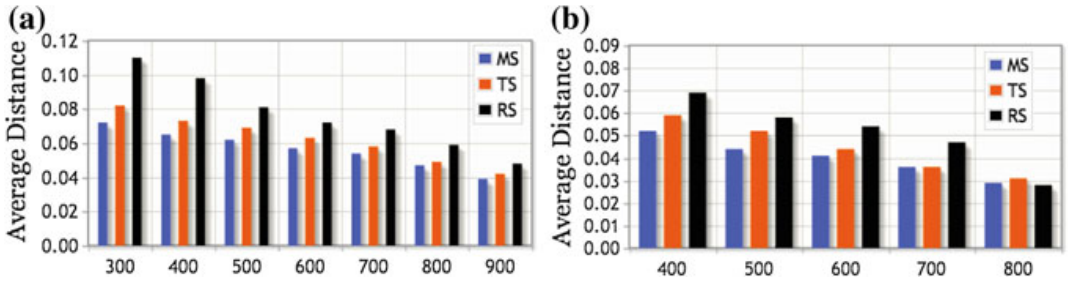



**Fig. 1** Evaluations on MS, TS, and RS using average distance; subfigures (**a**) and (**b**) represent our two different data sets. **a** Synthetic data set. **b** Yahoo data set

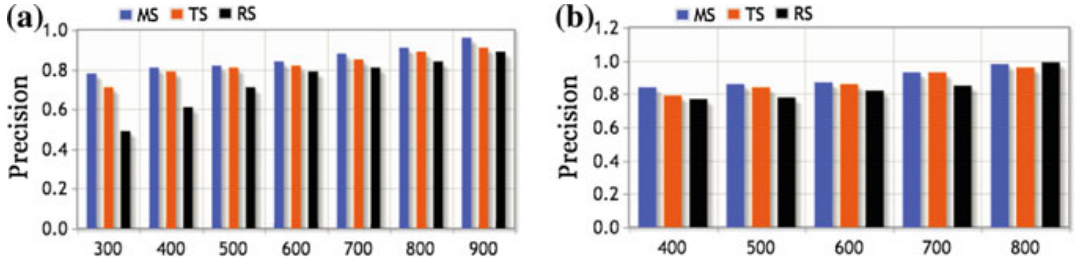



**Fig. 2** Evaluations on MS, TS, and RS using precision; subfigures (**a**) and (**b**) represent our two different data sets. **a** Synthetic data set. **b** Yahoo data set

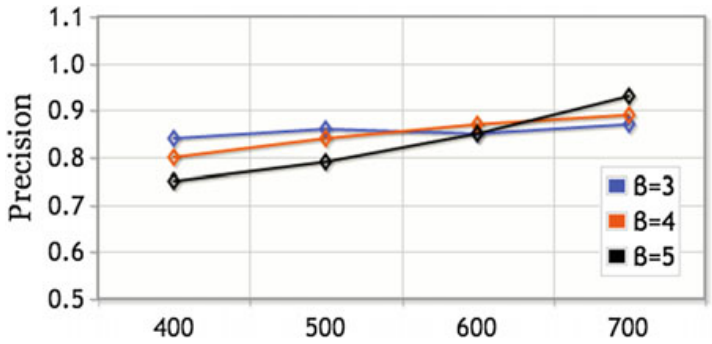



**Fig. 3** The influence on precision using different $\beta$; this experimental result is performed on our synthetic data set

sample applying the original hierarchical clustering method is referred to as RS. Figure 1a and b compare the average distance of clustering using presented three methods. In the figure, the $x$-axis represents the size of sample $s$. The size of pilot sample used in our method and in [1] is $s/2$. Figure 2a and b compare the precision of clustering using three methods.

As stated above, $\beta$ represents how many phases we should execute. The value of the precision is shown in Fig. 3 when we assign different values of $\beta$. The x-axis represents the size of sample s, and the different lines represent different values of $\beta$.

### 4.2 Experimental Results

Overall, we can make the following observation from our experiments. In order to achieve the same accuracy level, our method requires 25, 19 % fewer sampled data records over the two-phase strategy mentioned in [1], for the Synthetic and Yahoo data sets. And our multiple phases are flexible; it depends on the sample size we need to gain. As shown in Fig. 3, the sample size is small; we should turn down the value of $\beta$. Obviously, if the sample size is large, using a big value of $\beta$ can achieve better performance. Our experiments reflect significant reductions in mining cost, since the cost of obtaining each sample is lower.

## 5 Conclusions

In the past years, our research team has developed a lot of techniques in the area of deep web mining, and these works mainly focus on deep web query interface integration and deep web data extraction. A number of recent efforts have also been building deep web querying systems, trying to provide mediator-like support. However, given the volume of information contained in the deep web, it is desirable to obtain summary or key insights from one or more deep web data sources. Unfortunately, mining a deep web data source involves several unique challenges, which have not yet been adequately addressed. Tantan Liu proposed stratification-based hierarchical clustering over a deep web data source, and the experiment shows her method has a good performance. In this paper, according to active learning, instead of traditional one-time sample allocation, we gain samples in multiple phases, which improve the representation of our gained samples; then, the precision of clustering correspondingly improved.

We will apply our method to other deep web mining tasks, such as the frequent items and association rules, and further explore the current single-source data mining methods to a multi-source migration.

**Acknowledgments** This work is partially supported by NSFC (No. 61003054, No. 61170020); College Natural Science Research project of Jiangsu Province (No. 10KJB520018); Science and Technology Support Program of Suzhou (No. SG201257); Science and Technology Support program of Jiangsu province (No. BE2012075); and Open fund of Jiangsu Province Software Engineering R&D Center (SX201205).

## References

1. Tantan Liu, Gagan Agrawal (2012) Stratification based hierarchical clustering over a deep web data source. Int Conf Data Min, pp 70–81
2. Braga D, Ceri S, Daniel F, Martinenghi D (2008) Optimization of multi-domain queries on the web. VLDB endowment, 1:562–673

3. Cali A, Martinenghi D (2008) Querying data under access limitations. In: Proceedings of the 24th international conference on data engineering, pp 50–59
4. He H, Meng W, Yu C, Wu Z (2004) Automatic integration of web search interfaces with wise integrator. Int J Very Large Data Bases 12:256–273
5. Madhavan J, Afanasiev L, Antova L, Halevy A (2009) Harnessing the deep web: present and future. In: 4th biennial conference on innovative data systems research (CIDR)
6. Madhavan J, Ko D, Kot L, Ganapathy V, Rasmussen A, Halevy A (2008) Google's deep web crawl. VLDB Endowment, 1:1241–1252
7. Srivastava U, Munagala K, Widom J, Motwani R (2006) Query optimization over web services. In: Proceedings of the 32nd VLDB, pp 355–366
8. Wang F, Agrawal G, Jin R, Piontkivska H (2007) Snpminer: a domain-specific deep web mining tool. In: Proceedings of the 7th IEEE international conference on bioinformatics and bioengineering, pp 192–199
9. Jain Anil K, Dubes Richard C (1988) Algorithms for clustering data. Prentice-Hall Inc, Upper Saddle River
10. Zhang T, Ramakrishnan R, Birch M (1996) An efficient data clustering method for very large databases. ACM SIGMOD Rec, 25(2):103
11. McLachlan GJ, Basford KE (1988) Mixture models: inference and applications to clustering. Marcel Dekker, New York
12. Bar-Yossef Z, Gurevich M (2008) Mining search engine query logs via suggestion sampling. Proc VLDB Endow 1(1):54–65
13. Dasgupta A, Das G, Mannila H (2007) A random walk approach to sampling hidden databases. In: Proceedings of the 2007 ACM SIGMOD international conference on management of data (SIGMOD' 07), pp 629–640
14. Dasgupta A, Zhang N, Das G (2009) Leveraging count information in sampling hidden databases. In: Proceedings of the 2009 IEEE international conference on data engineering (ICDE' 09), pp 329–340
15. Hartigan JA, Wong MA (1979) A k-means clustering algorithm. Appl Stat 28:100–108
16. Ward JH Jr (1963) Hierarchical grouping to optimize an objective function. J Am Stat Assoc, 58:236–244
17. Liu T, Wang F, Agrawal G (2012) Stratified sampling for data mining on the deep web. Frontiers Comput Sci 6(2):179–196
18. Liu T, Agrawal G (2012) Stratified k-means clustering over a deep web data source. Knowl Disc Data Min, pp 1113–1121
19. Liu T, Agrawal G (2012, August). Stratified k-means clustering over a deep web data source. In: Proceedings of the 18th ACM SIGKDD international conference on knowledge discovery and data mining pp 1113–1121
20. Liu T, Agrawal G (2011) Active learning based frequent itemset mining over the deep web. International conference on data engineering, pp 219–230

# An Evolution-Based Robust Social Influence Evaluation Method in Online Social Networks

Feng Zhu[1], Guanfeng Liu[1], An Liu[1], Lei Zhao[1], and Xiaofang Zhou[1,2]

[1] School of Computer Science and Technology,
Jiangsu Provincial Key Laboratory for Computer Information Processing Technology,
Soochow University, Suzhou, China, 215006
[2] School of Information Technology and Electrical Engineering,
University of Queensland, Brisbane, Australia, 4072
{gfliu,zhaol,zxf}@suda.edu.cn

**Abstract.** Online Social Networks (OSNs) are becoming popular and attracting lots of participants. In OSN based e-commerce platforms, a buyer's review of a product is one of the most important factors for other buyers' decision makings. A buyer who provides high quality reviews thus has strong social influence, and can impact a large number of participants' purchase behaviours in OSNs. However, the dishonest participants can cheat the existing social influence evaluation models by using some typical attacks, like *Constant* and *Camouflage*, to obtain fake strong social influence. Therefore, it is significant to accurately evaluate such social influence to recommend the participants who have strong social influences and provide high quality product reviews. In this paper, we propose an Evolutionary-Based Robust Social Influence (EB-RSI) method based on the trust evolutionary models. In our EB-RSI, we propose four influence impact factors in social influence evaluation, i.e., Total Trustworthiness (TT), Fluctuant Trend of Being Advisor (FTBA), Fluctuant Trend of Trustworthiness (FTT) and Trustworthiness Area (TA). They are all significant in the influence evaluation. We conduct experiments onto a real social network dataset Epinions, and validate the effectiveness and robustness of our EB-RSI by comparing with state-of-the-art method, SoCap. The experimental results demonstrate that our EB-RSI can more accurately evaluate participants' social influence than SoCap.

**Keywords:** Social influence, trust, influence evaluation, social network.

## 1 Introduction

### 1.1 Background

In recent years, Online Social Networks (OSNs), like Facebook and Twitter have attracted lots of participants, where they can make new friends and share their experience. In a social network based e-commence platform, like Epinions (epinions.com), each participant can be a buyer or a seller. After a transaction, a buyer can write a product review and give 1 to 5 stars as the ratings for different aspects of the product, such as *Ease of Use*, *Customer Service*, *On-Time*

B. Benatallah et al. (Eds.): WISE 2014, Part II, LNCS 8787, pp. 141–157, 2014.

*Delivery*, etc. Then, when other buyers want to buy the same product from the same seller, they can view that product review and make the decision based on the review. Based on their own transaction experiences, these buyers can rate the reviews as *Not Helpful*, *Somewhat Helpful*, *Helpful*, and *Very Helpful* [14]. If a buyer usually provides *Very Helpful* product reviews, he/she can be trusted by other buyers. As indicated in *Social Psychology* [4,10,24] and *Computer Science* [3], a buyer prefers the recommendation from his/her trusted buyers over those from others. It means that a buyer is more likely to make a purchase decision based on the product reviews given by the trustworthy buyers. Then these trustworthy buyers have strong social influence that can affect others purchase behaviours in OSNs. These buyers are called *advisors* of those participants who trust their product reviews.

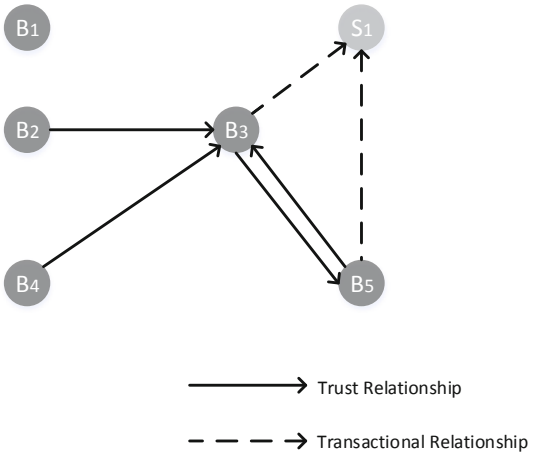



**Fig. 1.** A social network based e-commence platform

**Example 1:** Fig.1 depicts a social network based e-commerce platform which contains a seller (i.e., $S_1$) and five buyers (i.e., $B_1$ to $B_5$). $B_1$ is a new registered buyer, so there is no trust relationship between $B_1$ and any other buyers. $B_3$ is an advisor of $B_2$, $B_4$ and $B_5$, so there exist three trust relationships (displayed as a *solid line* in Fig.1) between $B_3$ and these three buyers. $B_3$ and $B_5$ have transactions with $S_1$, so there exist two transactional relationships (displayed as a *dashed line* in Fig.1) between $S_1$ and the two buyers. If $B_2$ wants to buy the product from $S_1$ and $B_3$ provides the reviews of that product, then $B_2$ will make the decision based on $B_3$'s reviews.

### 1.2 The Problem

Social network based e-commerce platform is in an open environment, where anybody can sign up to become a buyer and write reviews. Moreover, anybody

can give the *Helpfulness* of reviews making this product review scheme highly vulnerable to be attacked [14]. A dishonest advisor can cheat the product review system and obtain strong social influence via some typic attacks, like *Constant*[1], *Camouflage*[2], *Whitewashing*[3] and *Sybil*[4] [13]. These dishonest advisors who have strong social influence can harm the benefits of both buyers and sellers. Therefore, it is necessary and significant to develop a robust social influence evaluation method which can accurately evaluate the social influence of participants and defend against these typical attacks.

**Example 2:** Here we take *Camouflage* attack as a example. Recall Fig.1, suppose $B_5$ is a dishonest advisor. At first, $B_5$ provides high quality reviews to accumulate strong social influence. Then, $B_3$ conspires with $S_1$ and writes a very good review for the products that are sold by $S_1$ and have low quality. Then $B_2$ and $B_4$ will be cheated to buy the low quality products from $S_1$.

In the literature, various social influence methods [2, 5, 8, 9, 23, 25] have been proposed to compute the value of participants' social influences in OSNs, these methods study influence maximization under the popular independent cascade (IC) model [16] and evaluate social influence through the process of information diffusion [18]. These methods mainly focus on the current network status and ignore the trend of participants' historical influence. However, as indicated in [20], an accurate social influence evaluation method requires more influence information that not only the current influence level, but also the influence prediction relevant to forthcoming transactions. Thus, although a participant computed by the existing methods with strong social influence in the current OSN, if the trend of his/her influence is downward, the influence of that participant is more likely to decrease in the near future. In addition, the existing methods do not consider the occurrence of the above mentioned typical attacks from dishonest participants. Therefore, they cannot defend against these attacks to deliver accurate social influence evaluation results.

### 1.3 Contributions

In order to deliver accurate social influence evaluation results, in this paper, we first propose four influence impact factors, i.e., Total Trustworthiness (TT), Fluctuant Trend of Being Advisor (FTBA), Fluctuant Trend of Trustworthiness (FTT) and Trustworthiness Area (TA). These four factors are significant in social influence evaluation. We then propose a trust evolutionary model based on the Multiagent Evolutionary Trust (MET) model [13], and propose a novel Evolutionary-Based Robust Social Influence (EB-RSI) method based on the trust

[1] Dishonest advisors constantly provide unfairly positive/negative ratings to sellers.
[2] Dishonest advisors camouflage themselves as honest advisors by providing fair ratings to build up their trustworthiness first and then gives unfair ratings.
[3] A dishonest advisor is able to whitewash its low trustworthiness by starting a new account with the initial trustworthiness value.
[4] A dishonest buyer creates several accounts to constantly provide unfair ratings to sellers.

evolutionary models and our proposed four impact factors. We conduct experiments onto a real social network dataset, Epinions (epinions.com), and compare our EB-RSI method with the state-of-the-art social influence method, called SoCap [23]. The experimental results illustrate that our EB-RSI method can defend against the typical attacks and deliver more accurate social influence evaluation results than SoCap.

This paper is organised as follows. *Section 2* discusses the related work. *Section 3* introduces the preliminaries. *Section 4* proposes the four influence impact factors. *Section 5* proposes our EB-RSI method. In *Section 6*, we verify the effectiveness and robustness of our EB-RSI by comparing with the state-of-the-art method, SoCap. *Section 7* is the conclusions.

