ELSEVIER

Expert Systems with Applications 31 (2006) 56–68

Expert Systems with Applications

www.elsevier.com/locate/eswa

A process-mining framework for the detection of healthcare fraud and abuse

Wan-Shiou Yang [a,*], San-Yih Hwang [b]

[a] Department of Information Management, National Changhua University of Education, No. 1, Jin-De Road, Changhua 500, Taiwan, ROC
[b] Department of Information Management, National Sun Yat-sen University, No. 70, Lien-Hai Road, Kaohsiung 80424, Taiwan, ROC

Abstract

People rely on government-managed health insurance systems, private health insurance systems, or both to share the expensive healthcare costs. With such an intensive need for health insurances, however, health care service providers' fraudulent and abusive behavior has become a serious problem. In this research, we propose a data-mining framework that utilizes the concept of clinical pathways to facilitate automatic and systematic construction of an adaptable and extensible detection model. The proposed approaches have been evaluated objectively by a real-world data set gathered from the National Health Insurance (NHI) program in Taiwan. The empirical experiments show that our detection model is efficient and capable of identifying some fraudulent and abusive cases that are not detected by a manually constructed detection model.

Keywords: Healthcare fraud; Healthcare abuse; Clinical pathways; Classification model; Data mining

1. Introduction

Healthcare has become a major focus of concern and even a political, social, and economics issue in modern society. The medical expenditure required to meet public demand for high-quality and high-technology services is substantial. This phenomenon is likely to become more widespread and more intense due to the increasing average lifespan and decreasing birth rates of humans in many societies. People rely on health insurance systems, which are either sponsored by governments or managed by the private sector, to share the high healthcare costs.

Such an intensive need for health insurance has resulted in fraudulent and abusive behavior becoming a serious problem. According to a report (Health Insurance, 1992) published by the General Accounting Office in the US, healthcare fraud and abuse costs the US as much as 10% of its annual spending on healthcare, representing US$ 100 billion per year. Similar problems have been reported for the health insurance programs of other developed countries (Lassey, Lassey, & Jinks, 1997). The above figures indicate that detecting healthcare fraud and abuse is imperative.

Detecting healthcare fraud and abuse, however, needs intensive medical knowledge. Many health insurance systems rely on human experts to manually review insurance claims and identify suspicious ones. Most of the computer systems that are intended to help detect undesirable behavior require human experts to identify a set of features so as to develop the core of detection models. This results in both system development and claim reviewing being time-consuming, especially for the large government-sponsored national insurance programs in countries such as France, Australia, and Taiwan.

In this research, we propose a process-mining framework that utilizes the concept of clinical pathways to facilitate the automatic and systematic construction of an adaptable and extensible detection model. We take a data-centric point of view and consider healthcare fraud and abuse detection as a data analysis process. The theme of our approach is to apply process-mining techniques to gathered clinical-instance data to construct a model that distinguishes fraudulent behaviors from normal activities. This automatic approach eliminates the need to manually analyze and encode behavior patterns, as well as the guesswork in selecting statistics measures. The proposed framework is evaluated via real-world data to demonstrate its efficiency and accuracy.

This paper is organized as follows. Section 2 examines in more detail the problem of healthcare fraud and abuse. Related research efforts are also reviewed. Section 3 presents the process-mining framework of our research. Sections 4 and 5 describe in detail the methods for building the detection model. Section 6 presents the results of an evaluation of the detection power of the model on a real-world data set gathered from the

* Corresponding author. Tel.: +886 4 723 2105x7611; fax: +886 4 721 1162.
E-mail address: wsyang@cc.ncue.edu.tw (W.-S. Yang).

0957-4174/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2005.09.003

National Health Insurance (NHI) program in Taiwan. Section 7 concludes the current work and discusses the directions of future research.

2. The problem and the related work

The processing of health insurance claims involves three parties: service providers, insurance subscribers, and insurance carriers. The National Health Care Anti-Fraud Association defined healthcare fraud as 'an intentional deception or misrepresentation made by a person, or an entity, with the knowledge that the deception could result in some unauthorized benefit to him or some other entities' and healthcare abuse as 'the provider practices that are inconsistent with sound fiscal, business, or medical practices, and result in an unnecessary cost, or in reimbursement of services that are not medically necessary or that fail to meet professionally recognized standards for health care' (Guidelines to health care fraud, 1991).

The above definitions indicate that undesirable behavior can be performed by any of the three parties. However, further studies (He, Wang, Graco, & Hawkins, 1997; Pflaum & Rivers, 1991; Health care fraud, 2002) suggest that service providers account for the greatest proportion of fraud and abuse. The perpetrators of some types of fraud schemes (e.g. surgeries, invasive testing, and certain drug therapies) even deliberately and callously place their trusting patients at significant physical risk. Therefore, in this research, we focus on the detection of fraudulent and abusive behavior by service providers.

Currently, detecting such fraud and abuse relies heavily on medical knowledge. The carriers of nearly every insurance program around the world employ experts, who are pre-eminent in their specialty, to detect suspicious claims in their programs (He et al., 1997). These experts review medical claims, and verify the necessity of services according to the conditions of the patients. It is clear that this task is both effort and time-consuming, especially in the case of large-scale insurance programs.

The huge human effort raises the need of using information techniques to detect suspicious cases (Frieden, 1992). EFD, an expert system developed by Travel-Insurance Companies (Major & Riedinger, 1995), utilizes micro-knowledge (behavioral heuristics) coupled with information theory to select rules for performing the task of identifying outliers (fraud). Herb and Tom (Herb & Tom, 1995) derived fraud indicators and rules from the knowledge and experience of human experts to develop a computer-based expert to facilitate the work of insurance carriers.

Another line of research focuses on the use of more recent machine learning technologies. In this approach, features are often identified by expert consultants and used in the subsequent development of induction schemes. For example, the research by Sokol et al. (Sokol, Garcia, Rodriguez, West, & Johnson, 2001; Sokol, Gaarcia, West, Rodriguez, & Johnson, 2001), funded by the Health Care Financing Administration and the Office of the Inspector General in US, built a model that aimed to discriminate between normal and suspicious claims. For each care service, such as chiropractic services, laboratory/radiology procedures, and preventive medical services, a set of features is identified and an inductive model is accordingly developed to detect suspicious claims in a particular care service.

The work reported in Hall (1996); He et al. (1997)), funded by the Health Insurance Commission of Australia, aims to detect service providers who are practicing inappropriately. In this work, after discriminating features are determined (typically 25–30 features identified by specialists), a fuzzy–logic, neural-network-based induction algorithm is used to generate the detection model. This model is then used to tag suspicious service providers. Similarly, the work reported by Blue Cross and Blue Shield organizations (Cox, 1995) uses a fuzzy-based system to manage the claim profiles of service providers.

While the above-mentioned approaches reduce the workload of human experts, the enormous knowledge engineering task of identifying either discriminating rules or discriminating features is still required. Moreover, due to the manual and ad hoc nature of the development process, the resultant prototypes have limited extensibility and adaptability.

3. Research framework

In this section we introduce the concept of clinical pathways and our proposed process-mining framework that distinguishes fraudulent and abusive cases from normal ones.

The concept of *clinical pathways* (or *integrated care pathways*) was initiated in the early 1990s, and defined as 'multidisciplinary care plans, in which diagnosis and therapeutic intervention are performed by physicians, nurses, and other staff for a particular diagnosis or procedure' (Healy et al., 1998; Ireson, 1997). Clinical pathways are typically driven by physician orders and clinical industry and local standards of care. Once the pathways are created, they are viewed as algorithms of the decisions to be made and the care to be provided to a given patient or patient group. For example, the pathway of cholecystectomy (Ireson, 1997) begins with the preadmission process, which mainly involves preadmission testing and anesthesia consultation, goes though several assessments, surgery, and physician orders, and ends with a follow-up visit at the surgeon's office.

