

# 偏序约简框架下增量自学习推理:税务稽查的例子<sup>\*</sup>)

汤胤<sup>1</sup> 彭宏<sup>2</sup> 郑启伦<sup>2</sup>

(暨南大学管理学院电子商务教研室 广州 510632)<sup>1</sup> (华南理工大学计算机学院 广州 510640)<sup>2</sup>

**摘要** 本文讨论欺诈防范领域中税务稽查的例子。在相关文献基础上分析了目前线性推理的不足,提出构造偏序约简范例集,给出了CBR循环过程中范例获取、记忆、扩容、推理等算法,由此实现范例推理机增量自学习机制。算法相比线性检索和记忆有着较高的性能和准确度,在税务稽核选案、信用卡欺诈、公司财务数据审计方面都可以有相当广泛的应用。

**关键词** 范例推理,偏序,自学习

## Incremental Self-learning Reasoning in Briefed Partially-ordered Case Base: A Tax Inspection Case

TANG Yin<sup>1</sup> PENG Hong<sup>2</sup> ZHENG Qi-Lun<sup>2</sup>

(Lab of E-Commerce, College of Management, Jinan Univ., Guangzhou 510632)<sup>1</sup>

(College of Computer Science, South China Univ. of Tech., Guangzhou 510640)<sup>2</sup>

**Abstract** Tax inspection is a special category in fraud detection area. Based on the investigation on related works, this paper discusses the weaknesses of current linear reasoning in case-based reasoning context. We propose a partially-ordered briefed case base, under which algorithms of case retrieval, retain, adaptation in CBR circle are offered. This enables an incremental self-learning mechanism on target cases. Experimental result has shown advantages over linear reasoning mechanism. This reasoning machine can be widely embedded in tax inspection, credit card fraud detection, and financial auditing applications.

**Keywords** Case-based reasoning, Partial order, Self-learning

## 1 引言

西方发达国家典型的欺诈包括信用卡欺诈、计算机入侵、电信欺诈、健康及汽车保险欺诈等<sup>[1]</sup>。就在最近(2007年1月)美国南加州警方破获号称洛杉矶地区20年来最大的保险欺诈案,上百华人涉案<sup>[2]</sup>。近些年来,由于电子商务的蓬勃发展,网络信用欺诈也逐渐成为欺诈防范领域新的研究热点<sup>[3]</sup>。在税务稽查方面,西方发达国家制度比较规范,各种纳税事务的处理包括对纳税人的监控和稽查,都可通过计算机网络进行。然而在国内,税务稽查的智能推理领域却仍然少有人涉及<sup>[4]</sup>。

近10年来学术界不仅对欺诈检测有大量的研究,同时不断在适应新的欺诈领域和方式。Chiu和Tsai对信用卡欺诈从基于Web Service的协作机制入手,提出了欺诈模式挖掘算法,对常见的信用卡欺诈方式较有效果<sup>[5]</sup>。Xu利用频繁树从交易记录中对目标行为模式特征进行管理,按规则分析进行挖掘,给出了异常行为的适应性检测技术,该技术可以准确地从正常行为中将异常行为检测出来<sup>[6]</sup>。Kumar针对计算机入侵检测,基于Colored Petri Nets设计了匹配模型,给出了入侵检测的模式匹配方案<sup>[7]</sup>。该模型将已知的人侵案例进行组织,用于目标案例进行匹配。我们知道,在范例推理过程中,尤其是无监督的自动运行中,往往会有些案例很难识别,从而被错误归类。Wheeler和Aitken则利用范例推理,采用加阈值的最近邻方法进行范例获取,并综合利用多重选择标准进行案例诊断<sup>[8]</sup>。现实中的欺诈是不断地动态变化的,并

能根据监测系统进行智能反侦查<sup>[9]</sup>。因此我们的推理机必须具备学习机制,能够适应不断变化的欺诈模式。Phua提出了基于图论的配对检测技术,在一个滑动窗口内将一定时期内的案例与目标案例相配对,将其中每个属性的距离加总,得到相似度,再考虑目标案例的出度,将目标案例归入白名单或黑名单中<sup>[10]</sup>,较好地利用了近期其他案例与目标案例的相似距离。问题在于滑动窗口按照时间顺序滑动,大小并没有足够依据,容易丢失以往有用的信息。