## 2 Related Work

In the literature, many social influence evaluation approaches [2, 5–7, 11, 15–17, 19, 23] have been proposed to compute the value of participants' social influence in OSNs. Most of these approaches [5–7, 15–17, 19] attempt to model social influence through the process of information diffusion [18]. In addition, the problem of finding influencers is often studied as an influence maximization problem which is to find the *Top-K* ($K$ seed) nodes such that the value of influence is maximized. Kempe et al. [16] consider that the problem of finding a subset of influential nodes is an absolute optimization problem and indicate that influence maximization problem is NP-hard. They propose a greedy algorithm which guarantees $(1 - 1/e)$ approximation ratio. However, this algorithm has low efficiency in practice and thus it is not scalable with the network size. So, the followers devote themselves to renovate the algorithm to spend up the process of computing the influence value or improve the influence propagation model to adapt to the network proliferation. In order to improve the scalability, Chen et al. [6] propose an algorithm, which has a simple tunable parameter, for users to control the balance between the running time and the influence spread of the algorithm. Nevertheless, a single influence evaluation itself is #P-hard, which is also hard to be solved in polynomial time. Jung et al. [15] propose a novel algorithm IRIE that integrates the advantages of influence ranking (IR) and influence estimation (IE) methods for influence maximization. Kim et al. [17] provide a scalable influence approximation algorithm, Independent Path Algorithm (IPA) for IC model. In the model, they study IPA efficiently approximates influence by considering an independent influence path as an influence evaluation unit and it is also easily parallelized by adding a few lines expressions. Furthermore, in order to spend up the evaluation algorithm, Leskovec et al. [19] develop the CELF algorithm, which exploits submodularity to find near-optimal influencer selections, namely the obtained solutions are guaranteed to achieve at least a fraction of $\frac{1}{2}(1 - 1/e)$ of the optimal solution. However, the above methods do not consider the historical data of influence, so they cannot provide influence trend prediction about the forthcoming transactions. In addition, social network is in an open environment, there might be some unreliable reviews given by dishonest participants. But the

above methods do not adopt any strategies to identify those unreliable reviews and dishonest participants.

In addition, since the above methods capture only the process of information diffusion and not the actual social value of collaborations in the network. Subbian et al. [23] propose an approach, called SoCap, to find influencers in OSN by using the social capital value. They model the problem of finding influencers in OSN as a value-allocation problem, where the allocated value denotes the individual social capital. However, as ignoring the trend of influence, this method always cannot find high quality influencers who can keep their influences for a long term. In addition, this method is also vulnerable to the unfair rating attacks from dishonest participants. For the problem of unfair rating attacks, there are some models [1, 22] have been proposed to detect the fraudsters and fake reviews in OSNs. These models mainly focus on finding the fraudsters rather than defending against the attacks. Moreover, they did not consider the social influence evaluation methods by fraud detecting.

Furthermore, Yeung et al. [2] have studied the relations between trust and product ratings in online consumer review sites. And they propose a method to estimate the strengths of trust relations so as to estimate the true influence among the trusted users. This kind of strength cannot reflects the whole social influence of this user, because it represents just the local influence among the trusted users and excludes the whole influence effected in all users. In addition, Franks et al. [11] propose a method to identify influential agents in open Multi-Agent systems without centralised control and individuals have equal authority. They find out four single metrics are robustly indicative of influence, and they study which single metric or combined measure is the best predictor of influence in a given network. However, these single metrics need a massive computing time, and extra computing will be used to find out the best predictor of influence. So, this method is also not scalable with the network size.

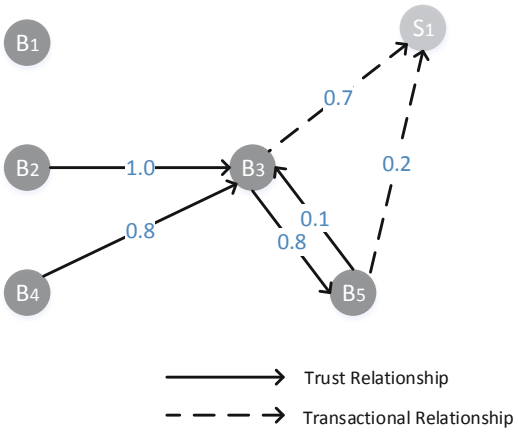



**Fig. 2.** A trust network with ratings

## 3 Preliminary

### 3.1 Social Network

In this paper, a social network is modeled as a directed graph $G = (V, E)$, where $V$ is the set of vertices and $E$ is the set of edges. Each vertex in $V$ represents a buyer or a seller, and each edge in $E$ represents a trust relationship between buyers or a transaction relationship between a buyer and a seller.

### 3.2 Trust Relationship and Transaction Relationship

We use $T$ and $R$ to denote trust relationships and transaction relationships respectively, where $T(B_i, B_j) \in [0, 1]$ represents buyer $B_i$'s trust value towards advisor $B_j$, and $R(B_i, S_j) \in [0, 1]$ represents a rating value provided by buyer $B_i$ to seller $S_j$. In addition, if $B_i$ has no experience with $S_j$, it is usually set the missing rating value $R(B_i, S_j)$ to 0.5 as a neutral value [13].

**Example 3:** Fig.2 depicts an OSN which contains four trust relationships, they are $T(B_2, B_3) = 1.0$, $T(B_4, B_3) = 0.8$, $T(B_5, B_3) = 0.1$ and $T(B_3, B_5) = 0.8$, and two transaction relationships, they are $R(B_3, S_1) = 0.7$ and $R(B_5, S_1) = 0.2$.

### 3.3 Evolutionary Trust Model

The Evolutionary Trust Model [13] is usually used to cope with possible unfair attacks from dishonest advisors. If a buyer has some dissimilar advisors, whose reviews are very different with the buyer's purchase experience. The buyer can evolve its trust relationships to absorb the advisors whose reviews are match the buyer's purchase experience and remove the dissimilar advisors. The evolutionary process have been detailed discussed in [13].

**Example 4:** In Fig.3, there is a trust relationship $T(B_2, B_3)$, and $B_3$ provides a good review of the product sold by $S_1$ ($R(B_3, S_1) = 0.7$). Suppose $B_2$ experienced a new transaction with $S_1$, and find the product sold by $S_1$ is not so good as described in $B_3$'s review ($R(B_2, S_1) = 0.2$), $B_2$ will evolve his/her trust

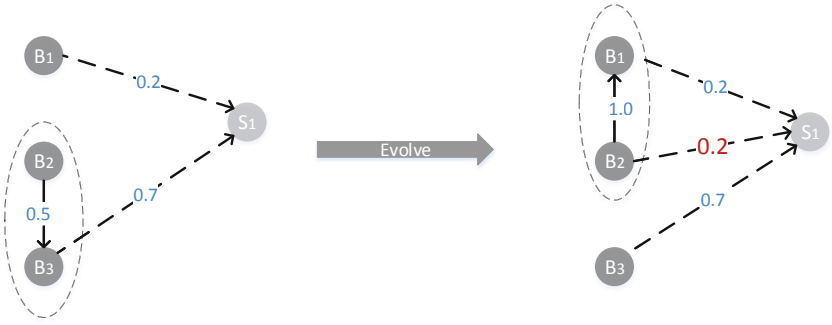



**Fig. 3.** Evolutionary process

relationships. As $B_1$ writes the review that contents match the experience of $B_2$, $B_2$ builds a new trust relationship with $B_1$ (i.e., $T(B_2, B_1) = 1.0$) and remove the trust relationship with $B_3$. Namely $B_1$ is added as a new advisor into the advisor team of $B_2$.

## 4 Influence Impact Factors

As indicated in [20], it is significant to take the current influence level and influence prediction into social influence evaluation. In our method, we propose four important influence impact factors, which are significant in delivering accurate social influence evaluation results.

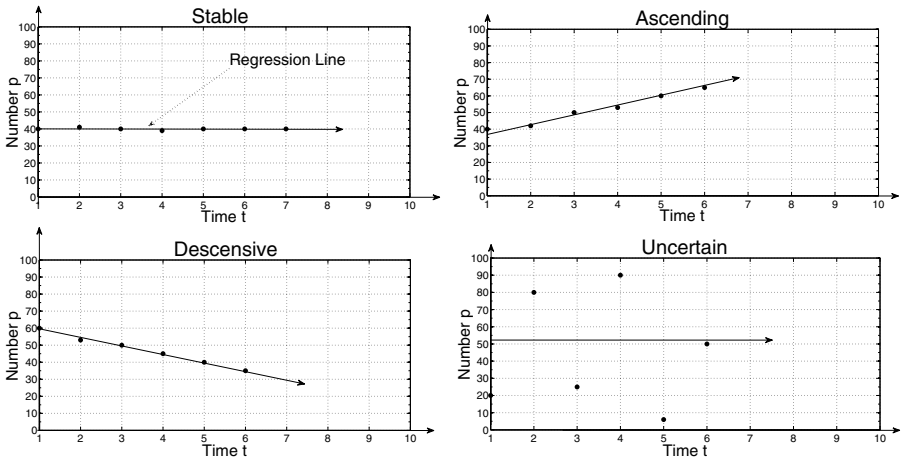



**Fig. 4.** Typical cases of FTBA

### 4.1 Total Trustworthiness (TT)

It shows the total trust value of an advisor given by other buyers. We use $TT(B_i, t_j)$ to denote $B_i$'s total trust value at time $t_j$, and $TT(B_i, t_j)$ is calculated as

$$TT(B_i, t_j) = \sum_{k=1}^{p} T(B_k, B_i) \tag{1}$$

where $p$ is the number of $B_i$'s in-degree.

### 4.2 Fluctuant Trend of Being Advisor (FTBA)

FTBA is used to illustrate the fluctuant trend of being advisor in a certain period. Some typical cases of FTBA, that are "stable", "ascending", "descensive" and "uncertain", are depicted in Fig. 4, where X axis is time and the Y axis is

the number of being advisor. We use a regression line to model FTBA, and the regression line's gradient (denoted as $grad_1$) and mean distance (denoted as $md_1$) can measure FTBA well [20]. The regression line is based on the least squares fit.

We denote the number of $B_i$'s in-degree at time $t_j$ as $N_i(B_i, t_j)$, and use $(t_s, N_i(B_i, t_s)), (t_2, N_i(B_i, t_2)), (t_3, N_i(B_i, t_3)), ..., (t_e, N_i(B_i, t_e))$ denote the given data points of $B_i$'s in-degree number from $t_s$ to $t_e$. Here and in the following of this paper, $t_s$ is the start time of the historical transactions and $t_e$ is the end time of that transactions. Then the regression line can be represented as follows:

$$y = kt + b \tag{2}$$

where $k$ and $b$ are constants to be determined, and $k$ represents the $grad_1$ value. As the distance from point $(t_j, N_i(B_i, t_j))$ to the regression line is

$$d(B_i, t_j) = \frac{N_i(B_i, t_j) - b - kt_j}{\sqrt{1 + k^2}}. \tag{3}$$

Based on the theory of least squares, the sum of squares of the distance can be calculated as follows:

$$S(B_i, p) = \sum_{j=1}^{p} d^2(B_i, t_j) = \sum_{j=1}^{p} \frac{(N_i(B_i, t_j) - b - kt_j)^2}{1 + k^2}. \tag{4}$$

Next our main task is to minimise the sum of squares of the distance $S(i, p)$ with respect to the parameters $k$ and $b$, with the method of undetermined coefficients.

Since function $S(B_i, p)$ is continuous and differentiable, as we known, based on method of two variables' function extremum, the minimization point of $S(B_i, p)$ makes the first derivative of function $S(B_i, p)$ be zero, and the second derivative positive, which could be easily proved by Taylor formula for function of two variables [21]. For this, we differentiate $S(B_i, p)$ with respect to $k$ and $b$, and set the results to zero, then we can get following results:

$$k = grad_1(B_i) = (-u - \sqrt{u^2 + 4})/2. \tag{5}$$

and

$$b = \frac{S_f - kS_t}{n}, \tag{6}$$

where $S_{f2} = \sum_{j=1}^{p} N_i^2(B_i, t_j)$, $S_f = \sum_{j=1}^{p} N_i(B_i, t_j)$, $S_t = \sum_{j=1}^{p} t_j$, $S_{t2} = \sum_{j=1}^{p} t_j^2$ and $S_{ft} = \sum_{j=1}^{p} N_i(B_i, t_j)t_j$. To indicate the results clearly, we define $u = \frac{pS_{f2} - S_f^2 + S_t^2 - pS_{t2}}{S_f S_t - pS_{ft}}$.

By now, we have worked out the $grad_1$ value (i.e. $k$). According to above results, the equation of mean distance as follows:

$$md_1(B_i) = \frac{\sum_{j=1}^{p} |N_i(B_i, t_j) - b - kt_j|}{p\sqrt{1 + k^2}}. \tag{7}$$

### 4.3 Fluctuant Trend of Total Trustworthiness (FTT)

FTT is used to illustrate the fluctuant trend of total trustworthiness from $t_s$ to $t_e$. FTT is quite similar with FTBA, we also use a regression line to indicate the FTT, and the regression line is based on the least squares fit. The gradient and mean distance of regression line are denoted as $grad_2$ and $md_2$ respectively. Let $((t_s, TT(B_i, t_s)), (t_2, TT(B_i, t_2)), (t_3, TT(B_i, t_3)), ..., (t_e, TT(B_i, t_e)))$ denote the given data points of $B_i$'s total trust value. So, we can obtain following equations by above FTBA' methods:

$$k' = grad_2(B_i) = (-u' - \sqrt{u'^2 + 4})/2 \tag{8}$$

$$b' = \frac{S_{Tt} - k' S_t}{n}, \tag{9}$$

where $u' = \frac{pS_{Tt2} - S_{Tt}^2 + S_t^2 - pS_{t2}}{S_{Tt} S_t - pS_{Ttt}}$, $S_{Tt2} = \sum_{j=1}^{p} TT^2(B_i, t_j)$, $S_{Tt} = \sum_{j=1}^{p} TT(B_i, t_j)$, $S_t = \sum_{j=1}^{p} t_j$, $S_{t2} = \sum_{j=1}^{p} t_j^2$ and $S_{Ttt} = \sum_{j=1}^{p} TT(B_i, t_j) t_j$.

$$md_2(B_i) = \frac{\sum_{j=1}^{p} |TT(B_i, t_j) - b' - k' t_j|}{p\sqrt{1 + k'^2}}. \tag{10}$$

### 4.4 Trustworthiness Area (TA)

TA is a measurement method to calculate the quality of trustworthiness from $t_s$ to $t_e$. We use the trustworthiness areas to represent a buyer's trusted level. In this way, the quality of buyer's trustworthiness can be expressed visibly and measurably. The trustworthiness areas are categorized into the positive area and the negative area by a division line (trust value is 0.5), and we depict some sample areas in Fig.5, where X axis is the times of being advisor and the Y axis

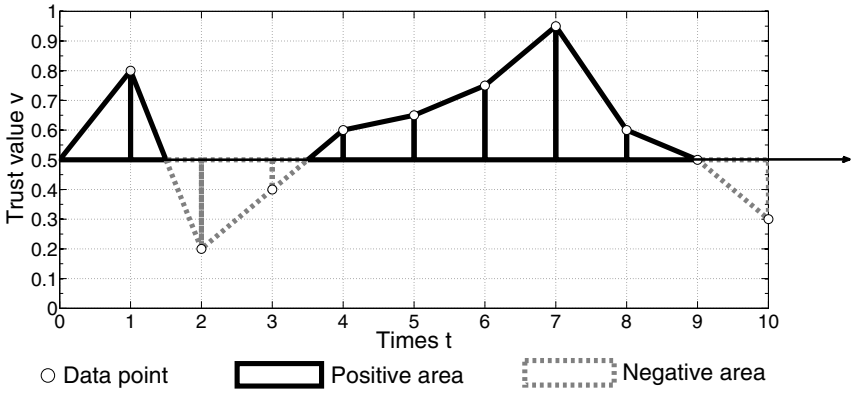



**Fig. 5.** Trustworthiness area in social network based e-commence platform

is the trust value given by other buyer based on a transaction. Finally, we define TA as:

$$TA(B_i, t_s, t_e) = PA(B_i, t_s, t_e) - NA(B_i, t_s, t_e) \tag{11}$$

where $PA(B_i, t_s, t_e)$ is positive area value and $NA(B_i, t_s, t_e)$ is negative area value from $t_s$ to $t_e$.