The application of clinical pathways is an efficient approach to analyzing and controlling clinical care processes. It aims to have medical staff performing the care services in the *right order*, enabling best practice—without rework and resource waste—to be implemented. Consider the cholecystectomy pathway. Care activities are sequenced on a timeline so that physicians can make suitable orders in accordance with the test results in the preadmission step, and anesthetic can be executed during the performance of surgery on the basis of anesthesia consult. In today's competitive healthcare environment, the competition advantage of a healthcare institution relies not only on outstanding professional quality but also on the agility of clinical care processes, and so the concept of clinical

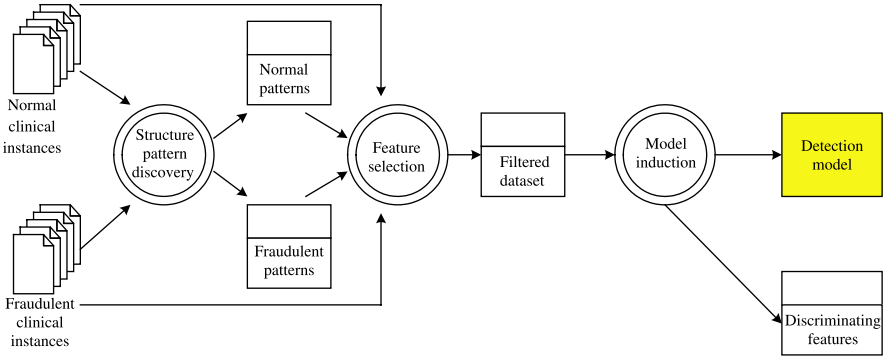


Fig. 1. The process-mining framework.

pathways is receiving considerable attention from managers in large hospitals around the world (Ireson, 1997).

This concept of clinical pathways shows great promise in detecting fraud and abuse by service providers. A care activity is very likely to be fraudulent if it orders suspiciously. For example, since physicians prefer performing simple, non-invasive tests before performing more complex, invasive tests, there is a high probability that the same set of care activities performed in a different order is fraudulent or abusive. Extensively, to accurately determine the appropriateness of a care activity performed on a particular patient, we must take into account the other activities performed on the patient. For example, while a single ambulant visit is normal, repetitive visits are problematic, especially where the average length of pathway instances is small. This observation initiates our idea that the clinical structures, including care activities and their order of execution, can be used to discriminate between normal and fraudulent cases.

A schematic of our process-mining framework is provided in Fig. 1. Generally, two sets of clinical instances, which are labeled as normal and fraudulent, serve as the input of the module for discovering structure patterns. This module produces a set of structure patterns that have occurred frequently, which then serve as features of clinical instances. Each clinical instance is considered an example that comprises a set of features and a class label (normal or fraudulent). A feature-selection module to eliminate redundant and irrelevant features further filters the resultant data set. The selected features and the data set are finally used to construct the detection model, which is performed by the induction module. The detection model is then used to detect the incoming instances that are fraudulent.

4. Structure pattern discovery

The first step of the proposed framework involves extracting patterns in a way amenable to represent structures of clinical instances. In this section, we explore the entrance problem: the discovery of structure patterns.

Typically, a clinical instance is a process instance comprising a set of activities, each of which is a logical unit of work performed by medical staffs. For example, a patient treatment flow may involve measuring blood pressure, examining respiration, and medicine treatment. These activities, each appearing over a temporally extended interval, may execute sequentially, concurrently, or repeatedly. For example, before giving any therapeutic intervention, diagnosis activities are usually executed to verify the condition of a patient. Also, more than one therapeutic intervention may be executed concurrently in order to increase the curative effect in some cases. As a result, if we want to extract structure patterns from clinical instances, we need to take structural characteristics of process—temporally extended intervals and various transitions—into consideration.

In this research, we apply structure pattern mining techniques proposed in our previous work (Hwang, Wei, & Yang, 2004; Wei, Hwang, & Yang, 2000) to identify a set of structure patterns from clinical instances. To make the paper self-contained, related definitions and algorithms in the context of clinical instances are described as below.

Definition 1. A clinical instance I is a set of triplets (V_i, st, et), where V_i uniquely identifies an activity, and st and et are timestamps representing the starting time and ending time of the execution of V_i in I, respectively.

Given a clinical instance, the temporal relationship between any activity pair can be classified into two types, *followed* and *overlapped*, as follows:

Definition 2. In a clinical instance I, an activity V_i is said to be *followed* by another activity V_j if V_j starts after V_i terminates in I.

Definition 3. In a clinical instance I, two activities, V_i and V_j, are said to be *overlapped* if V_i and V_j incur overlapped execution durations in I.

Definition 4. An activity V_i is said to be *directly followed* by another activity V_j in a clinical instance I if V_i is followed by V_j in I and there does not exist a distinct activity V_k in I such that V_i is followed by V_k and V_k is followed by V_j in I.

To represent the temporal relationships between activities in a clinical instance concisely, a *temporal graph* is defined as follows.

Definition 5. The pertinent *temporal graph* of a clinical instance I is a directed acyclic graph $G=(V, E)$, where V is the set of activities in I, and E is a set of edges. Each edge in G is an

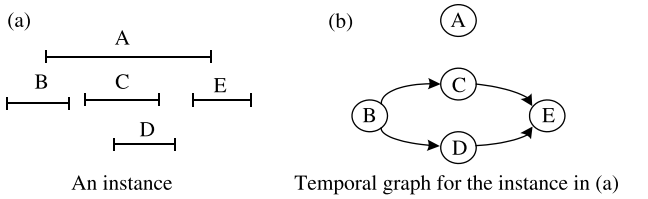


Fig. 2. Example of a clinical instance and the corresponding temporal graph.

ordered pair (V_i, V_j), where $V_i, V_j \in V$, $V_i \neq V_j$, and V_i is directly followed by V_j.

Fig. 2a and b shows a clinical instance and its temporal graph, respectively.

A structure pattern with a certain user-specified minimum support can also be represented as a temporal graph.

Definition 6. A temporal graph G is said to be *supported* by a clinical instance I if all followed and overlapped relationships that exist in G are present in I.

Definition 7. A temporal graph G is said to be *frequent* if it is supported by no less than $s\%$ of the clinical instances, where $s\%$ is a user-defined minimum support threshold.

Definition 8. A temporal graph $G=(V, E)$ is a *temporal subgraph* of another temporal graph $G'=(V', E')$ if $V \subseteq V'$ and for any pair of vertices $v_1, v_2 \in V$, there is a path in G connecting v_1 to v_2 if and only if there is a path in G' connecting v_1 to v_2. If G is a temporal subgraph of G', then G' is a *temporal supergraph* of G.

Therefore, given a set of temporal graphs, each of which represents a clinical instance, the discovery of structure patterns requires finding all frequent temporal graphs. Each such temporal graph is referred to as a structure pattern.