从方法论角度,对于上述欺诈的检测普遍是先采用一定技术衡量目标案例与白名单以及黑名单之间的距离再进行推断,推断过程包括分别对黑名单或白名单的全督导、半督导、无人督导、混合型,从而将目标案例归类并做出判断<sup>[11]</sup>。在技术方面,近些年来则更加广泛,包括主成分分析、神经网络、游离分析、决策树、贝叶斯网络、支持向量机、专家系统、基于模型推理以及数据挖掘等等。神经网络相关技术是应用最广泛的欺诈检测技术之一<sup>[1,9,11]</sup>。然而其问题在于,所产生的相似度量结果无法从根源上解释<sup>[1]</sup>。而基于数学模型的方法则面对纯数字,在计算过程中散失了数据中蕴含的丰富信息。限于篇幅,本文将只对与本文相关的最新技术进行讨论,具体可参考基于数据挖掘的欺诈检测研究文<sup>[11]</sup>。Phua对近10年来基于数据挖掘的欺诈检测研究做了较完整的概览,其将欺诈恰当归类,并从行业、数据特征、效能评估方法以及方法论方面对这些研究做了详细分析<sup>[11]</sup>。Kou则从技术角度对欺诈检测方法进行了归类<sup>[1]</sup>。

由于国情的不同,上述研究基本上未有涉及到税务稽查

<sup>\*</sup>省科技计划项目(2006B11201004)、暨南大学博士启动基金(51104653)。汤胤 博士,讲师,研究方向:商业智能、电子商务;彭宏 教授,博导,研究方向:智能计算技术;郑启伦 教授,博导,研究方向:智能计算技术、神经网络理论及应用。

领域应用。在国内,陈健建立了数据挖掘模型,详细介绍了以数据挖掘技术为核心的税务稽查选案系统的开发过程,从而达到有效地利用现有资源,提高税务稽查的科学性和准确度<sup>[4]</sup>。

本文从基于偏序约简框架给出一个范例推理的推理机,在推理过程中能够利用相关案例的疑似值计算得到目标案例的疑似值,进而不断地充实知识库,达到增量自学习的目的。

## 2 偏序约简框架的构造

传统的范例组织和存储基本上为线性模式,保存在数据库或其他方面。搜索的时候几乎遍历整个范例库。范例库属性有时候多达几十个,在范例数量增长的情况下,会带来搜索和索引的巨大系统负荷。如何能快速找到目标范例并进行匹配,是当前研究领域的热点问题。

我们知道,现实中的相似并不像“ $1 < 2, 2 < 3$ , 那么  $1 < 3$ ”那样的线性关系,而是呈现偏序相似<sup>[3,12]</sup>。 $c_i$  为范例库  $C$  中任意的一个范例,  $c_i \in C$ 。假设某应用中范例有  $N$  个属性,即  $\langle c_i < a_{i1}, a_{i2}, a_{i3}, \dots, a_{iN} \rangle$ 。

设  $a_1 \max, a_2 \max, a_3 \max, \dots, a_n \max$  分别为范例集中  $a_{11}, a_{12}, a_{13}, \dots, a_{1n}$  可能的最大值,构造包含这些最大值的范例  $c^{\max}$ ; 设  $a_1 \min, a_2 \min, a_3 \min, \dots, a_n \min$  分别为范例集中  $a_{11}, a_{12}, a_{13}, \dots, a_{1n}$  可能的最小值,构造包含这些最小值的范例  $c^{\min}$ 。本文的想法是将案例库按照约简后的偏序来组织,在范例检索的时候,寻找其偏序位置,即可得到其周围相似的案例作为源案例集。分值的计算推迟到最后推理阶段才进行,保存了案例中丰富的信息。

### 2.1 构造约简案例集

**定义 1** 约简属性集 将  $[a_i^{\min}, a_i^{\max}]$  等分为  $M-1$  个区间,得到  $M$  个元素的集合:

$A_i = \{a_{i1}, a_{i2}, a_{i3}, \dots, a_{iM}\}$  其中  $a_{i1} = a_i^{\min}, a_{iM} = a_i^{\max}$ 。 $A_i$  称为关于属性  $i$  的约简属性集。