## 5 The Evalution-Based Robust Social Influence Method

In this section, we propose an Evalution-Based Robust Social Influence (EB-RSI) method in OSNs. The pseudo-code summaries the process of our EB-RSI method which is given in Algorithm 1. In our method, we adopt the above mentioned four influence impact factors that contain six parameters, they are TT, FTBA's gradient, FTBA's mean distance, FTT's gradient, FTT's mean distance and TA. By means of setting up different weights for these parameters, different inclined results will be generated. The weight set can be provided by individual buyers or domain experts. We normalized the six parameters to help prevent parameters with initially large ranges (e.g., TT and TA) from outweighing parameters with initially smaller ranges (e.g., FTBA' gradient and FTT' gradient). We use the popular normalization methods, Z-Score and Min-max [12]. The details of our method are indicated as follows:

Let $P_i = \{P(B_i, k)|k = 1, ..., 6\}$ be a $B_i$'s parameter set, it represents TT, FTBA's gradient, FTBA's mean distance, FTT's gradient, FTT's mean distance and TA respectively (Line 1 to 4 in Algorithm 1 is to calculate these six parameter values of each buyer). And we use $W = \{W(B_i, k)|k = 1, ..., 6\}$ to indicate corresponding weights of such parameters. Thus, the *Z-score* and *Min-max* methods are calculated as follows:

**Algorithm 1.** EB-RSI Algorithm

**Input:** Buyer set $B$, all trust relationships $T$, all transaction relationship $R$, the start time of transactions $t_s$ and the end time of transactions $t_e$;
**Output:** All buyers' influence value set $M$;
1: Calculate each buyer's TT value $P(B_i, 1)$ at time $t_e$ using Eq.1;
2: Calculate each FTBA's gradient value $P(B_i, 2)$ and FTBA's mean distance value $P(B_i, 3)$ from $t_s$ to $t_e$ using Eq.5 and Eq.7;
3: Calculate each FTT's gradient value $P(B_i, 4)$ and FTT's mean distance value $P(B_i, 5)$ from $t_s$ to $t_e$ using Eq.8 and Eq.10;
4: Calculate each buyer's TA value $P(B_i, 6)$ from $t_s$ to $t_e$ using Eq.11;
5: Integrate above six parameters into parameter set $P$;
6: **for** *each* $P_i$ *in* $P$ **do**
7: Normalize $P_i$ to $Z_i$ using *Z-score* method;
8: Calculate buyer $B_i$'s social influence value $V_i$ using Eq.15;
9: Normalize $V_i$ to $M_i$ using *Min-max* method;
10: **end for**
11: Return $M$;

*Z-score* method is used to normalize the six parameters (Line 7 in Algorithm 1), the range is $[-1.0, 1.0]$, it is calculated as:

$$Z(B_i, k) = \frac{P(B_i, k) - M_k}{S_k}, \tag{12}$$

where $M_k$ is the mean value of $P_k$ ($P_k = \{P(B_i, k)|i = 1, ..., n\}$, where $n$ is the number of buyers) and

$$S_k = \frac{\sum_{k=1}^{6} |P(B_i, k) - M_k|}{6}. \tag{13}$$

*Min-max* method is used to normalize the social influence value (Line 9 in Algorithm 1), the range is $[0.0, 1.0]$, it is calculated as:

$$M_i = \frac{V_i - V_{min}}{V_{max} - V_{min}}, \tag{14}$$

where $V_i$ is $B_i$' social influence value (Line 8 in Algorithm 1), and it is calculated as

$$V_i = \sum_{k=1}^{6} W(B_i, k) \cdot Z(B_i, k), \tag{15}$$

$V_{min}$ and $V_{max}$ are that minimum and maximum social influence values, respectively, for the given OSN.

## 6 Experiments

In this section, we compare our proposed EB-RSI method with SoCap method and conduct experiments based on the following two aspects: (1) In order to investigate the effectiveness of our EB-RSI method, we analyse the *Influence Ranking Trend* (*IRT*), that is the ranking of social influences. This trend can be used to illustrate the stability of influence evaluated by the two methods. (2) In order to investigate the robustness of our EB-RSI, we compare the performances of EB-RSI and SoCap in social influence evaluation when facing some typical attacks.

### 6.1 Experimental Setting

**Experimental Datasets:** In our experiments, we adopt a real social network dataset, Epinions (epinions.com), where each node represents a buyer or a seller, and each link corresponds to a trust relationship between a buyer and his/her advisor. Our work focus on studying the effectiveness and robustness of social influence evaluation methods. In order to clearly observe the computation process of participants' social influence, we extract a sub-network that has 362 nodes (200 buyers and 162 sellers) and 5453 links (5055 trust relationships and 398 transaction relationships). The extracting method selects 200 buyers and their corresponding sellers and relationships from original dataset randomly.

**Table 1.** Four weight sets

| Parameter Names | Weight-set-1 | Weight-set-2 | Weight-set-3 | Weight-set-4 |
|---|---|---|---|---|
| TT | 0.4 | 0.2 | 0.1 | 0.2 |
| FTBA's gradient | 0.1 | 0.15 | 0.2 | 0.1 |
| FTBA's average distance | 0.1 | 0.15 | 0.2 | 0.1 |
| FTT's gradient | 0.1 | 0.15 | 0.2 | 0.1 |
| FTT's average distance | 0.1 | 0.15 | 0.2 | 0.1 |
| TA | 0.2 | 0.2 | 0.1 | 0.4 |

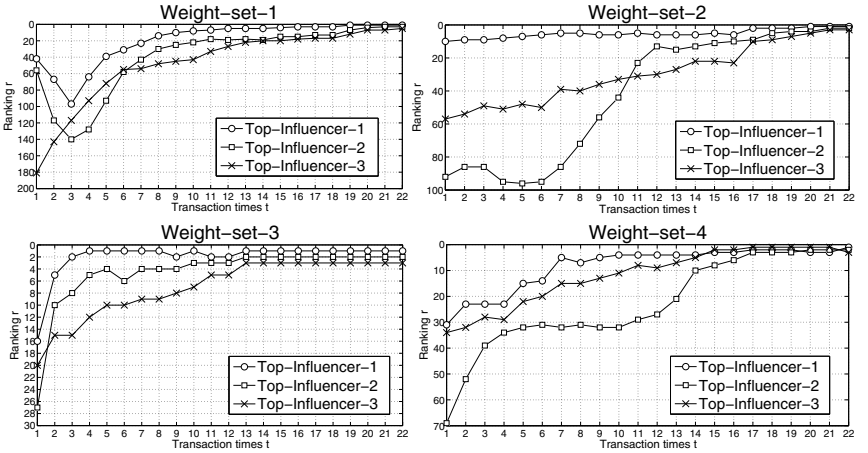



**Fig. 6.** The *Influence Ranking Trend* delivered by our EB-RSI

**Parameters Setting:** As introduced in *Section 5*, a buyer could specify different scales of weights for social influence evaluation. In our experiments, we set 4 groups of weights which have different combinations listed in Table 1. To investigate the effects of TT and TA to the influence evaluation, we set TT to 0.4 in Weight-set-1 and TA to 0.4 in Weight-set-4. All the impact factors in the cases of Weight-set-2 and Weight-set-3 have quite similar values. We investigate the *Top-3* influence values after a certain number of transactions with 4 groups of weights, and use "Top-Influencer-1", "Top-Influencer-2" and "Top-Influencer-3" to denote the *Top-3* influencers respectively.

**Experimental Environments:** All experiments were run on a machine powered by two Intel(R) Core(TM) i5-3470 CPU 3.20 GHz processors with 8GB of memory, using Windows 7. The code was implemented using Java 8 and the experimental data was managed by MySQL Server 5.6.

### 6.2 Experimental Results

**Exp-1: Effectiveness:** In order to investigate the effectiveness of our proposed EB-RSI method, we observe the *IRT*s of the *Top-3* influencers in a certain number of transactions. The experimental results are depicted in Fig.6 and Fig.7,

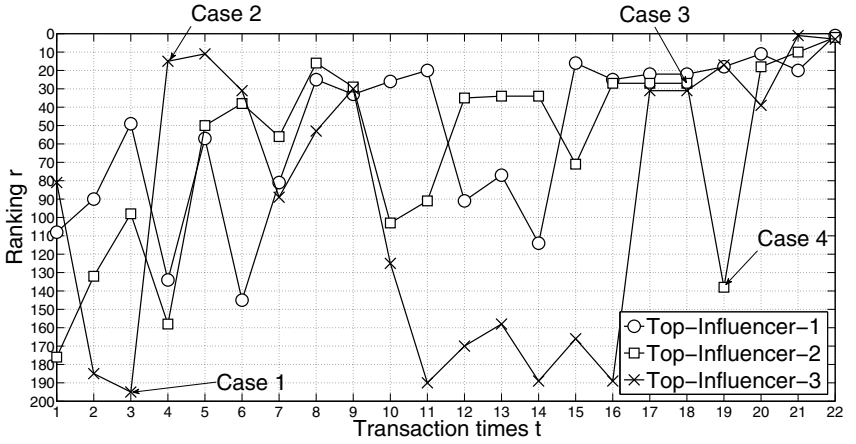



**Fig. 7.** The *Influence Ranking Trend* delivered by SoCap

where the X axis is transactions times and the Y axis is the influence ranking of buyers. The three curves in each graph represent the *IRT*s of the participants who have the *Top-3* influences after 22 transactions. From Fig.6, the *IRT* of each graph becomes stable after 22 transactions with different weights. From Fig.7, we can find that the *IRT*s of these influencers are fluctuant and unstable. For example, in *Case 1* and *Case 2*, although *Top-Influencer-3* has only one transaction, the influence ranking of *Top-Influencer-3* increases from 195 to 15 (see Table 2). Moreover, in *Case 3* and *Case 4*, *Top-Influencer-2* also has only one transaction, but the influence ranking of *Top-Influencer-2* decreases from 27 to 138 (see Table 2).

**Table 2.** The data of the cases in Fig.7

| ID | Transaction Times | Buyer ID | EB-RSI Value | SoCap Value | EB-RSI Ranking | SoCap Ranking |
|---|---|---|---|---|---|---|
| Case 1 | 3 | Top-Influencer-3 | 0.548 | 1.515 | 58 | **195** |
| Case 2 | 4 | Top-Influencer-3 | 0.567 | 1549.334 | 85 | **15** |
| Case 3 | 18 | Top-Influencer-2 | 0.629 | 51.991 | 50 | **27** |
| Case 4 | 19 | Top-Influencer-2 | 0.624 | 1.744 | 62 | **138** |

**Analysis:** This experimental result illustrates that the influencers identified by SoCap method are unstable with the increase of transactions. Because the SoCap method is complete based on current static OSN and ignore the influence trends. As a result, it cannot take the results of prediction into the social influence evaluation. Thus, SoCap may recommend a low quality influencer with weak influence to other buyers in the near future, e.g., the above mentioned *Top-Influencer-2* in *Case 3* and *Case 4* in Fig. 7. By contrast, our method can recommend the influencers who have stable influence and make other buyers obtain reliable product reviews. Therefore, our EB-RSI outperforms SoCap in the effectiveness.

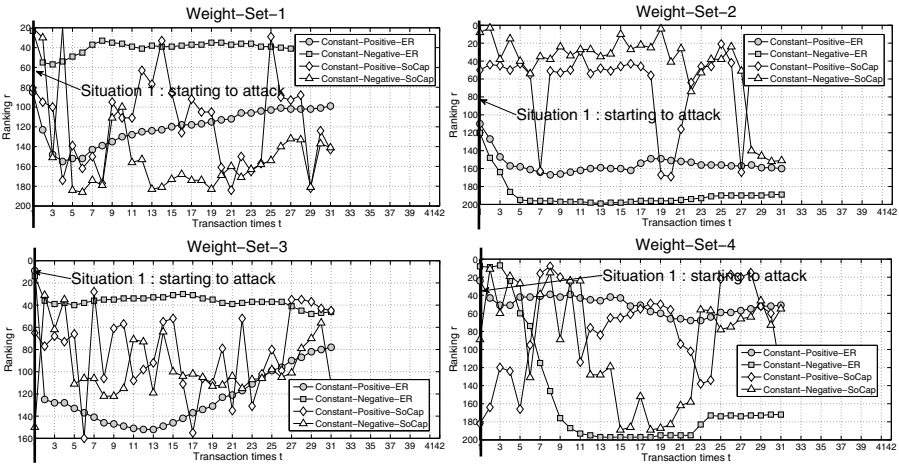



**Fig. 8.** The *IRT*s of buyers (Constant Attack)

**Exp-2: Robustness:** This experiment is to investigate if our method can defend against the typical attacks and recommend the honest *Top-K* influencers. As introduced in *Section 1*, it is necessary and significant to maintain the robustness of the social influence evaluation methods. We have introduced four typical attacks in *Section 1*, e.g., Constant[1], Camouflage[2], Whitewashing[3] and Sybil[4]. Since a new participant has only a few trust relationships and transitional relationships, and this new participant has very weak influence computed by our EB-RSI method, we skip the tests of Whitewashing attack and Sybil attack (these two attacks are all based on the new participant). The performances of the two methods when facing Constant attack and Camouflage attack are described as follows:

- *Constant*: We first select two buyers randomly and reset the rating values of all sellers given by the two buyers as 0 and 1 respectively, and give them 2 tags "Constant-Negative" and "Constant-Positive". After 31 transactions, the experimental results of *IRT*s are depicted in Fig.8, where we can see that the two *IRT*s by EB-RSI method (i.e., "Constant-Positive-ER" and "Constant-Negative-ER") decreases dramatically after starting transactions. This kind of downtrend contrast with the *IRT*s by SoCap method (i.e., "Constant-Positive-SoCap" and "Constant-Negative-SoCap"). 0 and 1 are two extreme values in the range of rating, which always reflects non-objective or dishonest opinions. Therefore, many buyers who have built trust relationships with the two buyers (i.e., "Constant-Positive-ER" and "Constant-Negative-ER") either removed the relationships or reduced the trust values. Then a robust social influence evaluation method should reduce the influences of such two types of buyers. Thus, our proposed EB-RSI method is more robust than SoCap method under the Constant attack.
- *Camouflage*: After 23 transactions, we reset the rating values of all sellers given by the *Top-2* influencers as 0 and 1 respectively, and give them 2

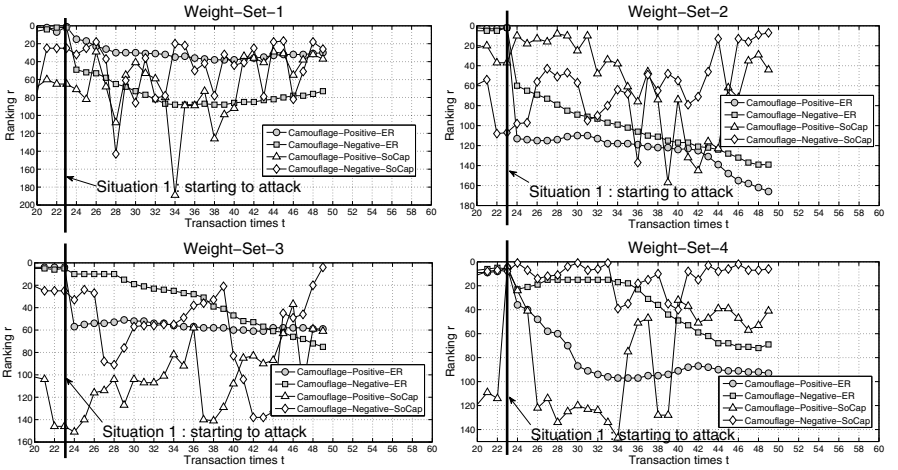



**Fig. 9.** The *IRT*s of buyers (Camouflage Attack)

tags, i.e., "Camouflage-Negative" and "Camouflage-Positive". The next 26 transactions (from 23 to 49) , The *IRT*s of *Top-2* influencers are depicted in Fig.9. From the figure, we can see that the two buyers' *Influence Ranking* (i.e., "Camouflage-Positive-ER" and "Camouflage-Negative-ER") by our EB-RSI method decreases dramatically after starting the attack (see Situation 1 in Fig.9). At the same time, the *Influence Ranking* of the two buyers (i.e., "Camouflage-Positive-SoCap" and "Camouflage-Negative-SoCap") by SoCap method are still unstable. As described in above *Constant* attack, the buyers who provide non-objective or dishonest reviews should be reprimanded (i.e., reduce the attackers' social influence). Therefore, whether reduce the influences of dishonest attackers becomes a key indicator of the robustness of influence evaluation method. Therefore, our EB-RSI method outperforms SoCap method in robustness.

## 7 Conclusion and Future Work

In this paper, we have proposed a novel EB-RSI method to accurately evaluate the social influence of participants in OSNs. In our model, we have taken four influence impact factors into consideration. In addition, to defend against the typical attacks from dishonest participants, we have adopted an evolutionary model to evolve trust relationships and transaction relationships. Furthermore, the effectiveness and robustness of our EB-RSI method have been validated by comparing with state-of-the-art method SoCap in a real OSN dataset.

In our future work, we plan to improve the efficiency of EB-RSI and apply it into a real social network based e-commerce platform to accurately evaluate the social influence of participants and then recommend the buyers with strong social influence to other buyers or retailers.

**Acknowledgements.** This work was supported by NSFC grant 61303019, 61073061, 61003044 and 61232006, and Doctoral Fund of Ministry of Education of China 20133201120012.