As with association rule (Agrawal & Srikant, 1994) and sequential pattern (Agrawal & Srikant, 1995) algorithms, we exploit the downward closure property of the support measure to improve the efficiency of searching for frequent temporal graphs. The downward closure property suggests that if a temporal graph G has support of at least $s\%$, any temporal subgraph of G must have support of at least $s\%$ or, conversely, if a temporal graph G has support of less than $s\%$, any temporal supergraph of G must have support of less than $s\%$. Accordingly, we adopted an iterative procedure similar to that in the Apriori (Agrawal & Srikant, 1994) and AprioriAll (Agrawal & Srikant, 1995) algorithms. Specifically, potentially frequent temporal graphs (called *candidate temporal graphs*) of size k can be constructed from joining frequent temporal graphs of size $k-1$. The clinical instances are then scanned to identify frequent temporal graphs of size k from the set of candidate temporal graphs of the same size. This procedure is executed iteratively until no further frequent temporal graphs are found. Let C_k and L_k denote the set of candidate temporal graphs and the set of frequent temporal graphs of size k, respectively. The structure pattern discovery algorithm is sketched as follows:

MiningStructurePatterns(S: a set of clinical instances): a set of structure graphs
{
  Scan S to find the set $TGraphSet_1$ of all activities with minimum support;
  $n=1$;
  Repeat {
    $n=n+1$;
    $\text{CandidateSet}_n =$
    $\text{GenerateCandidateGraph}(\text{TGraphSet}_{n-1})$;
    Scan S to find a subset $TGraphSet_n$ of $CandidateSet_n$ with minimum support;
  } Until $TGraphSet_n = \varnothing$;
  Return $TGraphSet_1 \cup TGraphSet_2 \cup \cdots \cup TGraphSet_{n-1}$;
}

Intuitively, two frequent temporal graphs of size $k-1$ can be joined if they differ only in one activity and contain the same temporal relationships for any pair of common activities. However, this simple-minded joining process will result in many redundant candidate temporal graphs. Consider the following example. Suppose the set of frequent temporal graphs in iteration 2 is $\{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$. Any pair in this set can be joined to form the candidate temporal graph of $A \rightarrow B \rightarrow C$. That is, three identical candidate temporal graphs of size 3 will be generated. In the following, a joining algorithm is proposed to eliminate (or at least control) such redundancy.

Definition 9. Let G be a temporal graph and v be a vertex in G. The operation of subtracting v from G, denoted as $G-\{v\}$, deletes v and its associated edges from G. In addition, transitive edges via v are reconstructed by connecting each source vertex of incoming edges of v to each destination vertex of outgoing edges of v.

This subtraction operation can be illustrated as follows. Fig. 3b–f shows all of the temporal subgraphs resulting from subtracting a vertex from the temporal graph G shown in Fig. 3a. When the vertex B is subtracted from G, edges $A \rightarrow C$

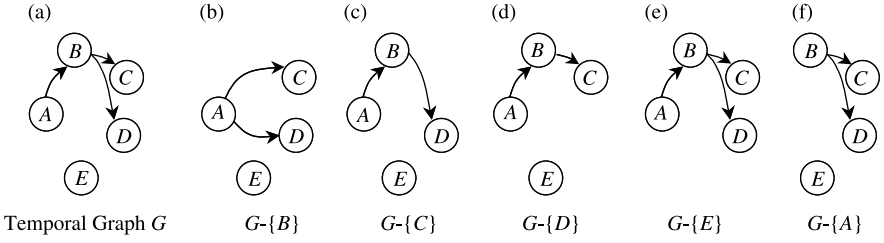


Fig. 3. Examples of the subtraction operation.

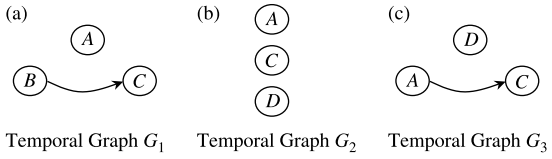


Fig. 4. Three example temporal graphs of size 3.

and $A \rightarrow D$ are reconstructed as shown in Fig. 3b. As shown in Fig. 3c, the deletion of the vertex C from Fig. 3a does not introduce any new edge in G since C does not have any outgoing edge. Fig. 3d–f illustrates the remaining temporal subgraphs derived from Fig. 3a by deleting D, E, and A, respectively.

Observation 1. Let s be a vertex with no incoming edges (called a source vertex) and e be a different vertex with no outgoing edges (called a sink vertex) in a temporal graph G. If G is frequent, both $G-\{s\}$ and $G-\{e\}$ must be frequent.

Based on this observation, to determine whether two frequent temporal graphs can be joined, only their source vertices and sink vertices need to be considered. Accordingly, we formally define joinable temporal graphs as follows:

Definition 10. Two temporal graphs G_i and G_j are said to be *joinable* if there exists a source vertex s in G_i and a sink vertex e in G_j such that $G_i-\{s\}=G_j-\{e\}$.

Consider the temporal graphs shown in Fig. 4. Designating vertex B as a source activity of G_1 (shown in Fig. 4a) and vertex D as a sink activity of G_2 (shown in Fig. 4b), these two temporal graphs are *joinable* since $G_1-\{B\}=G_2-\{D\}$. The temporal graphs G_1 and G_3 or G_2 and G_3, however, are not joinable.

Given two joinable temporal graphs G_i (with s being a source vertex) and G_j (with e being a sink vertex), the temporal relationship between any pair of activities (except that between s and e) present in G_i or G_j will be preserved in a resulting candidate temporal graph. Since more than one permissible temporal relationship between s and e may exist, the joining of two joinable temporal graphs of size $k-1$ can lead to multiple candidate temporal graphs of size k. The temporal relationship between s and e in a candidate temporal graph can be either (1) no edge exists between s and e or (2) an edge connects s to e. Note that the case where an edge connects e to s needs not be considered, as it results in a temporal graph with s and e not being source and sink vertices, respectively. From Observation 1, it is clear that if a temporal graph G with a source vertex s and a sink vertex e is frequent, both frequent temporal graphs $G-\{s\}$ (where s is a source vertex in G) and $G-\{e\}$ (where e is a sink vertex in G) must be joinable. Formally, the join set of two joinable temporal graphs G_i (with s being the source vertex) and G_j (with e being the sink vertex) is composed of

1 $G_i \cup G_j$[1], and
2 $G_i \cup G_j \cup \{s \circledR e\}$ if there does not exist a path from s to e in $G_i \cup G_j$.

[1] The union of two graphs $G_i=(V_i, E_i)$ and $G_j=(V_j, E_j)$ results in a new graph $G=(V_i \cup V_j, E_i \cup E_j)$.

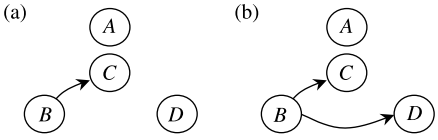


Fig. 5. Two candidate temporal graphs resulting from joining G1 and G2 in Fig. 4.

Consider the two joinable temporal graphs G_1 and G_2 shown in Fig. 4. The join set of G_1 (with B being a source vertex) and G_2 (with D being a sink vertex) includes two candidate temporal graphs of size 4, as shown in Fig. 5.

The described downward closure property can further be exploited to reduce the set of resulting candidate temporal graphs. A candidate temporal graph G of size k will not be frequent if any of its temporal subgraphs of size $k-1$ is not in L_{k-1} and, hence, should be eliminated from C_k. Such a pruning process requires, for each candidate temporal graph of size k, the derivation (using the subtraction operation defined in Definition 9) of all of its temporal subgraphs of size $k-1$. The pseudo codes of *GenerateCandidateGraph*() for generating a set of candidate temporal graphs of size k from a set of frequent temporal graphs of size $k-1$ and that of *DeriveSubgraph*() for deriving all temporal subgraphs of size $|G|-1$ for a temporal graph G are accordingly listed below.

```
GenerateCandidateGraph(TGS: a set of frequent temporal
graphs): a set of temporal graphs
{
  CandidateSet = ∅;
For (each pair of graphs (Gi, Gj) in TGS) {
  For (each source vertex s in Gi) {
    For (each sink vertex e in Gj) {
    If (Gi−{s} = Gj−{e}) { // joinable
      UG1 = Gi ∪ Gj; UG2 = Gi ∪ Gj ∪ {s → e};
      CandidateSet = CandidateSet È {UG1};
      If (there exists no path from s to e in UG1) {
      CandidateSet = CandidateSet ∪ {UG2};
      }
    }
    }
  }
  }
  For (each graph G in CandidateSet) {
  If (DeriveSubgraph(G) ∩ TGS ≠ DeriveSubgraph(G)) {
    CandidateSet=CandidateSet − {G};
  }
}
    Return CandidateSet;
  }
  DeriveSubgraph(G: a temporal graph): a set of temporal
graphs
{
 Subgraph = ∅;
For (each vertex v in G) {
Source=the set of vertices incident to v;
Sink=the set of vertices incident from v;
```

$SG = G - \{v\}$;
For (each vertex pair (v_s, v_d) where $v_s \in$ *Source* and $v_d \in$ *Sink*) {
If there does not exist a path between v_s and v_d in *SG* then $SG = SG \cup \{v_s \rightarrow v_d\}$;
}
Subgraph = Subgraph $\cup$ {SG};
}
Return *Subgraph*;
}

For brevity, detailed considerations (including the data structures) for improving the efficiency and scalability of the structure pattern mining techniques are not elaborated here. Besides, there exist other approaches for discovering structure patterns, though the experimental results showed that the above-mentioned algorithm achieved the best performance in most operating regions (Hwang et al., 2004). Hence, we consider only this algorithm in our subsequent discussions (further details are available elsewhere (Hwang et al., 2004; Wei et al., 2000)).