**定义 2** 约简案例集 对于案例的  $N$  个属性,它们的约简属性集的笛卡尔乘积为

$$A = A_1 \times A_2 \times A_3 \times \dots \times A_N$$

称为约简案例集,其中的元素称为约简案例。

我们将这个构造约简案例集的过程形式化表述为定义为  $A = \text{Construct}(c^{\min}, c^{\max}, M)$ 。

**定义 3** 约简案例集  $A$  上的关系  $R$  为,对任意的  $x, y \in A, x = \langle a_1, a_2, \dots, a_N \rangle, y = \langle b_1, b_2, \dots, b_N \rangle, xRy \Leftrightarrow a_1 \leq b_1$  and  $a_2 \leq b_2$  and  $a_3 \leq b_3, \dots, a_N \leq b_N$ 。记为符号  $<$ 。若不存在约简案例  $z \in A$ ,使得  $x < z < y$ ,则称  $y$  完全覆盖  $x$ 。很显然  $\langle A, R \rangle$  是个偏序集(证明略)。

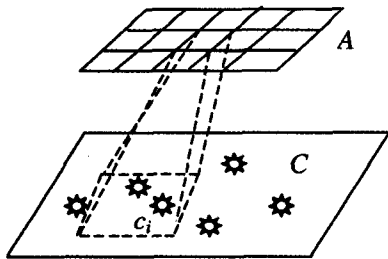


图 1 建立约简案例集  $A$  到实际案例集  $C$  的映射

**定义 4** 约简上(下)界 设实际案例  $\forall c, c \in C, c_i < \langle a_{i1}, a_{i2}, \dots, a_{ik}, \dots, a_{iN} \rangle$ 。根据约简属性集的定义,对案例  $c_i$  任意

的属性  $a_{ik}$ ,总是能在关于属性  $i$  的约简属性集  $A_i$  中找到一个区间,使得  $a_{ij} < a_{ik} < a_{i,j+1}$ 。又由于约简案例集是约简属性集的笛卡尔乘积,很显然,总是能找到  $\forall x, y, x, y \in A$ ,使得  $x < c_i < y$ ,并且  $y$  完全覆盖  $c_i, c_i$  完全覆盖  $x$ 。约简案例  $y, x$  称为实际案例  $c_i$  在约简案例集  $A$  中的约简上(下)界。注意,此时  $x, c_i$  和  $y$  三者本身为线性关系,这对我们在更小粒度上进行线性查找有很大帮助。

在实际中,案例有时候会十分集中,在部分约简上下界会充满相当多的案例。考虑到这个情况,我们可以将该约简上下界作为新的约简案例集边界,递归地划分区间。分裂过程仍与首层划分一致。此时案例的检索和保持会复杂一些。在实际应用中,我们将  $[a_i^{\min}, a_i^{\max}]$  进行了规范化(Normalization),将其映射到  $[0, 100]$  区间,并等分为 10 个区间,得到 11 个约简值。这样案例约简算法会非常快速高效得到约简上(下)界。

### 2.2 案例定位和检索

有了前面的工作,范例的定位十分简单,只需找到其约简范例即可定位。其过程包括对案例  $c$  的约简,得到约简上下界  $x, y$ ,形式化表述为  $\langle x, y \rangle = \text{Locate}(c)$ 。

#### 算法 1 案例检索 retrieval( $c, C$ )

Input: 案例  $c$ , 约简范例集  $A$ , 实际案例集  $C$   
Output: 案例集  $C'$   
(1)  $\langle x, y \rangle = \text{Locate}(c, A)$   
(2) while  $x, y$  为子约简范例集边界  
(3)  $M =$  取得该集合等分参数  
(4)  $A' = \text{Construct}(x, y, M)$   
(5)  $\langle x, y \rangle = \text{Locate}(c, A')$   
(6) end  
(7)  $C' =$  find all  $c_i$ , satisfy  $x < c_i < y$  //取得以  $x, y$  为约简上下界的所有实际案例,得到源案例集  $C'$ 。  
(8) return  $C'$

以这种方法获取的源范例集  $C'$  的特点是,其中所有的元素与目标案例  $c$  有很高的相似度,这给源范例项目标范例映射提供了很高的质量。之所以提取源案例集而非提取单个源范例,进行两阶段案例获取,好处是:(1)可以在不同阶段综合使用不同标准来对源范例集进行考察,从而选择最合适的范例进行转换;或者(2)综合计算该源范例集的数据,然后对目标范例进行推理。总之,两阶段案例获取可使得范例重用十分灵活。我们用的是第二种方案。

### 2.3 记忆

案例记忆事实上是将案例在一个比较小粒度上插入适合位置的过程。将案例  $c$  插入  $\langle x, y \rangle$  从而形成  $x < c_i < y$  的线性,我们把此过程形式化表达为  $\text{insert}(c, x, y)$ 。然而事情没那么简单,我们不但要考虑原有案例占位情况,还必须考虑递归分裂的情况。一般范例保留可在范例获取之后进行,这样我们可以利用取得的案例集  $C'$ 。