## References

1. Akoglu, L., Chandy, R., Faloutsos, C.: Opinion fraud detection in online reviews by network effects. In: ICWSM (2013)
2. Au Yeung, C.M., Iwata, T.: Strength of social influence in trust networks in product review sites. In: WSDM, pp. 495–504 (2011)
3. Bedi, P., Kaur, H., Marwaha, S.: Trust based recommender system for semantic web. In: IJCAI, pp. 2677–2682 (2007)
4. Berscheid, E., Reis, H.T., et al.: Attraction and close relationships. The Handbook of Social Psychology 2, 193–281 (1998)
5. Chen, W., Lu, W., Zhang, N.: Time-critical influence maximization in social networks with time-delayed diffusion process. In: AAAI, pp. 592–598 (2012)
6. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: KDD, pp. 1029–1038 (2010)
7. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: KDD, pp. 199–208 (2009)
8. Chen, W., Paik, I., Wang, J., Kumara, B.T., Tanaka, T.: Awareness of social influence on linked social service. In: 2013 IEEE International Conference on Cybernetics (CYBCONF), pp. 32–39 (2013)
9. Cho, Y.S., Ver Steeg, G., Galstyan, A.: Co-evolution of selection and influence in social networks. In: AAAI (2011)
10. Fiske, S.T.: Social beings: Core motives in social psychology. John Wiley & Sons (2009)
11. Franks, H., Griffiths, N., Anand, S.S.: Learning influence in complex social networks. In: AAMAS, pp. 447–454 (2013)
12. Han, J., Kamber, M.: Data Mining Concepts and Techniques: Data Preprocessing. Diane Cerra (2006)
13. Jiang, S., Zhang, J., Ong, Y.S.: An evolutionary model for constructing robust trust networks. In: AAMAS, pp. 813–820 (2013)
14. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2), 618–644 (2007)
15. Jung, K., Heo, W., Chen, W.: Irie: Scalable and robust influence maximization in social networks. In: ICDM, pp. 918–923 (2012)
16. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: KDD, pp. 137–146 (2003)
17. Kim, J., Kim, S.K., Yu, H.: Scalable and parallelizable processing of influence maximization for large-scale social networks? In: ICDE, pp. 266–277 (2013)
18. Kimura, M., Saito, K.: Tractable models for information diffusion in social networks. In: PKDD, pp. 259–271 (2006)
19. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: KDD, pp. 420–429 (2007)
20. Li, L., Wang, Y.: A trust vector approach to service-oriented applications. In: ICWS, pp. 270–277 (2008)
21. Okelo, B., Boston, S., Minchev, D.: Advanced Mathematics: The Differential Calculus for Multi-variable Functions. LAP Lambert Academic (2012)

22. Pandit, S., Chau, D.H., Wang, S., Faloutsos, C.: Netprobe: a fast and scalable system for fraud detection in online auction networks. In: Proceedings of the 16th International Conference on World Wide Web, pp. 201–210 (2007)
23. Subbian, K., Sharma, D., Wen, Z., Srivastava, J.: Finding influencers in networks using social capital. In: ASONAM, pp. 592–599 (2013)
24. Yaniv, I.: Receiving other people' s advice: Influence and benefit. Organizational Behavior and Human Decision Processes 93(1), 1–13 (2004)
25. Zhang, J., Liu, B., Tang, J., Chen, T., Li, J.: Social influence locality for modeling retweeting behaviors. In: AAAI, pp. 2761–2767 (2013)

# Study of Active Learning-based Trademark Number Recognition Method

**Yujie Shi[1], Jian Wu[1,*], Victor S. Sheng[2], Zhiming Cui[1] and Pengpeng Zhao[1]**

[1]The Institute of Intelligent Information Processing and Application, Soochow University, Suzhou 215006, China

[2]Department of Computer Science, University of Central Arkansas, Conway 72035, USA

Received: 13/01/2013; Accepted: 19/06/2013

**ABSTRACT**

In most image classification algorithms, the classifier model needs to train a large number of training samples. In practical application, labeling numbers of samples is a tedious and time-consuming task. So, how to select fewer suitable training samples from the numbers of unlabeled samples is a difficulty in the image classification algorithm. This paper proposes a trademark number recognition technique based on active learning algorithm. The method uses the human interaction to get trademark number area, and then uses the projection method to extract character characteristic which using the characteristic to split characters. Finally, use BvSB active learning algorithm to select high information samples which was used to train support vector machine classifier, and use the trained classifier to recognize trademark number. The experimental result shows that the classifier trained by the method has higher classification accuracy in the case of labeled fewer samples.

*Keywords:* Active learning, Trademark number recognition, BvSB, Support Vector Machine

## 1. INTRODUCTION

With the development of economy and technology, e-commerce technology is mature; and people gradually tend to the new shopping way of online shopping. But the types and styles of goods are various, and a commodity may have thousands of styles. So, how quickly and efficiently to find the target

Corresponding author. jianwu@suda.edu.cn

commodities in multitudinous goods has become the obstacles of the user shopping. As we all know, any piece of merchandise has a number, and the same style goods have one number, the different goods have diverse number. So, using trademark number to retrieve merchandise can greatly shorten the time of the user searching goods, and allows users to quickly and efficiently search the target commodity. Therefore, trademark number recognition technology has broad application prospects.

Trademark number recognition technology is that the system automatically identifies trademark number character from the image, and the identified trademark number can conveniently retrieve goods for user. This paper put forward a trademark number recognition technique based on active learning algorithm. Active learning algorithm mainly includes two parts of learning engine and sampling engine [1, 2]. Learning engine is the training process of classifier. The purpose of sampling engine is that under the least labeled expense, getting the labeled sample set which can extremely improve the generalization performance of the classifier [3, 4]. In this paper, using an active learning method based on the optimal label and suboptimal label (Best vs second Best, BvSB) proposed by Joshi *et. al* to chose the high information sample [5], and using the selected samples to train the SVM classifier. Support Vector Machine (SVM) is a new machine learning method which based on statistical learning theory and was proposed by vapnik *et. al* in 1995 [6, 7]. SVM can effectively solve the problem of small sample learning, nonlinear and high dimensional pattern recognition [8].

Trademark number recognition technology mainly contains three main steps, the first is trademark number area achieving, second is character segmentation, the third is character recognition. Getting the trademark number area is the key first step of trademark number recognition. Because of the influence of light and image noise, there are large errors using the traditional method to get the trademark number area for low-quality images. This paper put forward the way of human interaction to get the trademark number area. Finally, we use BvSB active learning algorithm to select the high information samples, and use the selected samples to train classifier which was used in trademark number recognition. Due to trademark number composed by uppercase letters and numbers, and the class number of sample is not large, so we use the classification algorithm of “one to many” for Support Vector Machine to identify trademark number. The experimental result shows that the classifier trained by the method has higher classification accuracy in the case of labeled fewer samples.

## 2. BvSB ALGORITHM

BvSB algorithm is the improvement for the active learning algorithm based on entropy. So, we first brief the active learning algorithm based on entropy. Sample set includes unlabeled sample set and labeled sample set. The paper set labeled sample set is $L = \{(x_1, y_1), (x_2, y_2), \mathrm{L}, (x_m, y_m)\}$ and $Y = \{1, 2, L\}$ is the possible labels for all categories; $y_i$ $y$ is the label of sample $x_i$; unlabeled sample set is $U = \{x_{m+1}, x_{m+2}, L, x_n\}$. The probability of sample $x_i$ in the unlabeled sample set belongs to every category is $\mathrm{P}(y_i | x_i)$. So, the standard of active learning algorithm based on entropy is as follows:

$$ENT^* = \arg\min_{x_i \in U} - \sum_{y_i \in Y} P(y_j | x_i) \log P(y_j | x_i) \tag{1}$$

The active learning algorithm based on entropy stipulate that the sample which has bigger entropy is harder to decide the category for the current classifier, so the information of the sample is higher. Then, the algorithm chose the maximum entropy sample to manual label, and then adds it to the training sample set which was used to update classifier. But, in the multi-class classification problem, entropy cannot always represent the uncertainty of sample. The classification uncertainties of some samples with smaller entropy may be higher than some other samples with slightly larger entropy [5]. According to this problem, Joshi *et. al* proposed BvSB active learning algorithm.

BvSB (Best-versus-Second-Best) is a direct active learning sample selection standard. In the standard, only consider the two sample categories which largest impact the classified results of sample, and ignore the less affected categories. So, this algorithm is simple to realize.

If a sample is close to the classified surface, it has large information. At the same time, the category of this sample was difficult to determine. So, if a sample has the similar probability between the best category and the second best category, the sample has higher information. The sample selected by BvSB standard has high representation, and BvSB standard has an effective measure. The standard of BvSB algorithm is as follows:

$$\begin{aligned} BvSB^* &= \arg\min_{x_i \in U}( \min_{y \in Y, y \neq y_{best}} (p(y_{best} | x) - p(y | x))) \\ &= \arg\min_{x_i \in U}(p(y_{best} | x) - p(y_{\mathrm{sec}\,ond-best} | x)) \end{aligned} \tag{2}$$

In the formula, the best label and second label of sample $x_i$ respectively are $y_{best}$ and $y_{second-best}$, thus the corresponding probability are $p(y_{best}|x_i)$ and $p(y_{second-best}|x_i)$.

The active learning algorithm of BvSB can make full use of the characteristic of the learning algorithm, and it has the features of condensed and efficient.

## 3. TRADEMARK NUMBER RECOGNITION

Figure 1 is the frame diagram of trademark number recognition method which is proposed in this paper and based on Support Vector Machine. There are three main steps in the entire process of trademark number recognition, and they respectively are trademark number area achievement, character segmentation and character recognition. First, get the trademark number area from the collected images. This is the first step and also a crucial step in the entire of trademark number recognition process. This paper realized by the way of human interaction. Then do character segmentation from the gotten trademark number area, and the segmentation result is obtained the single independent trademark number character, which is the most important step in the process of trademark number recognition in this paper. This paper uses the projection to get character feature, and then split character. Finally, using Support Vector

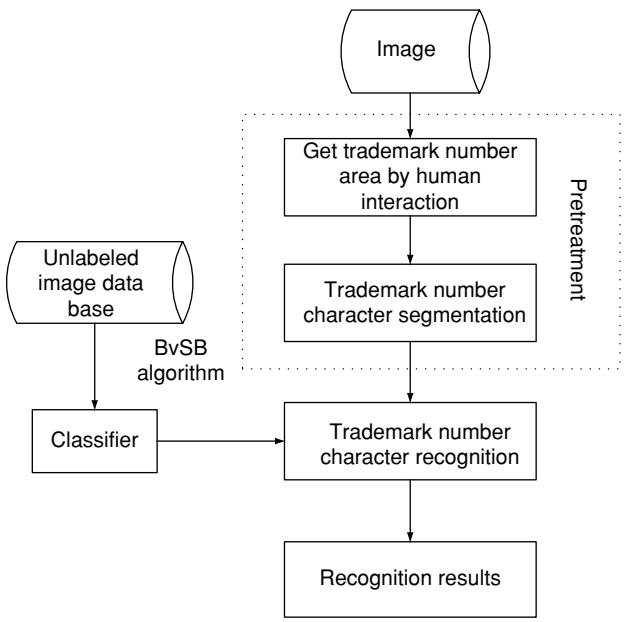



Figure 1. The frame diagram of trademark number recognition.

Machine to build a classifier which is used to identify trademark number, and then getting the recognition result by together the recognized single character. The next few sections will describe each step implementation process.

### 3.1. Obtain the Trademark Number Area

Trademark number recognition is also known as trademark number character position, which is to identify the required trademark number area from the whole image. Similarly to the license plate location method of license plate character recognition process, there are two common methods of trademark character position. One is the character positioning method based on the black-and-white image, while another one is based on the color image [9]. Although the above two methods have been put into practice, both of them have certain requirements on the image quality. Once the image has been polluted by noise, its image quality will be low ,which will be lead to a large deviation using the two methods to obtain trademark number regions, even worse is that it can't find trademark number character region.

Human interaction method is that user selects the trademark number region from the acquired image firstly, and then we shear the image around the trademark number area according to the line drawn by the user. Using the horizontal projection method, we can strike a one-dimensional histogram of image features which will be performed by filter processing afterwards. Analysis the histogram and find out the maximum value of the two sides that perpendicular to the area which has been selected by user. Finally obtain the final trademark number area after.

As shown in figure 2, the figure (a) is the acquired original image, figure (b) is the horizontal projection histogram of the sheared image after filtering and figure (c) is the result image of acquired trademark number area.

### 3.2. Character Segmentation

In essence, image segmentation is the classification process which according the characteristics different between the pixels in the image, that is the image was divided into a pixel region which have the consistent characteristics [10]. Trademark number character segmentation is to segment the characters in the trademark number area into single characters. But the image quality has some certain impacts for image segmentation results, and the common error character segmentation has the following two kinds.

In the portion of the trademark number image, since the light intensity is not enough, an independent character in binarization image is not continuous, that

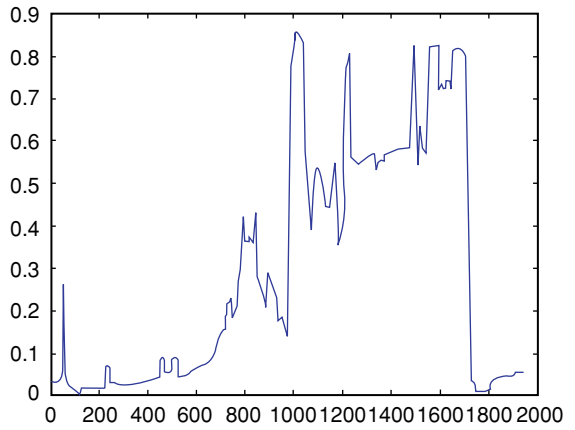





Figure 2. (a)The collected image; (b) the horizontal; (c) The result image of trademark number area.

is a character appears breakage. In this case, it is easy to segment a character into two images. Another case is that affected by the image noise, image character is defiled, and the result is that two or more characters are get together. At this point, we cannot separate the deferent characters in the character segmentation.

In order to preferably segment the trademark number character and try to avoid above problems, this paper using vertical projection method to extract character features. The specific method is as follows, vertical-projection

sheared image, remove the noise with the median filtering, calculate the histogram maximum, and obtain the initial segmentation lines. From the figure 3 (a), we can see that some extreme point segmentation lines do not meet the conditions. We can judge the segmentation lines whether is effective through the analysis of the split line adjacent to the two line of the image the morphology of connected area, and then filter out false segmentation line.

As shown in Figure 3, (a) is the feature histogram which was obtained from the vertical projection the character areas gotten from the above section. Because of the trademark number is black and the black character pixel value is low, so the sum of the area pixel value between two characters is high. Reflected in the histogram is that the maximum value is emerged. According to the maximum value position determine dividing line. (b) is the ultimately determined results graph of partition line. From the experimental results, it can be seen that this method can achieve good segmentation of characters.

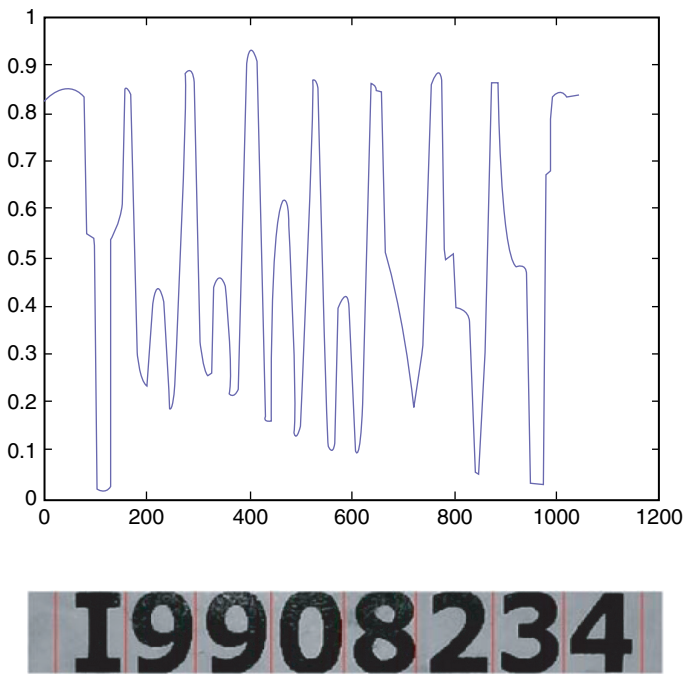



Figure 3. (a) Trademark number regional vertical projection feature histogram; (b) the finnal result figure of character segmentation.

### 3.3. Character Recognition

The process of character recognition in the trademark number recognition method mainly includes three problems. Firstly, how to select fewer suitable samples to manual label from the numbers of unlabeled samples. Second is the classifier model which was used in the trademark number recognition. Third is the image feature used to distinguish image. This paper use BvSB active learning algorithm to select the high information unlabeled samples to label, and BvSB algorithm has been introduced.

Trademark numbers are constituted by some characters, such as the capital letters and numbers. The classifier model contains 36 categories for number of capital characters being 26 and that of digitals being 10. According to the characteristic, this paper use the classification algorithm of "one to many" for Support Vector Machine to train classifier, and then finish the character recognition step with it. In the amount of image features, the paper extracts coarse mesh characteristics according to the feature of trademark number image. Then we will detail the main content of support vector machine and coarse mesh characteristics.

#### 3.3.1. Classifier model

For features of its powerful classification, generalization ability and flexible classification method, Support vector machine (SVM) has been widely applied in pattern recognition field. Support vector machine is first used to solve two kinds of classification problem. How to extend two kinds of classification method to multiple category classification is one of the important contents of study in SVM.

Basic thought of SVM is that dividing inhomogeneous point by straight line; when sample point cannot be divided by line, through the nonlinear transformation transforms input space into a high dimensional space. Commonly used kernel functions are as follows:

1) Linear Kernel function: $K(x, y) = xgy$
2) Polynomial kernel function: $K(x, y) = (< x, y > +1)^d$
3) Radiate basis function (RBF) kernel function: $K(x,y) = e^{\frac{-|x-y|^2}{\sigma^2}}$
4) Layer neural network (Sigmold) kernel function: $K(x, y) = \tanh(s < x, y > + c)$

Usually, kernel function is through a lot of experiments to obtain optimal parameters. Due to radial basis kernel function's corresponding feature space is infinite dimensional, limited sample in the feature space is sure linear separable, so

the radial basis kernel function is the most commonly used kernel function. Therefore, this paper uses radial basis kernel function to extraction SVM classifier, and the parameter is mapping the sample data to the appropriate feature space.

SVM classifier mainly has four classes which are respectively one-on-one, a pair of many, SVM decision tree method and multiclass SVM. Although "one to many" classification algorithm consider all the sample every classification, which eventually led to the slow training speed. But "one to many" classification algorithm is simple and easy to implement, and in article number recognition need only a total of 36 classifier; the scale is small, so this paper adopt the "one to many" classification algorithm to structure classifier.

The basic idea a pair of many is: when the training in a category of sample return for a class, the other to the rest of the sample for another kind, such $k$ categories of sample will structure $k$ SVM. We classify unknown sample for a class with maximum classification function value of that class.