5. Pattern feature selection

In our framework, frequent structure patterns discovered by the algorithm described in Section 4 are regarded as features. In practice, the number of features is often huge (usually more than 10,000). It is widely recognized that the number of features has a strong impact on the efficiency of an induction algorithm, and the inclusion of irrelevant or redundant features may degrade its accuracy. Therefore, it is imperative to reduce the feature set prior to constructing a detection model to decrease the running time of the induction algorithm and to increase the accuracy of the resultant model. This feature selection issue is addressed in this section.

Several studies have addressed the problem of feature selection. As noted in John, Kohavi, and Pfleger (1994), the proposed approaches fall into the following two categories: the wrapper model and the filter model. The wrapper model (Blum & Langley, 1997; Caruana & Freitag, 1994; John et al., 1994; Langley & Sage, 1994) scans through the space of feature subsets in search of the one that has the highest estimated accuracy from an induction algorithm. Specifically, the feature selection algorithm continuously interacts with the underlying induction algorithm, with the aim of choosing a subset of features that achieves the best classification result for the induction algorithm. While these methods have been shown to achieve some success on induction, they suffer from high computation cost and are not applicable to tasks with even a few hundred features.

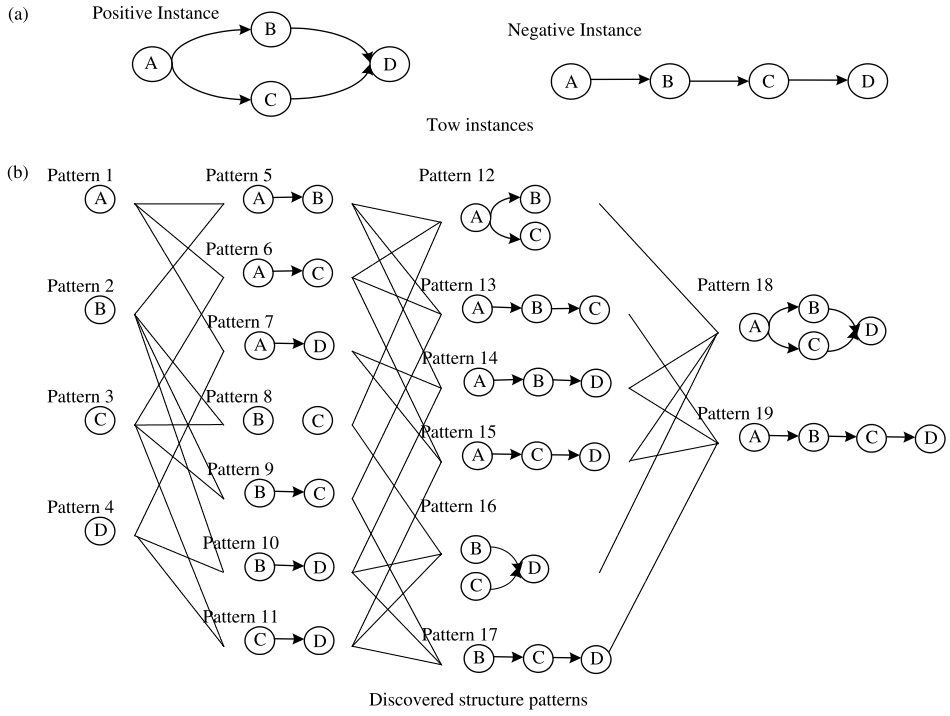
The filter model introduces a preprocessing step prior to induction. As such, the adoption of the induction algorithm does not interfere with the selection of the feature selection algorithm. A major benefit with the filter model is that it does not need to search through the space of feature subsets as required in the wrapper models, and is therefore efficient for domains containing a large number of features. Three of the most well-known filter methods are RELIEF (Kira & Rendell, 1992), FOCUS (Almuallim & Dietterich, 1994), and the Markov blanket filter (Koller & Sahami, 1996). In RELIEF, each feature is individually assigned a weight indicating its relevance to the class label, and a subset of features with the highest weights is selected. It is possible that RELIEF fails to remove redundant features, since two predictive (but highly correlated) features will both be selected. The FOCUS algorithm exhaustively searches all feature subsets in order to identify a minimal set of features that consistently label instances in the training data. This consistency criterion makes FOCUS vulnerable to noise in the training data. Moreover, searching the power set of features also makes this algorithm impractical for domains with a large number of features. Koller and Sahami developed a probability framework (the Markov blanket filter) for selecting an optimal subset of features (Koller & Sahami, 1996). Theoretically, this method eliminates a feature if it gives no additional information beyond that subsumed by a subset of remaining features (called the Markov blanket). Since finding the Markov blanket of a feature might be computational infeasible, this research resulted in an algorithm that computes an approximation to the optimal feature set.

Since the structure pattern discovery algorithm may generate a large number of structure patterns (or features), we focus our attention on the filter model due to its key advantage on computation cost. In our framework, the discovered structure patterns are regarded as features, each of which denotes whether a specific pattern is supported by an instance. Thus, each instance can be translated as a set of feature values with a class label, and our view on a translated example can be formally described as below.

Definition 11. A translated example e of an instance I is a pair ($\boldsymbol{f}$, c), where $\boldsymbol{f}$ and c are a feature vector and a class label, respectively. $\boldsymbol{f} = (f_1, f_2, \ldots, f_n)$ denotes an assignment of a set of Boolean features $\boldsymbol{F} = (F_1, F_2, \ldots, F_n)$, in which feature f_j is set to 1 if and only if instance I supports the corresponding pattern of F_j. c is an assignment of a categorical variable C.

Consider an example shown in Fig. 6. Suppose there are two clinical instances (Fig. 6a) that are considered as positive and negative classes. A 50% support threshold results in the discovery of 19 structure patterns, as shown in Fig. 6b, where edges represent the subgraph relationship. Therefore, the two translated examples shown in Fig. 6c are generated, each having 19 corresponding feature values and one class label. In the first translated example, 1 is assigned to Features 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 14, 15, 16, and 18 since the first instance supports 15 corresponding patterns, and 0 is assigned to the other features. Similarly in the second instance, Features 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, and 17 are assigned 1, and the other features are assigned 0. Moreover, the first example is labeled 0, while the second one is labeled 1.