#### 算法 2 insert( $c, A$ ) $c$ 为满足保存条件的案例。

Input: 案例  $c$ , 约简范例集  $A$   
(1)  $C' = \text{retrieval}(c, C)$  //  
(2) if  $C' = \emptyset$  then  
(3)  $\text{insert}(c, x, y)$   
(4) return  
(5) matched = false  
(6) for each  $c'_i$  in  $C'$   
(7) if  $c'_i < c$  then //完全覆盖  
(8)  $\text{insert}(c, c'_i, y)$   
(9) matched = true;  
(10) if  $c < c'_i$  then //完全覆盖  
(11)  $\text{insert}(c, x, c'_i)$   
(12) matched = true;  
(14) end  
(15) if matched=false then  
(16)  $\text{insert}(c, x, y)$   
(17) return

我们将范例库是否需要扩容形式化表达为 overload

( $C'$ ),具体过程可以简单地通过源案例集的元素数量来判断其是否需要分裂。

**算法3** split( $x, y, M'$ )

Input:需分裂的边界  $x, y$ ,源案例集  $C'$ ,分裂参数  $M'$   
 Output:子约简范例集  $A'$   
 (1) if overload( $C'$ ) then  
 (2)  $A' = \text{Construct}(x, y, M')$   
 (3) for each  $c$  in  $C'$   
 (4) insert( $c, A'$ )  
 (5) end

**2.4 推理过程**

传统范例推理的类映射过程是选取单个源范例做类比转换。与传统问题解决型推理不同的是,在我们税务稽查选案的应用中目的是判定目标范例是否有问题,一般来说只有“真”、“假”、“疑似”3种答案。所以,本应用里,以多个源范例进行综合计算,得到目标范例的疑似分值,比较其与指定风险阈值的大小。这样不断通过以往的案例疑似值来得到当前案例的疑似值,称为增量自学习机制。为了达到更精细的结果,我们还将获取到的源案例集  $C'$  进一步划分,目的是对可比源案例赋予更高的权重。设集合:

$$C'_1 = \{c_i | c_i < c \text{ 或 } c < c_i, c_i \in C', 1 \leq i \leq m\} \quad (1)$$

这里子集  $C'_1$  为  $C'$  中与  $c$  有线性关系(可比)的元素的集合,有  $m$  个元素。再设子集  $C'_2 = C' - C'_1$  为不包含与案例  $c$  可比的元素的集合,有  $n$  个元素。

**定义5** 对于进入源案例库的案例  $c$ ,疑似值  $S(c)$  定义为

$$S(c) = \frac{\sum_{c_i \in C'_1} \alpha S(c_i)}{m} + \frac{\sum_{c_j \in C'_2} (1-\alpha) S(c_j)}{n} \quad (2)$$

这里,  $\alpha$  为用户定义的权重,设为 0.70。

也就是说,如果与目标案例类似的案例都是问题案例的话,我们就有充分理由怀疑目标案例为问题案例,这也是符合人的思维规律的。定义两个风险阈值  $\delta_1, \delta_2, \delta_1 < \delta_2$ 。

**定义6** 合法案例定义为  $S(c) < \delta_1$

**定义7** 问题案例定义为  $S(c) > \delta_2$

**定义8** 疑似案例定义为  $\delta_1 < S(c) < \delta_2$

疑似案例可提交人工干预,由专家对疑似案例进行诊断并给出疑似调整值。推理机获得调整值后,再将该案例插入知识库中。

**4 实验结果**

实验目标是考察同等大小范例库和同样的相似度量方案下,偏序框架检索与线性检索机制的性能差异。范例来自某市税局 2005 年 1 年的稽查数据,20 万条。为了简便起见并去除商业敏感信息,我们经过约简和计算字段,只提取了其中的 6 个字段,包括进项、销项税额比较、销售税金负担率、加成率分析、销售利润率变化幅度(横向比较)、销售利润率变化幅度(纵向比较)。在对比实验中,我们线性查找案例库,对案例库中每个案例都进行欧式距离相似度计算,从而得到欧式距离低于某个阈值  $\delta$ (相似度最高)的案例集。采集系统时间可以得到两种算法的时间差异。而在本实验中,除了查找过程外,其余环境与对比实验中一样。