#### 3.3.2. The feature extraction of trademark number image

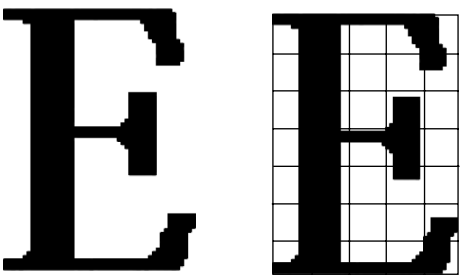
After two steps' processing, we get character a single article. Before article number identification, we need to extract feature article number. Common character statistical characteristics are mainly coarse mesh characteristics, per-pixel characteristics and 13 point features, etc. So this paper uses coarse mesh characteristics to realize article number recognition.

Coarse mesh characteristics divide the character into $N \times M$ grid and statistic the pixel number belongs to the character in each grid, each grid reflects a certain characteristics of character. In the recognition stage, the grid is together as the statistical features of the characters. Coarse mesh feature extraction method need to turn the character size and position normalization, in this paper, the unity of the character size is $70 \times 50$, into $7 \times 5$ grid. Each grid feature extraction of $10 \times 10 = 100$ pixels, each grid compute character pixels percentage.

With the letter "E" as an example, extract coarse mesh characteristics. Figure 4 shows $70 \times 50$ size for characters, which is divided into a $7 \times 5$ grid. Figure (c) shows the 35 grid is the percentage of the letter "E". In the grid of line 2 and column 3, because there is no pixel belongs to the letter "E", so percentage is 0.

#### 3.3.3. The process of character recognition

Figure 5 shows the process of character recognition, and we can evidently see that character recognition is an iterative process. Use BvSB active



| | | | | |
|---|---|---|---|---|
| 0.53 | 0.86 | 0.3 | 0.52 | 0.42 |
| 0.3 | 0.8 | 0 | 0.05 | 0.29 |
| 0.3 | 0.8 | 0 | 0.57 | 0 |
| 0.3 | 0.86 | 0.31 | 0.83 | 0 |
| 0.3 | 0.8 | 0 | 0.28 | 0 |
| 0.3 | 0.8 | 0 | 0 | 0.42 |
| 0.51 | 0.78 | 0.3 | 0.36 | 0.59 |

Figure 4. (a) Character diagram to identify; (b) 7 × 5 grid feature schematic diagram; (c) Coarse mesh characteristics letters percentage diagram of character "E".

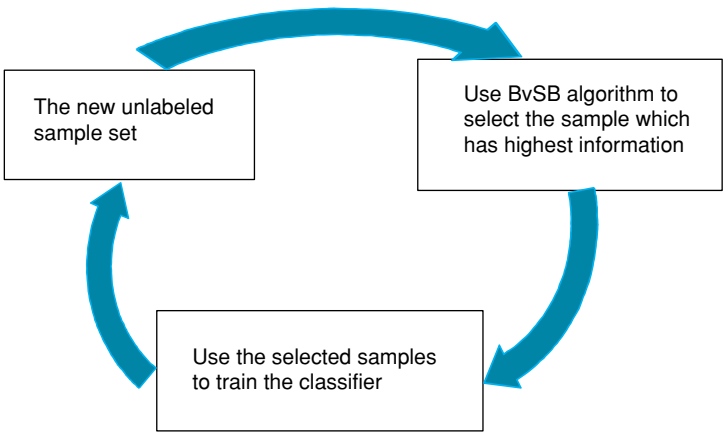



Figure 5. the iterative process of character recognition.

learning algorithm to select the high information samples from unlabeled sample set, at the same time, remove the selected samples from the unlabeled sample set. The training samples that were used to retrain the classifier have the maximum information which was calculated by the newest classifier. The end conditions of iteration are the classification accuracy of the classifier reaches the threshold or the number of training samples reaches the threshold.

Use coarse mesh characteristics to extract the character image feature, and use BvSB algorithm to select training samples. With this, it can improve the

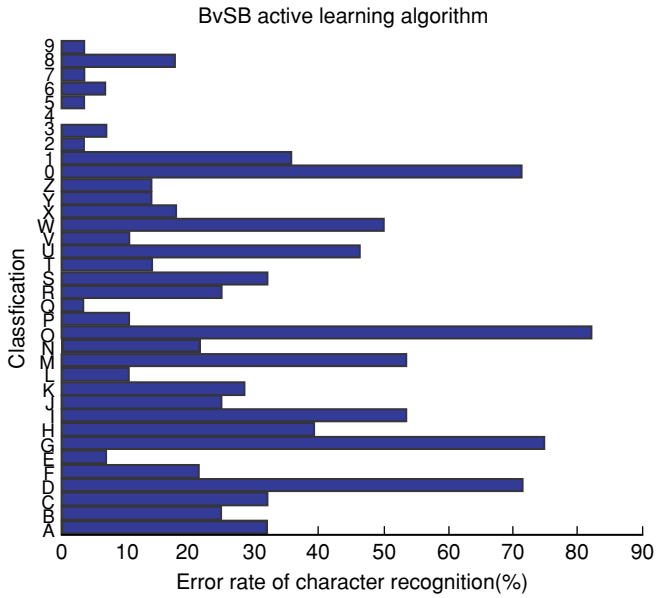



Figure 6. Error rate of character recognition.

classification accuracy of the classifier. In order to validate the effectiveness of the classifier, artificial character data set experiment is adopted in this paper. The data sets, there are 26 kinds of capital English letters and 10 kinds of digital, a total of 36 classes. The size of unlabeled sample set is 1260, and the size of test sample set is 1008. Initial classifier was trained by 36 samples that there is 1 sample in every class. Character recognition error rate result diagram is shown in figure 6 and transverse said error rate while longitudinal axis is on behalf of each character. As can be seen from figure 6, due to the number 0 and the letter o, G, D are quite similar, so the classification error rate of these characters is slightly high. The classification error rate of other characters is relatively low. This also shows that the classifier has higher correct classification rate.

## 4. EXPERIMENTAL ANALYSIS

In order to verify the effectiveness of the trademark number recognition methods, this paper use 103 pieces of images which random collected to do experiment. In order to verify the universality and persuasion of the algorithm, the experimental image data of this paper was obtained by different methods and different shooting equipment. Experimental procedure was realized by

Matlab 2010a on Dell PC machine. Machine configuration is as follows: Pentium(R) Dual-Core CPU E5300@2.60 GHz, 4G memory, Windows 7 operating system.

Because of the classifier in this experiment is obtained by training which use the artificial set with the size of 70 × 50. So after the two steps of trademark number access and character segmentation, the resulting single character need normalization processing. The size of character normalized processing is same to the size of characters in the training set.

As shown in Figure 7, there is the trademark number recognition result picture of the two sample images in the experimental dataset. The diagram (a) is one of the correct result figures of trademark number recognition, and (b) is one of the error result figures. From these two figures, we can clearly see that the pretreatment result of figure (a) is correct, and then with our classifier can get the correct trademark number result. The letter "L" in figure has segmentation error, and it was segmented to the character "1" and character "-". That is, figure (b) has the error result in the pretreatment process of trademark number recognition, and the final trademark number recognition results was error, too.

The main factors that affect pretreatment results in the trademark number recognition process have tree:Schematic diagram of segmentation error; Schematic diagram of image noise influence; Schematic diagram of illuminate impact. The three factors affect the pretreatment results. If the


Figure 7. (a) the correct recognition result ; (b) the error recognition result.

Table 1. The accuracy result of trademark number recognition.

| Image number | Recognition accuracy |
|---|---|
| 103 | 94.5% |

pretreatment cannot get the correct result, the final recognition will be also error. But, exclude the influence of the pretreatment, the trademark number recognition accuracy only decided by the accuracy of classifier. Using BvSB active learning algorithm to select parts of samples and train the classifier, and then use the classifier to do the camera trademark number recognition. Table 1 is the accuracy result of trademark number recognition:

## 5. CONCLUSIONS

The paper proposed a trademark number recognition based active learning algorithm. Use BvSB active learning algorithm to select samples, and then use the selected samples to train classifier. The method can chose the fewer samples, but the classifier has the higher accuracy. Character recognition is the last and the key step of the trademark number recognition technique, and it belongs to the image classification field. According the problem that training classifier need to artificial label numbers of samples in the traditional image classification, we use BvSB active learning algorithm to choose unlabeled samples. With this method, the trained classifier has higher classification accuracy with fewer samples. But the result of pretreatment influence the final result of the trademark number recognition. So, the next study focus is improving the accuracy of the pretreatment result.

## ACKNOWLEDGMENTS

This research was partially supported by the Natural Science Foundation of China under grant No. 61003054, 61170020, and 61170124, the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under grant No. 10KJB520018, the Key Technology Research and Development Program of Jiangsu Province under grant No. BE2012075, the Key Technology Research and Development Program of Jiangsu Province under grant No. SG201257, and the National Science Foundation (IIS-1115417).

## REFERENCES

[1] Wu Weining, Liu Yang, Guo Maozu, Liu Xiaoyan. Advances in active learning algorithms based on sampling strategy [J]. Journal of Computer Research and Development, 2012, 19(6): 1162–1173.

[2] Chen Rong, Cao Yongfeng, Sun Hong. Multi-class image classification with active learning and semi-supervised learning [J]. Active automatica sinica, 2011, 37(8): 954–962.

[3] D. Tuia, E. Pasolli, W.J. Emery. Using active learning to adapt remote sensing image classifiers [J]. IEEE Remote Sensing of Environment, 2011, 9(115), 2232–2242.

[4] Hanneke S. Rates of convergence in active learning [J]. The Annals of Statistics, 2011, 39(1): 333–361.

[5] Joshi A J, Porikli F, Papanikolopoulos N. Multi-class active learning for image classification [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2009: 2372–2379.

[6] Dai Shijie, Gao Zhangying, Chang Shuying. A recognition system for license plate based on SVM [J]. Microcomputer Information, 2009,25(9):26–28.

[7] Xiong Chunrong, Huang Wenming, Li Meijin. Research on license plate character recognition based on character characterisitic and Support Vector Machine [J]. Automation Application Technology,2010,29(1):64–66.

[8] BURGES C J C. A Tutorial on Support Vector Machines for Pattern Recognition [J]. Data Mining and Knowledge Discovery, 1998,2(2): 121–167.

[9] Zhan Tianxu. An investigation on the stability of object extraction [J]. IEEE Transactions on AES, 1997, 33(3): 1051–1060.

[10] Haralick R M, Shapiro L G. Image segmentation techniques [J]. Computer Vision and Graphics & Image Process, 1985, 30(2): 100–132.

Quan et al. BMC Bioinformatics 2014, 15(Suppl 12):S5
http://www.biomedcentral.com/1471-2105/15/S12/S5

BMC Bioinformatics

RESEARCH Open Access

# Improved packing of protein side chains with parallel ant colonies

Lijun Quan[1†], Qiang Lü[1,2*†], Haiou Li[1], Xiaoyan Xia[1,2], Hongjie Wu[1,2,3]

*From* IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2013)
Shanghai, China. 18-21 December 2013

## Abstract

**Introduction:** The accurate packing of protein side chains is important for many computational biology problems, such as ab initio protein structure prediction, homology modelling, and protein design and ligand docking applications. Many of existing solutions are modelled as a computational optimisation problem. As well as the design of search algorithms, most solutions suffer from an inaccurate energy function for judging whether a prediction is good or bad. Even if the search has found the lowest energy, there is no certainty of obtaining the protein structures with correct side chains.

**Methods:** We present a side-chain modelling method, pacoPacker, which uses a parallel ant colony optimisation strategy based on sharing a single pheromone matrix. This parallel approach combines different sources of energy functions and generates protein side-chain conformations with the lowest energies jointly determined by the various energy functions. We further optimised the selected rotamers to construct subrotamer by rotamer minimisation, which reasonably improved the discreteness of the rotamer library.

**Results:** We focused on improving the accuracy of side-chain conformation prediction. For a testing set of 442 proteins, 87.19% of $\mathcal{X}_1$ and 77.11% of $\mathcal{X}_{12}$ angles were predicted correctly within 40° of the X-ray positions. We compared the accuracy of pacoPacker with state-of-the-art methods, such as CIS-RR and SCWRL4. We analysed the results from different perspectives, in terms of protein chain and individual residues. In this comprehensive benchmark testing, 51.5% of proteins within a length of 400 amino acids predicted by pacoPacker were superior to the results of CIS-RR and SCWRL4 simultaneously. Finally, we also showed the advantage of using the subrotamers strategy. All results confirmed that our parallel approach is competitive to state-of-the-art solutions for packing side chains.

**Conclusions:** This parallel approach combines various sources of searching intelligence and energy functions to pack protein side chains. It provides a frame-work for combining different inaccuracy/usefulness objective functions by designing parallel heuristic search algorithms.

## Introduction

The accurate packing of side chains plays a very important role in modelling protein structures. In ab initio structure prediction, the goal is to choose a rotamer for each position so that the molecule is close to the natural structure. In homology modelling, the goal is to predict the structure of a protein that is homologous to another of a known structure [1,2]. In protein design, the goal is to find an amino acids sequence that will fold into a particular backbone [3]. In flexible ligand docking, the goal is to display a structural change ranging from large movements of entire domains to small side-chain rearrangements in the binding site [4-6]. Based on Anfinsen's hypothesis [7], the problem of packing side chains is usually mapped into a combinatorial optimisation problem and can be solved in a number of ways. However, a fixed backbone, an energy function and a possible rotamer set are always foundations of this widely studied formulation. All the current existing

* Correspondence: qiang@suda.edu.cn
† Contributed equally
[1]School of Computer Science and Technology, Soochow University, Suzhou, 215006, China
Full list of author information is available at the end of the article

algorithms for the side-chain problem can be divided into two categories, heuristic and deterministic.

The side-chain problems have been proven as non-deterministic polynomial-time hard (NP-hard) [8-10]. Even when an approximate solution is sought within $O(cnR)$ from the optimum, where $c$ is a constant, $n$ is the number of residues and $R$ is the average number of rotamers per residue [11,12], the packing side chains cannot be solved successfully. Computational complexity analysis suggests that any global optimisation algorithms for this problem may, in the worst case, run in exponential time [11]. When they converge, dead-end elimination (DEE) algorithms [13,14] are designed to find the global minimum energy. Heuristics are not guaranteed to find a global minimum, but they almost always find a low-energy conformation in a reasonable time [15]. Therefore, heuristic algorithms become a natural choice for tackling the side-chain modelling problem. Traditionally, all heuristic approaches solve such side-chain problems as a single-objective optimisation Problem (SOP), using Monte Carlo (MC) [16], Ant Colony (AC) [17], and Simulated Annealing (SA) [18]. Some of the heuristic methods combine multiple strategies, such as a combination of DEE and the $A^*$ algorithm [19], and combination of SA and MC [20-22]. The common feature of these heuristic approaches is that they all use an optimisation based on a single objective function.

Another method for solving the side-chain problem was by using the theory of decomposing the underlining residue relationship. One such method is SCWRL [23-25,15], which is widely used because of its speed, accuracy and ease of use. SCWRL3 decomposes original residue graphs to connected subgraphs, which cannot be disconnected by the removal of a single vertex. They find the global minimal energy conformation for the residues in these subgraphs [25]. The authors who proposed the SCWRL methods also observed that residues with a single rotamer or a single neighbour can be eliminated from the residue graph. Then SCWRL4 [15] transfers the original residue graphs to a tree for speeding up the solver. However, in the tightly packed environments of protein interiors, these methods will inherently lead to atomic clashes and hinder the prediction accuracy. Therefore, a new method, CIS-RR, performs clash detection-guided iterative searches (CIS) of side-chain rotamers whilst continuously optimising side-chain conformations using a conjugate gradients method [26].

In general, methods for predicting side chains seem to be limited not by the quality of search algorithms, but also by the quality of the energy functions employed [23]. An energy function typically consists of a combination of weighted energy terms. It is not hard to find different approaches, which develope distinctive kinds of energy functions. For example, SCWRL3 use an energy function based on logarithmic probabilities of rotamers and a simple repulsive steric energy term [25]. However, SCWRL4 also uses a short-range, soft van der Waals interaction potential between atoms rather than the linear repulsive-only function used in SCWRL3, as well as an anisotropic hydrogen bond function similar to that used in Rosetta [15,27]. The energy function of CIS-RR is also a modified the energy function of SCWRL3. The first improvement is to add attractive energy and weights to the van der Waals potential. The second improvement is to penalise the drifting of side chain dihedral angles away from the nearest rotamer library values for the original rotamer term. The existence of different energy functions implies that all energy functions are inaccurate in a universal sense (inaccuracy), but each of them is very useful in some specific sense (usefulness). This hypothesis is referred to as the inaccuracy/usefulness property [28]. The approaches based on SOP all use a single inaccuracy energy function to model side chains, so the results are sometimes inaccurate in a quantitative sense for discriminating native or near-native conformations.

In this study, a novel approach is proposed to assemble the usefulness and decrease the inaccuracy of different energy functions. We believe that it is more reasonable to model packing side chains as a multi-objective optimisation problem (MOP). Different energy functions should be combined to the best possible extent. As this idea has been successfully applied to de novo prediction of protein backbone [28,29], we also used parallel ant colony optimisation based on SHOP (SHaring One Pheromone matrix) [30]. Our parallel strategy is not for speeding up the predictor, but can be used to hybridise the usefulness of different energy functions. All energy functions can be adopted by an individual colony. In this way, we can avoid the sensitivity of the optimised parameters of energy functions, so we expect to obtain better generality of our predictor. This parallel strategy has been validated experimentally.