By Definition 11, the pattern classifier takes as input a translated example e and predicts the class to which the associated instance belongs. The classifier must make its



(c)

	Feature																			Class
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
Translated example	1	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	0	0
	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	1

Two translated examples

Fig. 6. Instances and translated examples.

decision based on the assignment $\boldsymbol{f}$ associated with example e. Therefore, we use a conditional probability distribution to model the classification problem. That is, for each assignment of value $\boldsymbol{f}$ to $\boldsymbol{F}$, we define a conditional distribution $\Pr(C|\boldsymbol{F}=\boldsymbol{f})$ on the different classes, C. Given a set of translated examples, each of which is represented as a set of feature values $\boldsymbol{f}$ and a class label c, the feature selection problem involves finding a small subset $\boldsymbol{F}'$ of $\boldsymbol{F}$ such that $\Pr(C|\boldsymbol{F}=\boldsymbol{f})$ and $\Pr(C|\boldsymbol{F}'=\boldsymbol{f}')$, where $\boldsymbol{f}'$ is a projection of $\boldsymbol{f}$ on $\boldsymbol{F}'$, are as close as possible.

We utilize probabilistic reasoning to reduce the feature space so as to minimize the amount of lost information. Intuitively, features that cause a small difference on the two distributions (with and without these features) are those that provide less additional information beyond what we would obtain from the other features. This intuition can be captured via the notion of *conditional independence*, originally proposed by Pearl (Pearl, 1988).

Definition 12. (Pearl, 1988). A set of variables $\boldsymbol{Y}$ is said to be *conditionally independent* of another set of variables $\boldsymbol{X}$ given some set of variables $\boldsymbol{Z}$ if, for any assignment values of $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ to the variables $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$, respectively, $\Pr(\boldsymbol{X}=\boldsymbol{x}|\boldsymbol{Y}=\boldsymbol{y}, \boldsymbol{Z}=\boldsymbol{z})=\Pr(\boldsymbol{X}=\boldsymbol{x}|\boldsymbol{Z}=\boldsymbol{z})$. That is, $\boldsymbol{Y}$ provides no information about $\boldsymbol{X}$ beyond what is already in $\boldsymbol{Z}$.

Thus, if a feature F_i is conditionally independent of the class variable C, given $F-F_i$ we can eliminate F_i without increasing the distance from the desired distribution. While it is impractical to test for conditional independence, this idea sheds light on a solution. Specifically, we exploit sub/superrelationships among structure patterns to efficiently eliminate features.

Definition 13. Let F_i correspond to a k-sized pattern i. A feature F_j, corresponding to pattern j, is said to be a *descendant* of F_i if j is a subpattern of i. A descendant of F_i is also a *child* of F_i if it is of size $k-1$. The set of children of F_i is denoted as Child(F_i), and the set of descendants of F_i is denoted as Descendant(F_i).

Definition 14. A feature F_j is said to be a *parent* of F_i if F_i is a child of F_j. A feature F_j is said to be an ancestor of F_i if F_i is a descendant of F_j. We use Parent(F_i) and Ancestor(F_i) to denote the set of parents and ancestors, respectively, of F_i.

Take Fig. 6 as an example again. Features 1, 2, 3, 4, 5, 6, and 8 are obviously the descendants of Feature 12, since their corresponding patterns are subpatterns of the pattern of Feature 12 (see Fig. 6b). Among them, Features 5, 6, and 8 are the children of Feature 12. Similarly, Feature 18 is a parent of Feature 12 since the corresponding pattern of Feature 18 is a superpattern of that of Feature 12, and their sizes differ by 1.

Let X be a set of instances, each supporting some subpattern of I, and Y be a set of instances that support i.

Obviously, $X \supseteq Y$ due to the downward closure property. Therefore, if X falls into the category of a particular class, Y must belong to the same class. From the classification point of view, F_i thus provides no further information than that provided by a subset of the descendants of F_i. This observation is formalized by Lemma 1.

Lemma 1. Let $\boldsymbol{A} \subseteq$ Descendant(F_i) be a set of features and $\boldsymbol{E}$ be the set of translated examples that have a value of 1 in every feature of $\boldsymbol{A}$. If every translated example in $\boldsymbol{E}$ has the same class label $c_l \in C$, then F_i is conditionally independent of C given $\boldsymbol{A}$.

Clearly, if we can find a set of features $\boldsymbol{A} \subseteq$ Descendant(F_i) such that the condition stated in Lemma 1 is satisfied, then for any feature F_j that is an ancestor of F_i, F_j must be conditionally independent of C given $\boldsymbol{A}$, since $\boldsymbol{A}$ is also a subset of Descendants(F_j). This results in the following corollary, which is derived directly from Lemma 1.

Corollary 1. Let $F_j \in$ Ancestor(F_i) and $\boldsymbol{A} \subseteq$ Descendant(F_i) be a set of features. Let $\boldsymbol{E}$ be the set of translated examples that have a value of 1 in every feature of $\boldsymbol{A}$. If every translated example in $\boldsymbol{E}$ has the same class label $c_l \in C$, then F_j is conditionally independent of C given $\boldsymbol{A}$.

Lemma 1 and Corollary 1 allow a conditionally independent feature F_i and all its ancestors to be eliminated if there exists a set of features $\boldsymbol{A} \subseteq$ Descendant(F_i) that satisfies the condition stated in Lemma 1. In this case, F_i and all its ancestors are considered to be subsumed by $\boldsymbol{A}$ as they provide no further information in terms of classification. However, enumerating all feature subsets and conducting such a test is still impractical since the number of feature subsets is exponential with the total number of features. To remedy this problem, we derive Theorem 1 that further reduces the search space for such a feature set $\boldsymbol{A}$.

Theorem 1. Let $\boldsymbol{B} =$ Child(F_i) and $\boldsymbol{A} \subseteq$ Descendant(F_i). Further, let E_A be the set of translated examples that have a value of 1 in every feature of $\boldsymbol{A}$, and E_B be the set of translated examples that have a value of 1 in every feature of $\boldsymbol{B}$. If every translated example in E_A has the same class label $c_l \in C$, then every translated example in E_B must have the same class label c_l. In other words, if F_i is conditionally independent of C given $\boldsymbol{A}$, then F_i must be conditionally independent of C given $\boldsymbol{B}$.

As a result, for a given feature F_i, we can simply verify its children. If every translated example that has an assignment of **1** in every child of F_i has the same class, F_i and all ancestors of F_i can be eliminated. Since smaller patterns have more ancestors, verifying features in the order from small to large provides the potential to eliminate features in earlier stages. Therefore, features are listed in ascending order of size, and our algorithm sequentially verifies whether a feature and the class variable are conditionally independent. The pseudo-code of the algorithm *AncestorPruning*() is listed below:

AncestorPruning($\boldsymbol{T}$: a training data set; $\boldsymbol{F}$: a set of features):
$\boldsymbol{G}$: a subset of $\boldsymbol{F}$
// Suppose features in $\boldsymbol{F}$ are listed in ascending size order
{
G ← F;
For (each feature F_i in G) {
class ← ϕ;
For (each translated example e in $\boldsymbol{T}$) {
If ($e.Child(F_i) = \mathbf{1}$) { // $\boldsymbol{e}$ has **1** in every child feature of F_i
class ← class ∪ e.class;
}
} // end of example loop
If (|*class*| = 1 // belong to only one class) {
$\boldsymbol{G} = \boldsymbol{G}$ - {F_i} - Ancestor(F_i);
}
}// end of feature loop
Return $\boldsymbol{G}$;
}// end of AncestorPruning

Let us illustrate *AncestorPruning*() using the example shown in Fig. 6. Features 1–4 pass our test since they do not have any children. Features 5–11 also pass the test because the corresponding patterns of their children are supported by both instances, each having different classes. Feature 12 is the first feature that does not pass the test, because only one instance supports all corresponding patterns of its children—Features 5, 6, and 8. Therefore, Feature 12 and its ancestor Feature 18 are eliminated. For the same reason, Feature 13, 16, 17, and 19 are eliminated in subsequent steps.