实际案例库保存在 mysql5.0 数据库中,约简案例集只在有需要索引的时候才生成记录,并指向相应实际案例,无须单独存表。每层分裂参数  $M$  各设为 11, 6, 3。由 java 代码运行得到结果,系统平台为 Windows XP, Pentium 866, 512 兆内存。为了消除其他代码运行本身对系统资源的消耗对实验

的影响,放大系统时间的差异,对同样过程我们循环了 10 次。由此对比两个算法对系统资源的消耗情况。

表 1 同样范例库下偏序检索与线性检索对比

次	比率(偏序耗时/线性耗时)	性能提高
1	0.672	32.80%
2	0.718	28.20%
3	0.740	26.00%
4	0.588	41.20%
5	0.781	21.90%
6	0.801	19.90%
7	0.745	25.50%
8	0.699	30.10%
平均	0.718	28.20%

在另外一个实验中,我们考察在不同大小范例库下对偏序框架检索性能的影响。结果如图 2 所示。

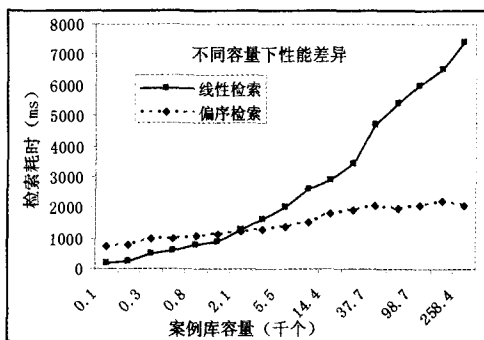


图 2 不同大小范例库偏序检索的性能差异

对于判别的准确度,我们采用常用的 true positive rate (tpr)和 false positive rate (fpr)以及二者的比值 Receiver Operating Characteristic(roc = tpr : fpr)<sup>[11]</sup> 作为指标。

表 2 范例预测效果差异

	tpr	fpr	roc
线性检索	84.22%	8.72%	965 : 1
偏序检索	88.19%	6.23%	1415 : 1

从实验数据来看,偏序框架下 tpr 略高而 fpr 略低,说明正确报告问题案例的比例提高,同时误报比例降低,说明得到的源范例集相比线性检索有更高的质量。偏序检索性能比线性检索平均提高了 28.2%。而且,偏序范例检索随着案例库的增长系统资源消耗呈现减速递增的状况,而线性检索方式则呈现线性递增的情况。另外,偏序框架下范例检索并非总是高效率的,在案例库数量较少(<2000)的情况下,采用线性查找方式会获得更高的效率。

**结论** 本文在偏序框架下,构造约简范例集,给出了范例检索和获取的算法,讨论了 CBR 循环过程中检索、记忆、扩容、推理等过程,提出了范例推理机增量自学习机制,算法相比线性检索和记忆有着较高的性能和准确度,在税务稽查选案、信用卡欺诈、公司财务数据审计方面都有着广阔应用。

**参考文献**

1 Kou Y, Lu C, Sirwong Wattana S, et al. Survey of fraud detection techniques. In: Proc. of the 2004 International Conference on Networking, Sensing, and Control, 2004. 749~754

# 一种具有随机邻居元胞自动机的混沌特性<sup>\*</sup>

刘向东<sup>1,2</sup> 张俊星<sup>1</sup> 焉德军<sup>1</sup> 何希勤<sup>2</sup>

(大连民族学院非线性信息技术研究所 大连 116600)<sup>1</sup> (辽宁科技大学理学院 鞍山 114004)<sup>2</sup>

**摘要** 研究了一种具有随机邻居的 2 值元胞自动机模型的动力学性质,给出了其理想状态的动力学模型,分析了该模型的混沌特性,并通过分叉图、Lyapunov 指数和 Schwarzian 导数解释了模型由倍分叉通向混沌的过程。最后,通过计算对比,分析了非理想状态与理想状态下模型动力学性质的差异。

**关键词** 元胞自动机,混沌,分叉图, Lyapunov 指数

## Chaos of a Kind of Cellular Automaton with Random Neighborhoods

LIU Xiang-Dong<sup>1,2</sup> ZHANG Jun-Xing<sup>2</sup> YAN De-Jun<sup>1</sup> HE Xi-Qin<sup>2</sup>

(The Research Institute of Nonlinear Information Technology, Dalian Nationalities University, Dalian 116600)<sup>1</sup>

(Faculty of Science, Liaoning University of Science and Technology, Anshan 114044)<sup>2</sup>

**Abstract** The dynamical characters of a kind of two value cellular automaton with random neighborhoods are studied. The dynamical model of the automaton in the ideal condition is given out, and the chaotic properties of the model are analyzed. The bifurcation plot, the Lyapunov exponents, and the Schwarzian derivative of the model are calculated to explain that the route to chaos the model takes is period-doubling bifurcations. Finally, the different behaviors between the ideal model and non-ideal model are pointed out.