## Methods

We propose a novel parallel ant colony optimisation (ACO) metaheuristic frame-work for packing protein side chains by single-heuristic multi-objective algorithms (SHMO) to reduce the inaccuracy of a single energy. We denote a heuristic algorithm by $h$ and different energy functions by $\varepsilon = \{E_1, \ldots, E_k\}$, where the number of threads amount to $k$. This type of algorithm is generally denoted by $\prod_h (E_i|\Theta)$ where $\Theta$ refers to the control parameters in terms of heuristic search algorithms and can usually be tuned empirically before starting, or adaptively during the algorithm [28]. In the pacoPacker algorithm, $h$ adopts ACO, and $\Theta$ contains two variables, private and public. To be more specific, all ant colonies share one common pheromone matrix $T$

as a public variable, and each ant colony has a private variable including heuristic matrix $H_i$ and two other parameters, $\alpha_i$ and $\beta_i$ . $A = \{\alpha_1, \ldots, \alpha_k\}$, determines the importance of the pheromone and $B = \{\beta_1, \ldots, \beta_k\}$, determines the importance of the heuristic matrix $H = \{H_1, \ldots, H_k\}$. This paper's method can be described as $\prod_{AC}(E_i|\alpha_i, \beta_i, H_i, T)$. The Rosetta3.4 platform [31] is quite mature and supports the object-oriented paradigm, therefore pacoPacker uses Rosetta3.4 for building rotamer libraries, constructing interaction graphs, and scoring structures. Using Rosetta3.4 and OpenMP [32], our scheme is easy to implement.

### Search space

For an amino-acid sequence $t$ with $n$ length of residues, its side chains are packed with the lowest free energy. Let the rotamer library for $t$ be $R = \{R_1, \ldots, R_n\}$, where the rotamer set is $R_i = \{r_1, ..., r_{m_i}\}$ for each residue $i$ in $t$, the number of rotamers belonging to $R_i$ amount to $m_i$, and different rotamer sets have a different quantity of rotamers. Rotamers were read from Dunbrack backbone dependent rotamer library (2010 version), such that frequencies and dihedral angles varied with the backbone dihedral angles $\Phi$ and $\psi$ [33].

### Energy function

We adopted the same energy functions used by Rosetta. These scores are combinations of different weights and energy items, such as residue-environment and residue-residue interactions, secondary structure packing, chain density and excluded volume [28]. It does not matter which function is more accurate as all the energy functions share the inaccuracy/usefulness property. The Rosetta energy functions are adopted here to illustrate the implementation of our parallel approach. We forked eight threads to run separately using different energy functions, which rule out any side-chain-independent energy terms. Different threads have different private variables, which are listed in Table 1. Table 1 shows the weight of each score term on different score functions. Each score term is represented by letter (A, B, etc.), which correspond to Table 2.

### Implementation of the algorithm

Eight parallel threads were created in our SHMO implementation. Figure 1 depicts the design of pacoPacker. Using a protein backbone as the input of pacoPacker, the rotamer library is generated based on the target sequence by using the Rosetta platform. The outputs are proteins with side chains predicted by ant colonies. From the information shown in Figure 1, eight different ant colonies share a single common pheromone matrix $T$ to exchange their search experience asynchronously. Each colony is directed by its own energy functions, which both co-evolve towards a better state.

Next, we will focus on a single ant colony to pack side chains. Construction by an ant colony is described as follows:

1. Conduct side chains based on the selection equation for each ant.
2. Perform the local search on each odd-numbered iteration ant.
3. Update global best ant $s_{gb}$with iteration best ant $s_{ib}$ if $E(s_{ib})$ is lower.
4. Update the pheromone matrix $T$ based on $s_{gb}$.
5. If the termination criterion is met, let's return to $s_{gb}$, or repeat steps 1 to 5.

In this workflow, each colony terminates when one of the following criteria is met: the colony runs for a specified number of iterations; and there is no energy improvement during the last several iterations. Two important equations, the selection equation and the update pheromone matrix equation are explained below.

Each ant conducts the conformation by assembling rotamers from $R$. The ant picks up a rotamer $r_j$ from the rotamer set $R_i \in R$ for residue $i$. For $g^{th}$ thread, the

**Table 1 Score function and ACO parameters**

| Thread ID | Score function | Score terms | | | | | | | | | | | | | | | | | $\alpha$ | $\beta$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | A | B | C | D | E | F | G | H | I | G | K | L | M | N | O | P | Q | | |
| 1 | standard | 0.8 | 0.44 | 0.65 | 0.004 | 0.49 | 0.56 | 1.17 | 1.17 | 1.17 | 1.1 | 0.5 | 2 | 5 | 5 | 1 | 0 | 0 | 3 | 1 |
| 2 | score12 | 0.8 | 0.44 | 0.65 | 0.004 | 0.49 | 0.56 | 1.17 | 0.585 | 1.17 | 1.1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 3 | score12 full | 0.8 | 0.44 | 0.65 | 0.004 | 0.49 | 0.56 | 1.17 | 0.585 | 1.17 | 1.1 | 0.5 | 2 | 5 | 5 | 1 | 0 | 0 | 1 | 2 |
| 4 | score12minpack | 0.8 | 0.44 | 0.65 | 0.004 | 0.49 | 0.56 | 1.17 | 0.585 | 1.17 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 3 |
| 5 | score13 | 0.6921 | 0.1754 | 0.5253 | -0.00764 | 0.53 | 0.63 | 1.322 | 0.336 | 2 | 1.883 | 0.5 | 2 | 5 | 5 | 1 | 0.571 | 0 | 2 | 1 |
| 7 | score13 | 0.6921 | 0.1754 | 0.5253 | -0.00764 | 0.53 | 0.63 | 1.322 | 0.336 | 2 | 1.883 | 0.5 | 2 | 5 | 5 | 1 | 0.571 | 0 | 1 | 1 |
| 8 | pack no hb env dep | 0.8 | 0.1 | 0.65 | 0.004 | 0.49 | 0.56 | 1.17 | 1.17 | 1.17 | 3.1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 3 | 1 |
| 6 | RosettaHoles score | The RosettaHoles scores are based on packing information about a cavity ball and the local region surrounding it, most importantly the contact surface area of atoms surrounding the cavity with respect to a sequence of probe radii. | | | | | | | | | | | | | | | | | 1 | 2 |

**Table 2 Score terms**

| Score term | Label | Description |
|---|---|---|
| fa_atr | A | lennard-jones attractive |
| fa_rep | B | lennard-jones repulsive |
| fa_sol | C | lazaridis-jarplus solvation energy |
| fa_intra_rep | D | lennard-jones repulsive between atoms in the same residue |
| fa_pair | E | pairwise electrostatics term derived from statistics on the pdb database |
| fa_dun | F | internal energy of sidechain rotamers as derived from Dunbrack's statistics |
| hbond_lr_bb | G | long range (beta or loop) backbone-backbone hydrogen bonds |
| hbond_sr_bb | H | short range (helix) backbone-backbone hbonds |
| hbond_bb_sc | I | sidechain-backbone hydrogen bond energy |
| hbond_sc | J | sidechain-sidechain hydrogen bond energy |
| dslf_ss_dst | K | distance score in current disulfide |
| dslf_cs_ang | L | csangles score in current disulfide |
| dslf_ss_dih | M | dihedral score in current disulfide |
| dslf_ca_dih | N | $C\alpha$ dihedral score in current disulfide |
| pro_close | O | proline ring closure energy |
| envsmooth | P | Statistically derived fullatom environment potential |
| atom_pair_constraint | Q | Harmonic constraints between atoms involved in Watson-Crick base pairs specified by the user in the params file |

rotamer selection is determined by the current heuristic and historical knowledge, described by the following selection equation (Equation 1):

$$r_j^* = \begin{cases} \max_{r_j \in R_i} [\tau_{ij}]^{\alpha_g} [\eta_{ij}]^{\beta_g}, & \text{if } q < q_0; \\ \text{randomly pick up } r_j \text{ from } R_i, & \text{otherwise.} \end{cases} \quad (1)$$

Where $\tau_{ij}$ is defined later in Equation 3, which denotes the useful experience accumulated by previous searches, $\eta_{ij}$ denotes the heuristic value. Let the heuristic matrix be: $H_g = \prod_{i \in n, j \in m_i} \eta_{ij}$, where $\eta_{ij}$ is the energy difference induced by residue $i$ picking up rotamer $r_j$ , which is standardised according to Equation 2.

$$\eta_{ij} = \frac{\pi}{2} - \arctan\ \Delta E. \quad (2)$$

$q_0$ tunes the bias between the two selection policies. A random probability $q$ will be generated when a rotamer is needed. Once the rotamer is picked, $r_j^*$ is inserted into the protein backbone from the position of residue $i$.

The second formula updates the pheromone matrix $T$ after all the ants have finished their work in an iteration. Let the pheromone matrix be: $T = \prod_{i \in n, j \subseteq m_i} \tau_{ij}$, where $\tau_{ij}$ is the pheromone value accumulated by residue $i$ packing rotamer $r_j$. For each $r_j$ of residue $i$ in $s_{gb}$, the value is updated using Equation 3.

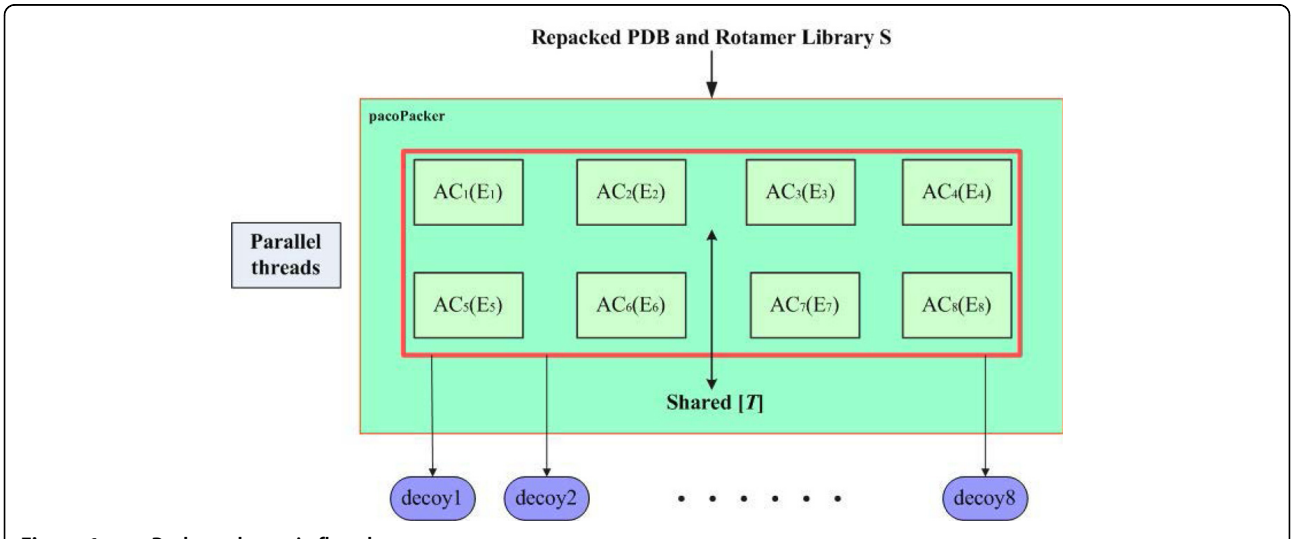



**Figure 1 pacoPacker schematic flowchart**.

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}. \quad (3)$$

Where $\rho \in [0, 1)$ is the pheromone evaporation factor, and $\Delta\tau_{ij}$ is calculated by a quality function which converts the energy value to a certain amount of pheromone. We describe this situation in Equation 4.

$$\Delta\tau_{ij} = \begin{cases} \frac{\pi}{2} - \arctan\frac{E(s_{gb})}{n}, & \text{if } r_j \text{ of residue } i \in s_{gb}; \\ \tau_{ij}, & \text{otherwise.} \end{cases} \quad (4)$$

Our SHMO scheme is simple with the help of OpenMP. The pheromone matrix is extracted from AC, and multiple colonies are run as parallel threads with private variables in each colony to co-evolve with the common pheromone matrix.

### Rotamer minimization

Rotamer minimisation was implemented in two ways. First, the pacoPacker runs on each normal rotamer as it is placed; after that, the pacoPacker runs a global minimisation on the side chains at all the packable positions. We will not provide much detail about this method, as the Rosetta3.4 mechanism was adopted to achieve it. Second, pacoPacker runs a gradient minimisation on each rotamer as it is placed and keeps the minimised rotamers. To use this second method, we devised a new data structure to remember minimised rotamers (Figure 2). If there are $M = m_1 + m_2 + \cdots + m_n$ rotamers, and each normal rotamer has its own alternative obtained by minimising itself, they are called subrotamers. We describe the set of subrotamers for $r_j$ from $R_i$ as $A_{ij}$ , which can be calculated quantitatively by Equation 5, where $i \in n, j \in m_i, r_j \in R_i$

$$\begin{cases} A_{ij} = \{r_j\} \\ A'_{ij} = \{\min(r_j), r_j\} \\ \vdots \quad \vdots \qquad \vdots \\ A^n_{ij} = \{\min(RandomP(A^{n-1}_{ij})), A^{n-1}_{ij}\} \end{cases} \quad (5)$$

A detailed explanation of this equation is shown in Figure 3. An ant selects the rotamer $r_j$ for the $i^{th}$ residue based on Equation 1, then find its subrotamers $A_{ij}$ as shown in step 5 in Figure 3, and randomly picks up a

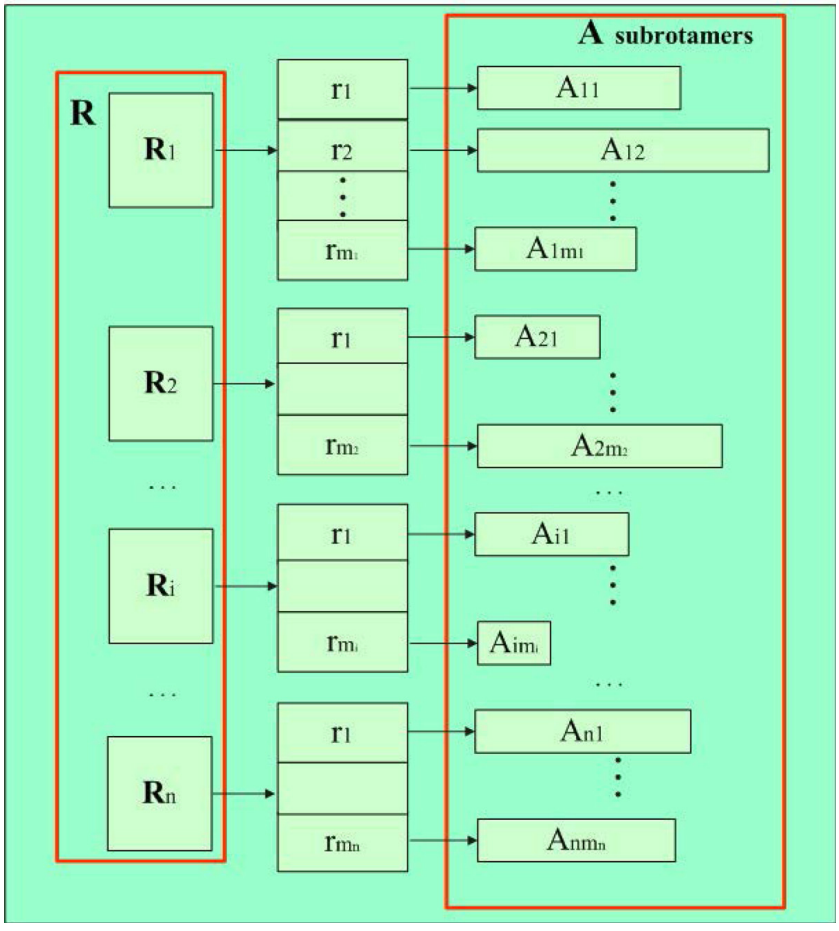



**Figure 2** Data structures of rotamers and subrotamers.

**input** : the $x$th ant construction $M_x$.
**output**: the $x$th ant which has new side-chain.
randomly sorting residue positions;
**for** $i \leftarrow 1$ **to** $n$ **do**
  choose rotamer $r_j$ based on equation (1);
  $u = r_j$;
  **if** $A_{ij}$ *is not empty* **then**
    randomly pick up a subrotamer $z$ from $A_{ij}$;
    $u = z$
  **end**
  $u =$ minimization($u$);
  **if** *subrotamer* $u$ *is accepted* **then**
    $A_{ij} \leftarrow u$;
    $r_j = u$;
  **end**
  $M_x \leftarrow$ SubstituteF($r_j$, $M_x$).
**end**

**Figure 3 Ant constructed side chains by minimising each placed rotamer**.

subrotamer from $A_{ij}$ to replace the primary rotamer at position *i*. The $9^{th}$ step attempts to optimise the subrotamer achieved by Rosetta. All minimisation algorithms in Rosetta choose a vector as the descent direction, determine a step along that vector, then choose a new direction and repeat [31]. We selected "dfpmin" as an exact line search for these steps. If this minimised subrotamer results in a drop in energy, it was kept and made into the residue *i*. Minimisation needs more time, so for researches with sufficient time who want to obtain more accurate results, this application would be a good choice.