We expect *AncestorPruning*() to eliminate a large number of structure patterns in practice. To further reduce the number of patterns before applying the induction algorithm, we propose applying a general feature-selection algorithm. Among previous filter methods, we choose the Markov blanket filter on the basis of both efficiency and redundancy elimination. It is a direct application of the approach described by Koller and Sahami (Koller & Sahami, 1996). We briefly describe this application in the context of structure pattern elimination to give the reader an overall view of the entire approach.

Definition 15 (Pearl, 1988). Let $F_i \in \boldsymbol{F}$ be a feature and $\boldsymbol{M} \subseteq \boldsymbol{F}$ be a set of features that does not contain F_i. $\boldsymbol{M}$ is a Markov blanket for F_i if {F_i} is conditionally independent of $(\boldsymbol{F} \cup C) - \boldsymbol{M} - \{F_i\}$ given $\boldsymbol{M}$.

Obviously, if $\boldsymbol{M}$ is a Markov blanket of F_i, then F_i is conditionally independent of the class C given $\boldsymbol{M}$. Therefore, if a Markov blanket of F_i can be found, the filter method can safely remove F_i from $\boldsymbol{F}$. Also, Koller and Sahami further proved that a feature tagged as unnecessary based on the existence of a Markov blanket remains unnecessary in later phases when more features are eliminated (Koller & Sahami, 1996). They accordingly adopted a greedy strategy that eliminates unnecessary features one by one based on the existence of a Markov blanket.

In most cases, however, few if any features will have a Markov blanket of limited size. Therefore, it is necessary to construct an approximate Markov blanket that is close to the real one. To reduce computational overhead, only feature sets of a specific size K are examined. The intuition behind the

construction of an approximate Markov blanket is that, if a feature F_i does have a Markov blanket, F_i will directly influence the features of its Markov blanket. Therefore, an approximation to the Markov blanket—some set of K features that are strongly correlated with F_i—can be constructed heuristically.

In order to determine how close an approximation is to the real Markov blanket, Koller and Sahami further defined the expected cross-entropy between a feature F_i and an approximated Markov blanket M_i, as shown in Eq. (1). A lower value of $\Delta(F_i|M_i)$ indicates a closer approximation. Thus, the feature F_i that has the lowest $\Delta(F_i|M_i)$ value is most likely to have a Markov blanket in the remaining features, and thus should be eliminated first. This procedure is performed iteratively until only the desired number of features remain. This algorithm is called *MarkovBlanketFilter*(G, N), which takes the original set of features G and the desired number of features N as input and produces a reduced set of features. A detailed description of the algorithm is available elsewhere Koller and Sahami (1996).

$$\Delta(F_{\mathrm{i}}|M_{\mathrm{i}}) = \sum_{f_{\mathrm{m_i}}, f_{\mathrm{i}}} \Pr(M_{\mathrm{i}} = f_{\mathrm{m_i}}, F_{\mathrm{i}} = f_{\mathrm{i}}) \times D\big(\Pr\big(C|M_{\mathrm{i}} = f_{\mathrm{m_i}}, F_{\mathrm{i}} = f_{\mathrm{i}}\big), \Pr\big(C|M_{\mathrm{i}} = f_{\mathrm{m_i}}\big)\big) \tag{1}$$

where

$$D(\mu, \sigma) = -\sum_{x \in \text{propability space}} \mu(x) \log \frac{\mu(x)}{\sigma(x)}$$

denotes the cross-entropy of distribution μ to σ.

The complete algorithm *FeatureSelection*( ) that combines the two stages mentioned above is listed below.

FeatureSelection(***T***: a training set; ***F***: a set of features; N: an integer): ***G***: a set of features

// Suppose features in ***F*** are listed in ascending order of their sizes

```
{
//First Stage
  G = AncestorPruning(T, F);
//Second Stage
  If (|G| > N) {
    G = MarkovBlanketFilter(G, N);
  }
  Return G;
}
```

The first stage (*AncestorPruning*()) of our feature-selection algorithm takes O(nmc) time, where n is the total number of features before pruning, m is the number of translated examples, and c is the average number of children of a feature. This is because eliminating a single feature requires the scanning of all translated examples and, for each example, checking all c children features for the value of 1. The second stage, as reported in (Koller & Sahami, 1996), requires $O(p^2(m+\log p))$ operations for first computing the pairwise cross-entropy matrix and sorting it, where p is the number of features that pass from first stage. The subsequent feature selection process requires $O(rpKms2^K)$ time, where r is the number of features after the first stage, K is a small constant that represents the size of candidate Markov blankets (called the conditional set), and s is the number of classes.

It is clear from the above analysis that the running time of the Markov blanket filter algorithm will increase dramatically with larger K. However, a larger K is more likely to subsume the information in the feature, thereby forming a Markov blanket. However, a larger conditioning set, as formed by larger K, may in turn fragment the training set into many small chunks, thereby reducing the accuracy of the probability and hence the cross-entropy estimates. Therefore, there is a trade-off for setting K in terms of classification accuracy.

If a large extent of redundant information can be eliminated at the first stage, there is a high probability that a smaller conditional set will result in a satisfactory approximation. A smaller conditional set reduces the number of chunks and hence increases the accuracy of cross-entropy estimates, and the running time also decreases dramatically since the computation complexity of the second stage is exponential with K. Therefore, the combined approach is particularly suitable for our problem -a domain with a huge number of structure patterns.

6. Evaluation

Our experimental data are the medical claims submitted to the Bureau of National Health Insurance (BNHI) in Taiwan. The BNHI was founded in 1995 to administer the NHI program in Taiwan. Through risk pooling, the BNHI is responsible for providing the public with comprehensive medical care, including health prevention, clinical care, hospitalization, residential care, and social rehabilitation. As of June 2002, more than 21 million individuals were enrolled in the NHI (coverage rate of 96%) and more than 16 thousand medical institutions were contracted in the program, representing about 92% of the medical institutions nationwide (http://www.nhi.gov.tw). The medical care expenditure of NHI has increased dramatically since its inception in 1995 (http://www.nhi.gov.tw). In 1998, the total expenditure of BNHI was NT$ 267 billion (about US$ 8 billion) which, compared to 1995, represents a 34% increase in total healthcare expenditure and a 20% increase in healthcare expenditure per enrollee. With the large number of enrollees and the rapid increase in expenditure, the NHI program is an interesting platform for investigating our model for detecting fraud and abuse. Therefore, we consulted with medical specialists and collected medical data from the NHI for evaluating the effectiveness and efficiency of our detection model.

According to reports of the BNHI (http://www.nhi.gov.tw), medical claims from gynecology departments of various hospitals have exhibited a rapid increase in expenditure as well as a high number of rejected cases relative to the total numberr of claims. For this reason, we decided to focus our attention on medical cases from gynecology departments. By consulting with physicians of gynecology departments, we further chose pelvic inflammatory disease (PID) as our major

target of detection, since PID is the most common disease in gynecology departments and both diagnosis and treatment methods of PID are representatives of gynecology departments.

We collected data from a regional hospital that is a service provider of the NHI program. We initially gathered data relating to 2543 patients from the gynecology department of the hospital between July 2001 and June 2002 and prepared two data sets—normal and fraudulent—using the following steps:

(1) *Filtering out noisy data:* The treatment data of each patient was regarded as an instance, and we removed instances that had missing or noisy attribute values. In this step we removed 77 instances.
(2) *Identifying activities:* Based on the domain knowledge provided by experts, we identified medical activities in the remaining instances. Some activities, such as examination of blood pressure, were performed routinely and thus discarded. We finally identified 127 medical activities in this step.
(3) *Identifying fraudulent instances:* Two gynecologists were involved in the identification of fraudulent instances. They examined all instances, among which 906 instances were judged by both gynecologists as fraudulent.
(4) *Selecting normal instances:* We then randomly selected 906 cases from the remaining instances that both gynecologists considered normal cases. As a result, a total 1812 instances were used in our experiments.