**Keywords** Cellular automaton, Chaos, Bifurcation diagram, Lyapunov exponent

## 1 引言

元胞自动机构建简单,物理意义直观,构成方式繁杂,变种多样,行为复杂,因此成为模拟复杂现象、解决复杂问题的有力工具<sup>[1,2]</sup>。元胞自动机尤其适合简单机制引发的复杂现象的原理层次的研究。正是基于对这些机制的探讨,对于元胞自动机行为分类的研究成为元胞自动机的一个重要的研究课题和核心理论,其中最具影响力的当属 S. Wolfram 在上世纪 80 年代初做的基于动力学行为的元胞自动机分类<sup>[3]</sup>。S. Wolfram 在详细研究一维元胞自动机的演化行为及大量的计算机实验的基础上,将所有元胞自动机的动力学行为归纳为四大类。

(1) 平稳型:自任何初始状态开始,经过一定时间演化后,元胞空间趋于一个空间平稳的构形。

(2) 周期型:经过一定时间演化后,元胞空间趋于一系列简单的固定结构。

(3) 混沌型:自任何初始状态开始,经过一定时间演化后,元胞自动机表现出混沌的非周期行为,所生成的结构的统计特征不再变化,通常表现出分形特征。

(4) 复杂型:出现复杂的局部结构,或者说是局部的混沌,其中有些结构会不断地传播。

从研究元胞自动机的角度讲,最具研究价值的是具有第 3、4 类行为的元胞自动机。因为第 3 类元胞自动机可以仿真很多混沌系统,而第 4 类元胞自动机具有“突现计算”功能,可以用作广义计算机以仿真任意复杂的计算过程。

S. Wolfram 还通过计算机实验近似地给出了一维元胞自动机中上述四种吸引子或模式所占的比例,见表 1。可以看出,具有一定局部结构的复杂模式出现的概率相对要小一

<sup>\*</sup>国家自然科学基金资助项目(60573124);辽宁省自然科学基金资助项目(20040948)。刘向东 教授,博士,主要研究方向为非线性信息处理和计算机网络;张俊星 副教授,博士,主要研究方向为信号处理和网络技术;焉德军 副教授,博士,主要研究方向为非线性信息处理;何希勤 教授,博士,主要研究方向为人工智能。

- 2 南加州破获保险诈骗案 逮捕逾百位华人. <http://www.chineseinla.com/t/page-viewtopic/t-4616.html>
- 3 Tang Y. Online Reputation Scoring from Historical Feedbacks Based on Partially-ordered Similarity. In: Electronic Commerce Research and Application. Elsevier, second revision. Jan. 2007
- 4 陈健. 税务稽查选案中的 KDD 应用. 系统工程, 1999, 17(5): 62~66
- 5 Chiu C, Tsai C. A web services-based collaborative scheme for credit card fraud detection. In: Proc. of 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, 2004
- 6 Xu J, Sung A H, Liu Q. Tree based behavior monitoring for adaptive fraud detection. In: 18<sup>th</sup> International Conference on Pattern Recognition, Vol. 1. Los Alamitos, CA, USA: IEEE Computer Society, 2006. 1208~1211
- 7 Kumar S, Spaford E H. A pattern matching model for misuse intrusion detection. In: Proceedings of the 17th National Computer Security Conference, 1994. 11~21
- 8 Wheeler R, Aitken S. Multiple algorithms for fraud detection. Knowledge Based Systems, 2000, 13(3): 93~99
- 9 Weatherford M. Mining for fraud. IEEE Intelligent Systems, 2002, 17(4): 4~6
- 10 Phua C, Gaylor R, Lee V. On the Approximate Communal Fraud Scoring of Credit Applications
- 11 Phua C, Lee V, Smith K, et al. A comprehensive survey of data mining-based fraud detection research. Artificial Intelligence Review.
- 12 Tang Y. Partially ordered template-based matching algorithm for financial time series. In: Proc. IEA-AIE2006. Lecture Notes in Computer Science, Vol 4031. France: Springer Berlin/Heidelberg, 2006. 1059~1070