## Results

The principal idea behind pacoPacker is to make the parallel ant colonies share only one pheromone matrix, which can combine different energies to guide each ant in constructing protein side-chain conformations. We tested pacoPacker by making comparisons with two popular side-chain modelling programs, CIS-RR and SCWRL4. CIS-RR combines a novel clash-detection guided iterative search (CIS) algorithm with continuous torsion space optimisation of rotamers (RR) [26]. SCWRL4 is an improved version of SCWRL3 [25] which uses the new rotamer library, more efficient search algorithms and a soft Vander Waals potential plus hydrogen bonding based scoring function [15]. All these predictors are based on discrete rotamers.

### Experimental settings

We performed all the tests on a computer cluster containing 20 nodes with 16-core 1.9 GHz AMD Opteron CPU per node under Linux 2.6.18 and GCC 4.1.2. CIS-RR and SCWRL4 were ran using their default settings to produce one prediction for each test instance. We ran pacoPacker, with eight ant colonies running in parallel, on the same test instances. As all these threads were synchronised to work out eight predictions and each is a nondeterministic approach, different numbers of decoys for each test instance were generated. The number of predictions for each test instance ranged from 2130 ([PDB:1CBN] 46 residues) to 4650 ([PDB:1B9O] 635 residues). We selected the highest accuracy rate of each test instance from pacoPacker to compare with CIS-RR and SCWRL4.

The benchmark instances were taken directly from other research, which contained 442 protein targets with lengths of 46 to 1184 amino acid residues [26,15]. Because [PDB:2QOL] cannot be predicted by CIS-RR and [PDB:1G8Q] is considered as a missing main chain atom by Rosetta, we excluded them from this benchmark. A fair evaluation is a difficult task, so we used two criteria to assess the accuracy of side chain packing. One was defined as the percentage of correctly predicted $\mathcal{X}_1$ and $\mathcal{X}_{12}$ angles within thresholds of 40° and 20° compared with the native structures. The second criterion was the root mean square deviation (RMSD) of the side-chain heavy atoms [34]. Both evaluation methodologies are adapted from third-party software [26,35], where they consider residues with symmetric terminal groups, or with a possibly flipped terminal group.

### Protein chain based evaluation performance

Firstly, we compared pacoPacker with CIS-RR and SCWRL4 in side-chain modelling. As shown in Table 3 for the accuracy improvement in terms of correct $\mathcal{X}$ dihedral

**Table 3 Comparison of pacoPacker, CIS-RR and SCWRL4 in the 442 structure set**

| Method | $\mathcal{X}_1(40^\circ)$ | $\mathcal{X}_1(20^\circ)$ | $\mathcal{X}_{12}(40^\circ)$ | $\mathcal{X}_{12}(20^\circ)$ | RMSD (Å) |
|---|---|---|---|---|---|
| SCWRL4 | 82.80% | 79.61% | 74.98% | 68.21% | 2.07 |
| CIS-RR | 84.88% | 82.07% | 77.13% | 70.13% | 1.62 |
| pacoPacker | 87.19% | 83.53% | 77.11% | 70.02% | 1.60 |

angles and RMSD, pacoPacker is comparable to the recently developed side-chain programs. As SCWRL4 showed relatively poor performance, so we only present a detailed comparison between pacoPacker and CIS-RR. Within 40°, the $\mathcal{X}_1$ of the whole protein was improved by 2.31% with pacoPacker (87.19% by pacoPacker versus 84.88% by CIS-RR), and the $\chi 12$ was comparable (77.11% by pacoPacker versus 77.13% by CIS-RR). A similarly consistent trend was also seen for the accuracy rate of $\mathcal{X}_1$ and $\mathcal{X}_{12}$ within 20°. In case of the other metrics, pacoPacker is the best with its lowest RMSD.

We made further comparisons between the three predictors. In Figures 4 to 7, each symbol represents a single protein target, a red cross denotes a better pacoPacker yield and a blue criss-cross denotes a worse yield. Some differences between the two methods were less than 0.5% for the accuracy of $\mathcal{X}$ dihedral angles and 0.005Å for RMSD, respectively. These are denoted by a green asterisk. As shown in Figures 4 and 6, when compared with CIS-RR, there were 342, 210 and 242 targets predicted by pacoPacker for $\mathcal{X}_1$, $\mathcal{X}_{12}$ and RMSD respectively, showing that it has the advantage over CIS-RR. Moreover, Figures 5 and 7 show that pacoPacker was better than SCWRL4 for 332, 211 and 267 targets for $\mathcal{X}_1$, $\mathcal{X}_{12}$ and RMSD respectively. These results clearly show that pacoPacker has a high reliability based on SHOP.

### Individual residues based evaluation performance

Next, we sought to evaluate how pacoPacker works on different types of amino acids. Figure 8 shows that pacoPacker improved the percent correct of both $\mathcal{X}_1$ and $\mathcal{X}_{12}$ dihedral angles. For $\mathcal{X}_1$, excluding Ala and Gly, pacoPacker has 15 types of amino acids holding the top spot. In Glu, Lys and Ser, they had an average increase of more than 5%. PacoPacker made the greatest contribution to the accuracy of $\mathcal{X}_1$. It also can be proven from the situation that pacoPacker made the greatest contribution to the accuracy of $\mathcal{X}_1$ via its accurate prediction of Ser and Thr. The residues, which were predicted accurately, were predominantly aliphatic and aromatic residue types. For $\mathcal{X}_{12}$, pacoPacker accounted for 6 types of amino acids in the lead, whilst CIS-RR accounded for 5 and SCWRL4 accounted for 3. Previous research has shown that for the short polar amino acids (Asp, Asn and Ser), CIS-RR shows lower performance, which could be due to the difference in scoring functions [26]. However, pacoPacker improves them both in $\mathcal{X}_1$ and $\mathcal{X}_{12}$, which has again shows the importance of combining different energies.

### Effects of rotamer minimisation

From the results presented in the previous two sections, we show the superiority of $\mathcal{X}_1$ while the performance of $\mathcal{X}_2$ is not strong. For example, when compare the number of red crosses on Figure 4(A) with Figure 4(B), pacoPacker has 342 best-performing proteins for $\mathcal{X}_1$, which is more than the 210 best-performing proteins for $\mathcal{X}_{12}$. In addition, Cys, Ser, Thr and Val only on wing $\mathcal{X}_1$, clearly dominate the area of $\mathcal{X}_1$. High quality $\mathcal{X}_1$ is significant for side-chain prediction, because it is a foundation of residue. On the other side, there is still room for improvement of $\mathcal{X}_2$, so we naturally optimised each

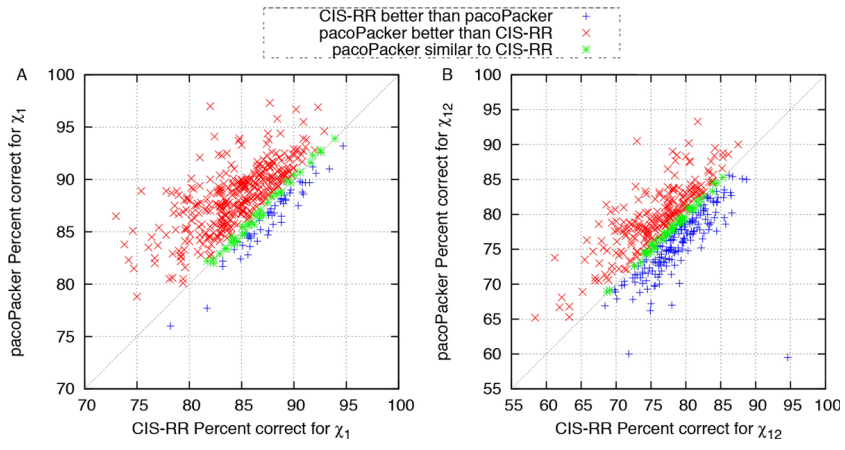



**Figure 4 Comparison of pacoPacker and CIS-RR for $\mathcal{X}$ angles in the 442 structures set**. Each symbol corresponds to a single protein target. Red crosses (blue criss-crosses) denote pacoPacker yields better (worse) results. Targets marked by green asterisks mean pacoPacker is comparable with CIS-RR. All $\mathcal{X}$ angles are within 40°.

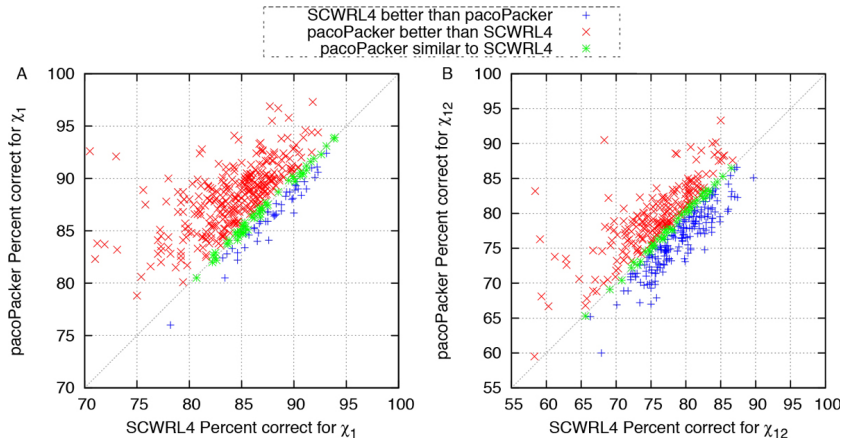



**Figure 5 Comparison of pacoPacker and SCWRL4 for $\mathcal{X}$ angles in the 442 structures set.** Each symbol corresponds to a single protein target. Red crosses (blue criss-crosses) denote pacoPacker yields better (worse) results. Targets marked by green asterisks mean pacoPacker is comparable with SCWRL4. All $\mathcal{X}$ angles are within 40°.

rotamer as it was placed (rotamer minimization). An overview of how this method performs is given below.

Figure 9 shows the effects of minimisation by comparing RMSD among three different models, and test instances is randomly from the benchmark as above. Model 1 (blue asterisk) uses gradient minimisation on each rotamer when it is placed (the method presented in this paper), model 2 (red solid box) packs the same way as model 1 but then runs a global minimisation on the side chains at all packable positions, and model 3 (green box) with normal rotamers is optimised by global minimisation only. Figure 9 shows that models 1 and 2 both decrease the RMSD compared with model 3, which means that our method can contribute to the quality of repacking. Most of time model 1 is comparable with model 2, so we can only use our method to gain optimisation as well as global minimisation. However, there were 18 proteins (data not shown), which had higher RMSD predicted by rotamer minimization. These can be classified into two groups: Those which already have high accuracies of $\mathcal{X}_1$ and $\mathcal{X}_{12}$ within 20°( with approximately 80% accuracy) and those which are

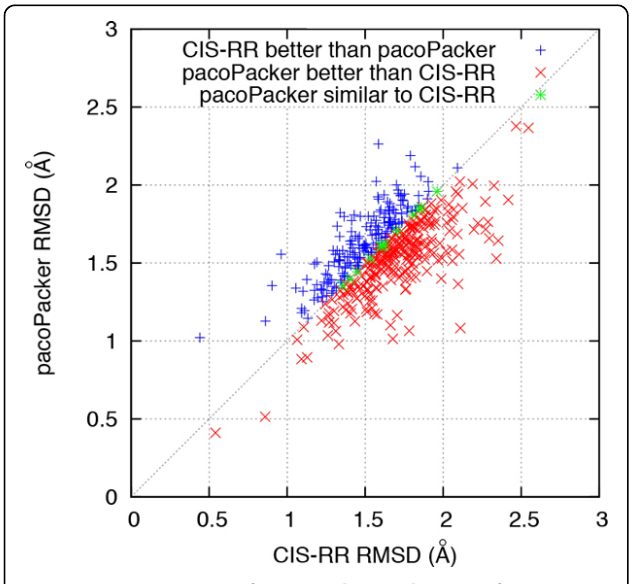



**Figure 6 Comparison of pacoPacker and CIS-RR for RMSD in the 442 structures set**. Each symbol corresponds to a single protein target. Red crosses (blue criss-crosses) denote pacoPacker yields better (worse) results. Targets marked by green asterisks mean pacoPacker is comparable with CIS-RR.

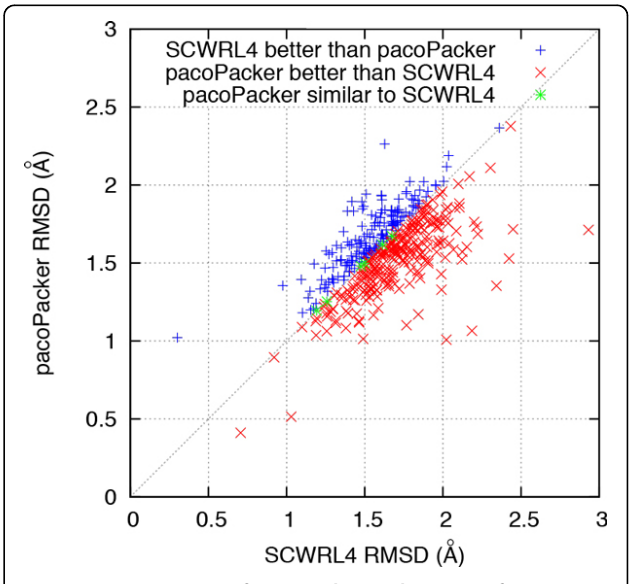



**Figure 7 Comparison of pacoPacker and SCWRL4 for RMSD in the 442 structures set**. Each symbol corresponds to a single protein target. Red crosses (blue criss-crosses) denote pacoPacker yields better (worse) results. Targets marked by green asterisks mean pacoPacker is comparable with SCWRL4.

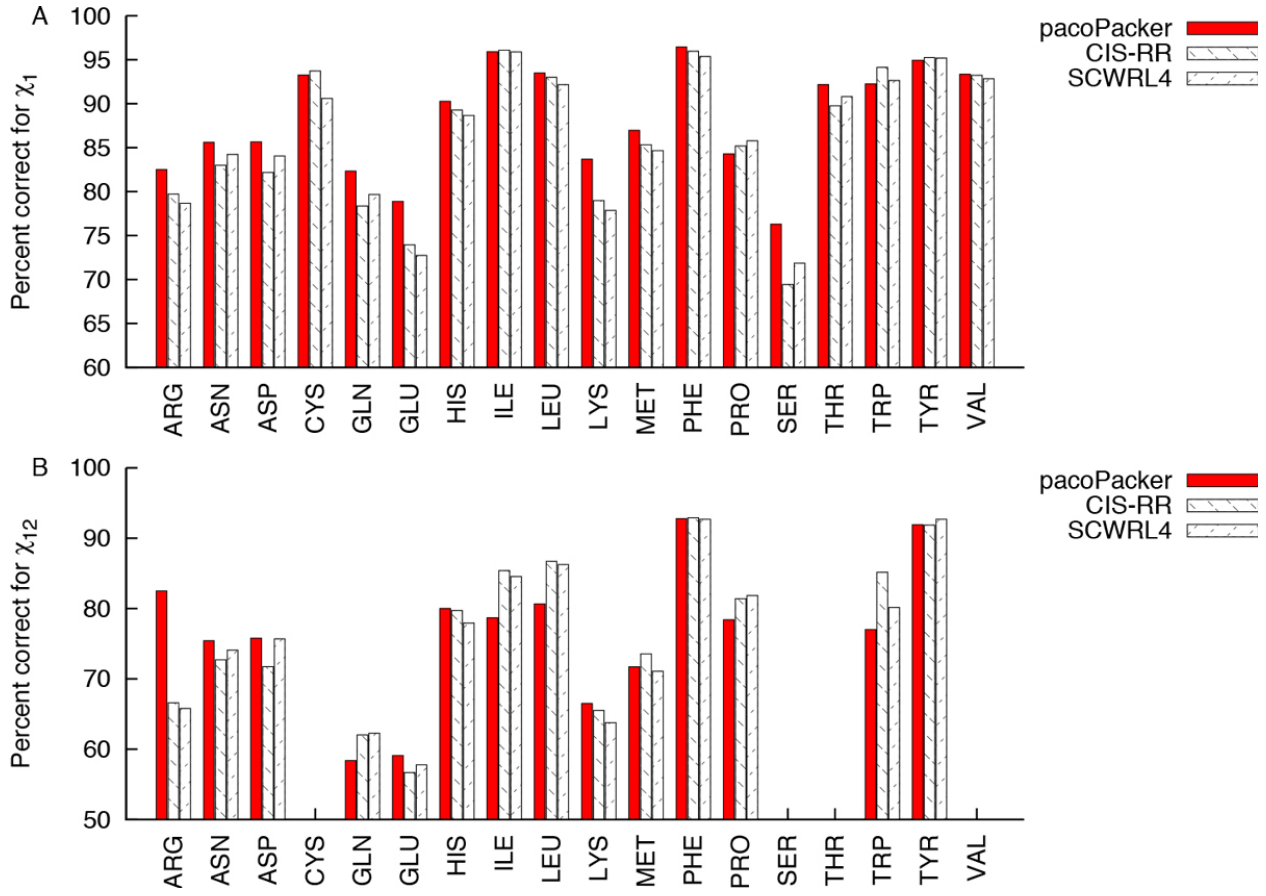



**Figure 8 Analysis of the performance of pacoPacker, CIS-RR and SCWRL4 on different amino acids**. (A) Percent correct within 40° for the $\mathcal{X}_1$ angle. (B) Percent correct within 40° for the $\mathcal{X}_{12}$ angle. The test data used the 442 proteins set. Red, back slash and forward slash indicate the test results of pacoPacker, CIS-RR and SCWRL4, respectively.

large in size, including [PDB:2OTU] (976 residues), [PDB:1OK7] (739 residues), [PDB:1YTL] (631 residues), [PDB:2EPI] (388 residues). This means that structural integrity is important for proteins that are large in size, because rotamer minimisation cannot play a full role.