We adopted the Classification Based on Associations algorithm (CBA) (Liu, Hsu, & Ma, 1998) as our induction method. Also, in order to evaluate the detection model, we consider two measures, *sensitivity* and *specificity*, that are often used in medical diagnosis and in the detection of fraudulent behavior (Friedman & Wyatt, 1997; Lavrac, 1999). The *Sensitivity* is the proportion of fraudulent cases that are identified as fraudulent by a system, and *Specificity* is the proportion of the normal cases that are classified as normal by the system. Clearly, a detection system is considered to have good performance if it has both high sensitivity and high specificity.

6.1. Number of features deducted

In order to construct our detection model, patterns are first discovered using the structure pattern discovery algorithm, then translated as features, and finally filtered by the feature subset selection algorithm. Fig. 7 shows the number of features selected in our model. These patterns (features) are discovered at different support thresholds, ranging from 10 to 2% at 2% decrements. Fig. 7a shows the number of initial features (discovered by the structure pattern discovery algorithm) and the number of features that pass the first stage of feature subset selection. Fig. 7b shows the number of features that are eliminated by the first stage of feature subset selection divided by the number of initial features.

As expected, the number of initial features increased as the minimum support decreased. While the number of remaining features still increased moderately as a function of support threshold, a large proportion of features is eliminated by the first stage of feature subset selection. For example, at a support threshold of 2%, an average of 30,701 features is initially discovered while only 3120 features pass the test. Further, as shown in Fig. 7b, the number of eliminated features divided by the number of initial features grows substantially as the minimum support decreases.

6.2. Prediction power with the first stage of feature subset selection

We next investigated the sensitivity and specificity of our detection model, which are constructed by features selected by the first stage of feature selection. At support thresholds of 2–6%, because many (more than 1000) features pass the first stage of feature subset selection, we further filter features by

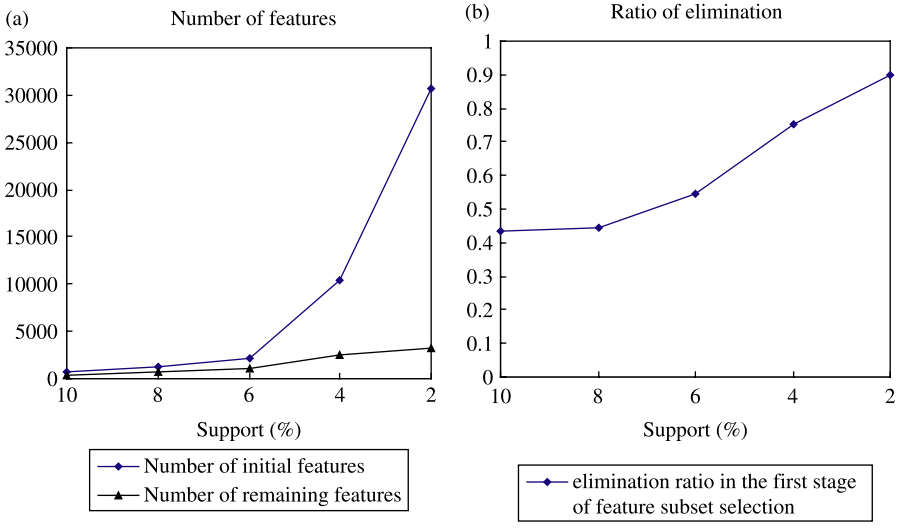


Fig. 7. Effects of feature subset selection.

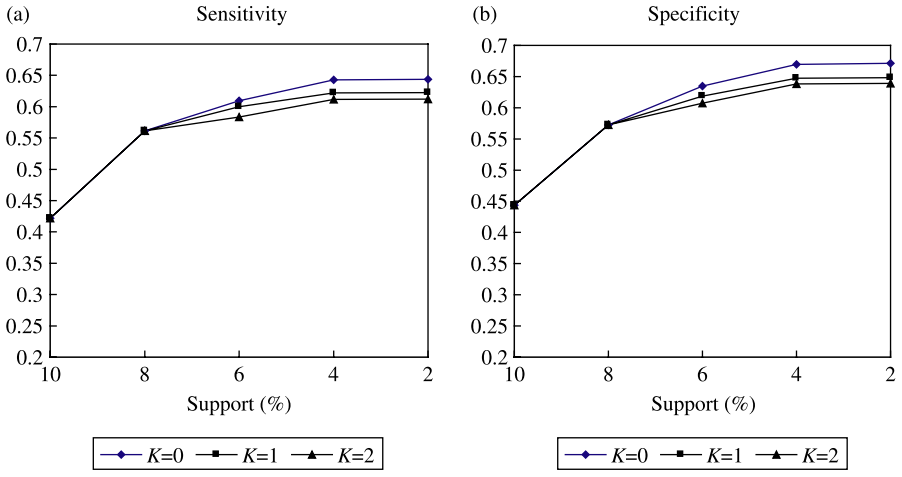


Fig. 8. Sensitivity and specificity of the detection model with the first stage of feature subset selection.

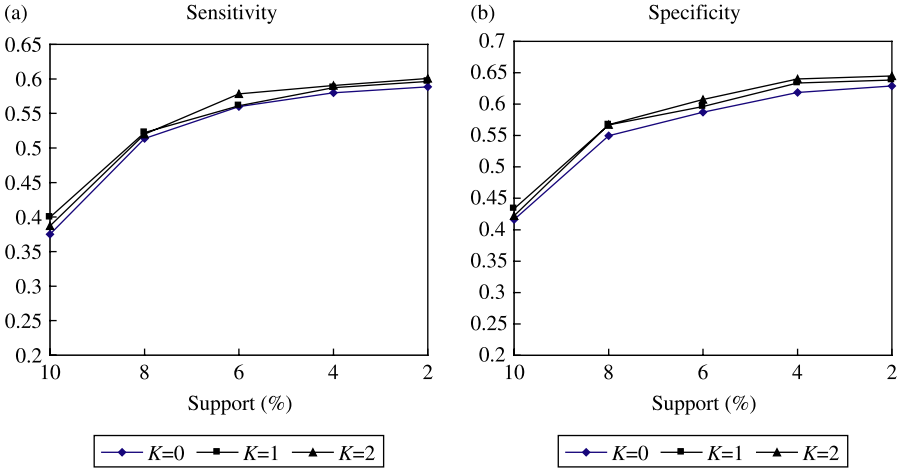


Fig. 9. The sensitivity and specificity of the detection model without the first stage of feature subset selection.

applying the Markov blanket filter (the second stage of feature subset selection) with various blanket sizes ($K=0$, 1, 2). One thousand features ($N=1000$) are finally selected in these cases. Also, since the CBA is most accurate when the minimum support is 1–2% (Liu et al., 1998), we set the support and confidence of the CBA to 1 and 50%, respectively. The resultant sensitivity and specificity of our detection model are depicted in Fig. 8.

Fig. 8 shows that the sensitivity and specificity of the detection model increased as the support threshold decreased. This is as expected, since a lower support threshold indicates the discovery of more features and thus the provision of more information for the classification task. The best sensitivity and specificity (64 and 67%, respectively) are obtained at a support threshold of 2%. It is also worth noting that the best sensitivity and specificity are both obtained at a conditioning level of $K=0$. This demonstrates that a great extent of redundant information has been eliminated in the first stage of feature subset selection, and thus a low conditioning level ($K=0$) is sufficient to further filter out correlated information.

6.3. Prediction power without the first stage of feature subset selection

We also investigated the sensitivity and specificity of our detection model in which all features were selected by a Markov blanket filter with various conditioning settings. The settings of this experiment were the same as the previous one except for the omission of the first stage of feature selection.