## Discussion

Under the inaccuracy/usefulness property hypothesis, SOP is not an ideal computational model for protein structure prediction [28]. This means that even if the corresponding SOP is completely solved, the SOP answer may not be correct, and in most cases it will not be perfect. PacoPacker proposes a novel hybrid parallel approach to repack protein side chains based on SHOP [28,30].

Table 4 shows the distribution of best conformations for each protein from pacoPacker on different threads. The best conformations are constructed on different threads, where each energy is very useful in some specific sense, but is inaccurate in a universal sense. Therefore, we need an approach based on MOP. For using MOP to solve protein structure prediction problems, the Pareto-based approach, which focuses on the dominance analysis of the solutions found by the search, will probably result in a large Pareto front with solutions where no single energy function can be dominant. PacoPacker is different as it does not construct a Pareto front, but collects the best solutions found by parallel search procedures directed by different energy functions. The SHOP strategy was proposed as a useful parallel ACO method [30]. Using SHOP, these multiple colonies of pacoPacker can exchange their search experiences asynchronously and co-evolve towards better solutions while each colony is guided by its own objective function and algorithm parameters [28]. In 442 structures test set, the close half targets of pacoPacker maintain optimum accuracy, unlike that in the other two programs. Why does the pacoPacker approach have a good performance?

Firstly, from the view of an individual colony, the pheromone matrix accumulates the search experience of ants, which describes which rotamer should be a priori considered as the choice for each residue. Such an experience bias is established by evaluating the conformations found

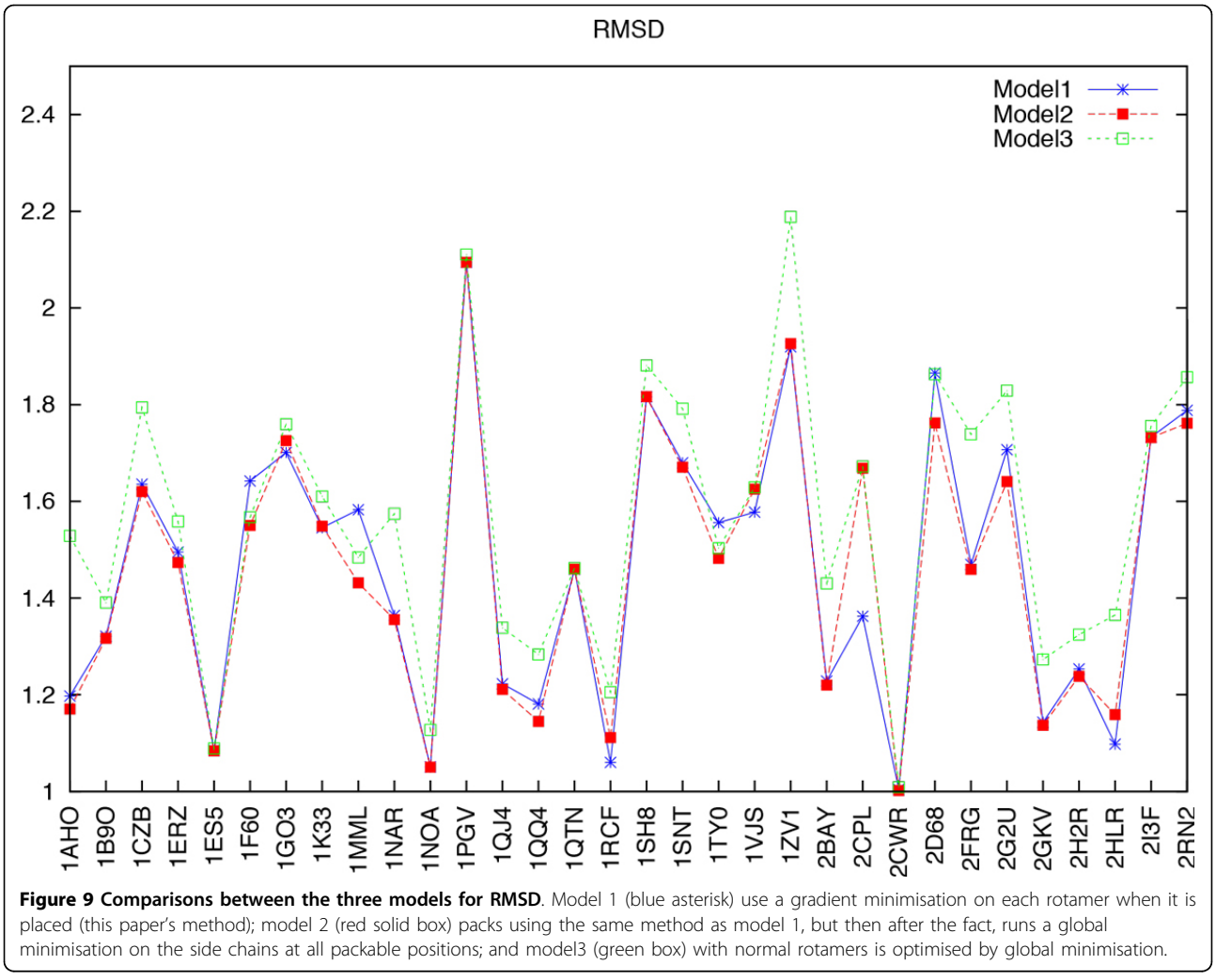



**Figure 9 Comparisons between the three models for RMSD**. Model 1 (blue asterisk) use a gradient minimisation on each rotamer when it is placed (this paper's method); model 2 (red solid box) packs using the same method as model 1, but then after the fact, runs a global minimisation on the side chains at all packable positions; and model3 (green box) with normal rotamers is optimised by global minimisation.

by the previous generation of ants using the corresponding energy function. Then by sharing $T$, each colony can achieve different search experiences from other colonies asynchronously, and each colony is also directed by their own energy functions to co-evolve towards a better state. The process of sharing one $T$ can accumulate the search experience of all parallel ant colonies and propagate the bias among them. As the pheromone matrix $T$ provides an indeterministic bias for all the running colonies, it may be easier to find better solutions.

For example, [PDB:2FLU] was one of the most accurate predictions from paco-Packer with a RMSD of 0.98, while the second most accurate prediction was 1.33 from CIS-RR. The best conformation appeared in the $27^{th}$ generation of thread 8, which ends on this generation. The other threads ended incrementally after the $29^{th}$ generation. In this situation, almost all threads stop at the same time, which gives pheromone matrix $T$ enough time to learn experiences fairly from different threads. There were some poor solutions, such as [PDB:1WVH] where the RMSD was increased by 1.23 with pacoPacker. In this case, the best conformation of pacoPacker was structured by thread 6 on the $40^{th}$ generation, and other threads stopped after $25^{th}$ generation. This may be because some threads accomplish too early so that the pheromone matrix $T$ learns search experiences with bias, which may be solved with more time. From a user perspective, we summarise when pacoPacker performs well in Table 5. This shows that the proportion of proteins repacked increased as the sequence length decreased. Therefore pacoPacker can

**Table 4 Best conformations of pacoPacker distributed on different threads**

| ID | Thread0 | Thread1 | Thread2 | Thread3 | Thread4 | Thread5 | Thread6 | Thread7 |
|---|---|---|---|---|---|---|---|---|
| Quantity | 29 | 31 | 35 | 29 | 33 | 39 | 107 | 139 |

**Table 5 The proportion of proteins repacked by pacoPacker with lower RMSD compared with other predictors**

| Sequence Length | Number | CW CIS-RR | CW SCWRL4 | CW both |
|---|---|---|---|---|
| >500 | 53 | 28.3% | 30.2% | 13.2% |
| 500~400 | 34 | 41.2% | 38.2% | 20.6% |
| 400~300 | 62 | 56.5% | 62.9% | 41.9% |
| 300~200 | 108 | 59.3% | 66.7% | 51.9% |
| 200~100 | 139 | 63.3% | 67.6% | 51.1% |
| <100 | 46 | 76.1% | 76.1% | 63.0% |

The first column denotes the range of sequence length; the second column records the number of proteins; the remaining columns show the proportion of proteins packed side chains by pacoPacker with smaller RMSD than CIS-RR or SCWRL4 alone, or combined.

provide the highest accuracy for packing side chains when the sequence length is lower than 400 amino acids.

## Conclusions

In summary, pacoPacker makes each heuristic search work with its own energy function and they complement each other in a qualitative way. Different energy functions train search trajectories to obtain different search intelligences. Our parallel strategy diffuses the intelligence to all the parallel searches by SHOP, so that all ant colonies can share their accumulated hybridised intelligence. Such co-evolvement guided by multiple objective functions simultaneously has an impact on the nature folding procedure of native proteins [28]. The prediction accuracy of packing side chains was improved for most of the proteins, which proves that pacoPacker has feasibility and practical value, but at a cost of increased CPU time. However, an important reason for using pacoPacker is that it does not need training and tuning of the energy function parameters before the predictor can work.

#### Competing interests

The authors declare that they have no competing interests.

#### Authors' contributions

Q Lü designed and developed the pacoPacker framework. LJ Quan implemented and improved pacoPacker. LJ Quan, HO Li and HJ Wu performed the experiments. LJ Quan and XX Xia drafted the manuscript. All of the authors read and approved the manuscript.

#### Acknowledgements

The authors acknowledge the support received from Rong Chen for helping with the analysis of the experiments and Caixia Wang for helping with the preparation of the paper. Funder had no role in study design, data collection and analysis, decision to publish, or preparation of the paper.

#### Declarations

This study was supported by a grant from the National Natural Science Foundation of China (No. 61170125).

This article has been published as part of *BMC Bioinformatics* Volume 15 Supplement 12, 2014: Selected articles from the IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2013): Bioinformatics. The full contents of the supplement are available online at http://www.biomedcentral.com/bmcbioinformatics/supplements/15/S12.

#### Authors' details

[1]School of Computer Science and Technology, Soochow University, Suzhou, 215006, China. [2]Jiangsu Provincial Key Lab for Information Processing Technologies, Suzhou, 215006, China. [3]School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou, 215006, China.

Published: 6 November 2014

#### References

1. Smith CA, Kortemme T: **Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction.** *Journal of molecular biology* 2008, **380(4)**:742-756.
2. Davis IW, Arendall WB III, Richardson DC, Richardson JS: **The backrub motion: how protein backbone shrugs when a sidechain dances.** *Structure* 2006, **14(2)**:265-274.
3. Kingsford CL, Chazelle B, Singh M: **Solving and analyzing side-chain positioning problems using linear and integer programming.** *Bioinformatics* 2005, **21(7)**:1028-1039.
4. Gaudreault F, Chartier M, Najmanovich R: **Side-chain rotamer changes upon ligand binding: common, crucial, correlate with entropy and rearrange hydrogen bonding.** *Bioinformatics* 2012, **28(18)**:423-430.
5. Raveh B, London N, Zimmerman L, Schueler-Furman O: **Rosetta, flexpepdock ab-initio: simultaneous folding, docking and refinement of peptides onto their receptors.** *PLoS One* 2011, **6(4)**:18934.
6. Wang C, Schueler-Furman O, Baker D: **Improved side-chain modeling for protein-protein docking.** *Protein Science* 2005, **14(5)**:1328-1339.
7. Anfinsen CB, Haber E, Sela M, White F Jr: **The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain.** *Proceedings of the National Academy of Sciences of the United States of America* 1961, **47(9)**:1309.
8. Pierce NA, Winfree E: **Protein design is np-hard.** *Protein Engineering* 2002, **15(10)**:779-782.
9. Unger R, Moult J: **Finding the lowest free energy conformation of a protein is an np-hard problem: proof and implications.** *Bulletin of Mathematical Biology* 1993, **55(6)**:1183-1198.
10. Hart WE, Istrail S: **Robust proofs of np-hardness for protein folding: general lattices and energy potentials.** *Journal of Computational Biology* 1997, **4(1)**:1-22.
11. Xie W, Sahinidis NV: **Residue-rotamer-reduction algorithm for the protein side-chain conformation problem.** *Bioinformatics* 2006, **22(2)**:188-194.
12. Chazelle B, Kingsford C, Singh M: **A semidefinite programming approach to side chain positioning with new rounding strategies.** *INFORMS Journal on Computing* 2004, **16(4)**:380-392.
13. Desmet J, De Maeyer M, Hazes B, Lasters I: **The dead-end elimination theorem and its use in protein side-chain positioning.** *Nature* 1992, **356(6369)**:539-542.
14. Desmet J, De Maeyer M, Lasters I: **Theoretical and algorithmical optimization of the dead-end elimination theorem.** *Pac Symp Biocomput* 1997, **2**:122-133.
15. Krivov GG, Shapovalov MV, Dunbrack RL: **Improved prediction of protein side-chain conformations with scwrl4.** *Proteins: Structure, Function, and Bioinformatics* 2009, **77(4)**:778-795.
16. Gray JJ, Moughon S, Wang C, Schueler-Furman O, Kuhlman B, Rohl CA, Baker D: **Protein-protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations.** *Journal of molecular biology* 2003, **331(1)**:281-299.
17. Hsin JL, Yang CB, Huang KS, Yang CN: **An ant colony optimization approach for the protein side chain packing problem.** *Proceedings of the 6th WSEAS International Conference on Microelectronics, Nanoelectronics, Optoelectronics* 2007, 44-49.
18. Roitberg A, Elber R: **Modeling side chains in peptides and proteins: Application of the locally enhanced sampling and the simulated annealing methods to find minimum energy conformations.** *The Journal of chemical physics* 1991, **95(12)**:9277-9287.
19. Leach AR, Lemon AP, *et al*: **Exploring the conformational space of protein side chains using dead-end elimination and the a* algorithm.** *Proteins Structure Function and Genetics* 1998, **33(2)**:227-239.
20. Kuhlman B, Baker D: **Native protein sequences are close to optimal for their structures.** *Proceedings of the National Academy of Sciences* 2000, **97(19)**:10383-10388.

21. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, Jacak R, Kaufman K, Renfrew PD, Smith CA, Sheffler W, *et al*: **Rosetta3: an object-oriented software suite for the simulation and design of macromolecules.** *Methods Enzymol* 2011, **487**:545-574.
22. Holm L, Sander C: **Fast and simple monte carlo algorithm for side chain optimization in proteins: application to model building by homology.** *Proteins: Structure, Function, and Bioinformatics* 1992, **14(2)**:213-223.
23. Bower MJ, Cohen FE, Dunbrack RL Jr: **Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool.** *Journal of molecular biology* 1997, **267(5)**:1268-1282.
24. Dunbrack RL Jr: **Comparative modeling of casp3 targets using psi-blast and scwrl.** *Proteins: Structure, Function, and Bioinformatics* 1999, **37(S3)**:81-87.
25. Canutescu AA, Shelenkov AA, Dunbrack RL: **A graph-theory algorithm for rapid protein side-chain prediction.** *Protein science* 2003, **12(9)**:2001-2014.
26. Cao Y, Song L, Miao Z, Hu Y, Tian L, Jiang T: **Improved side-chain modeling by coupling clash-detection guided iterative search with rotamer relaxation.** *Bioinformatics* 2011, **27(6)**:785-790.
27. Kortemme T, Morozov AV, Baker D: **An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein-protein complexes.** *Journal of molecular biology* 2003, **326(4)**:1239-1259.
28. Lü Q, Xia XY, Chen R, Miao DJ, Chen SS, Quan LJ, Li HO: **When the lowest energy does not induce native structures: parallel minimization of multi-energy values by hybridizing searching intelligences.** *PloS one* 2012, **7(9)**:44967.
29. Lv Q, Wu H, Wu J, Huang X, Luo X, Qian P: **A parallel ant colonies approach to de novo prediction of protein backbone in casp8/9.** *Science China Information Sciences* 2013, **56(10)**:1-13.
30. Lv Q, Xia X, Qian P: **A parallel aco approach based on one pheromone matrix.** In *Ant Colony Optimization and Swarm Intelligence. Volume 4150.* Springer; 2006:332-339.
31. Rohl CA, Strauss CE, Misura KM, Baker D: **Protein structure prediction using rosetta.** *Methods in enzymology* 2004, **383**:66-93.
32. Dagum L, Menon R: **Openmp: an industry standard api for shared-memory programming.** *Computational Science & Engineering, IEEE* 1998, **5(1)**:46-55.
33. Shapovalov MV, Dunbrack RL Jr: **A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions.** *Structure* 2011, **19(6)**:844-858.
34. Miao Z, Cao Y, Jiang T: **Rasp: rapid modeling of protein side chain conformations.** *Bioinformatics* 2011, **27(22)**:3117-3122.
35. Eyal E, Najmanovich R, Mcconkey BJ, Edelman M, Sobolev V: **Importance of solvent accessibility and contact surfaces in modeling side-chain conformations in proteins.** *Journal of computational chemistry* 2004, **25(5)**:712-724.

doi:10.1186/1471-2105-15-S12-S5
**Cite this article as:** Quan *et al.*: Improved packing of protein side chains with parallel ant colonies. *BMC Bioinformatics* 2014 **15**(Suppl 12):S5.