Table 1
Discriminating features identified in Chan and Lan (2001)

Feature name
1. Case Type
2. Department Type
3. Patient Type
4. Partial Payment Type
5. Drug Days
6. Physician Gender
7. Drug Fee
8. Diagnosis Fee
9. Examination Fee
10. Drug Administration Fee

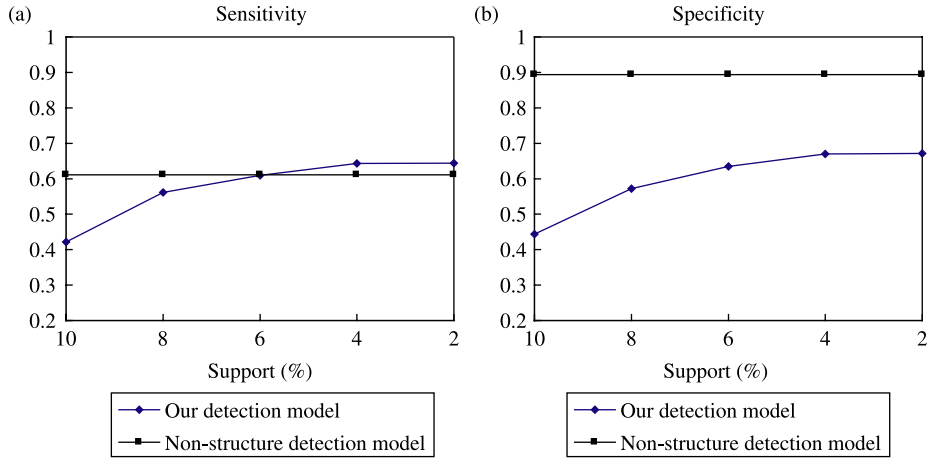


Fig. 10. Comparison of our model and the non-structure detection model.

The sensitivity and specificity of the resultant detection model are depicted in Fig. 9.

It can be seen that the best sensitivity and specificity (60 and 64%, respectively) were both obtained at a conditioning level of $K=2$. Comparison with the results shown in Fig. 8 indicates that the performance of this detection model is slightly worse, which is as expected because the Markov blanket filter uses only approximations to eliminate features. Moreover, the conditioning setting ($K=2$) shows that it is necessary to have a higher conditioning level to filter redundant information, resulting in a longer computation time.

6.4. Comparison of detection models

We finally compare our detection model with that proposed by Chan and Lan (Chan & Lan, 2001), which was designed to detect suspicious claims in the Taiwan NHI program. The features identified in Chan and Lan (2001), as shown in Table 1, were mainly derived from various expense fields of claims by experts' consultants. This model is called non-structure detection model as it makes use of features not related to structures. The resultant sensitivities and specificities of the two detection models are shown in Fig. 10.

Fig. 10 clearly shows that the non-structure detection model, which mainly involves expense features, has high specificity but relatively low sensitivity. This is because normal examples tend to have low expenses, and thus result in a high specificity; whereas fraudulent examples have variable expenses, and thus result in a low sensitivity. Similar conclusions were reported in Chan and Lan (2001). Compared with their detection model, our detection model has more balanced values of sensitivity and specificity. Also, the specificity of their detection model is higher than ours, while the sensitivity of our detection model is slightly higher at low support thresholds.

The comparison of sensitivity in Fig. 10 is not intended to demonstrate that one model is better than the other, but rather to illustrate where the differences lie. Of fraudulent examples returned by the non-structure detection model, our detection model captures 69% of the examples on average. Some examples, such as overdose, are not returned by our detection model. In contrast, of the fraudulent examples returned by our detection model, their detection model captures 63% of the examples on average. Some examples, such as those that have repeated ambulant visits while still have low expense, are not returned by their detection model. This illustrates the differences between our structure driven approach and the non-structure driven approach.

7. Conclusion

In this research, we have outlined a framework that facilitates the automatic and systematic construction of systems that detect healthcare fraud and abuse. We investigated the mining of frequent patterns from clinical instances and the selection of features that have higher discrimination power. The proposed approaches have been evaluated objectively using a real-world data set gathered from the NHI program in Taiwan. The empirical experiments show that our detection model is efficient and capable of identifying some fraudulent and abusive cases that are not detected by a manually constructed detection model.

This work could be extended in several directions. First, the handling of noisy data in this context remains a challenging problem. Second, there are many cost factors in healthcare fraud and abuse detection, and so building detection models that can be easily adjustable according to site-specific cost policies is important in practice. Finally, we believe that it is beneficial and natural to integrate a healthcare fraud and abuse detection system with a cost-restricted system, so that the detection system can communicate with the cost-restricted system when determining the appropriate actions to take.

Appendix A

Proof of Lemma 1. We need to prove $\Pr(C|\boldsymbol{A}, F_i)=\Pr(C|\boldsymbol{A})$. We divide the domain of A into two cases: (1) vectors with at least one 0 value and (2) the vector whose values are all 1.

Case 1. Let T be the set of translated examples. Consider an example $\bar{e} \in T-E$. $\bar{e}$ must be 0 for some feature in $\boldsymbol{A}$. Clearly,

the F_i value of $\bar{e}$ must be 0. Let a_1 be any value of A in $\bar{e}$; we have $\Pr(C|A=a_1)=\Pr(C|A=a_1,F_1=0)$.

Case 2. Consider an example $e\in E$. e must be 1 for every feature in A. Let a_2 be the vector whose values are all 1. Since all translated examples in E belong to class c_1, we have

$$\Pr(C=C_l|A=a_2)=1=\Pr(C=C_l|A=a_2,F_i=0)$$
$$=\Pr(C=c_1|A=a_2,F_i=1),\quad \text{and}$$

$$\forall \bar{C}_l\in dom\,(c),\bar{C}_l\neq C_l.$$

$$\Pr(C=\bar{c}_1|A=a_2)=0=\Pr(C=\bar{c}_1|A=a_2,F_i=0)=\Pr(C$$
$$=\bar{c}_1|A=a_2,F_i=1),$$

From the above two cases, we can easily conclude that $\Pr(C|\boldsymbol{A},F_i)=\Pr(C|\boldsymbol{A})$. In other words, F_i and C are conditionally independent given $\boldsymbol{A}$. □

Proof of Corollary 1. Obviously, $F_i\in$Descendant(F_j). Since $\boldsymbol{A}\subseteq$Descendant(F_i), we can induce that $\boldsymbol{A}\subseteq$Descendant(F_j). From Lemma 1, it is clear that F_j and C are conditionally independent given $\boldsymbol{A}$. □

Proof of Theorem 1. Suppose $\boldsymbol{A}\neq\boldsymbol{B}$ (otherwise this theorem holds). Let $X\in\boldsymbol{A}-\boldsymbol{B}$ and $Y\in\boldsymbol{B}$ be an ancestor of X. After replacing X with Y, we obtain a new set of features $\boldsymbol{B}'\subseteq\boldsymbol{B}$. Let $\boldsymbol{E}_{\boldsymbol{B}}{}'$ be the set of translated examples that have a value of 1 in every feature of $\boldsymbol{B}'$. By the downward closure property, $\mathbf{E}_{\mathbf{A}}\supseteq\mathbf{E}_{\mathbf{B}}{}'$. Therefore, each example in $\mathbf{E}_{\mathbf{B}}{}'$ must have the same class c_l. That is, F_i and C are conditionally independent given $\boldsymbol{B}'$. Since $\boldsymbol{B}'\subseteq\boldsymbol{B}$, F_i and C are conditionally independent given $\boldsymbol{B}$. □

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the international conference on very large data bases, Santiago, Chile, June* (pp. 487–499).

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. *Proceedings of international conference on data engineering, Taipei, Taiwan, March* (pp. 3–14).

Almuallim, H., & Dietterich, T. (1994). Learning Boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, *69*(1/2), 297–305.

Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*(1/2), 245–271.

Caruana, R., & Freitag, D. (1994). Greedy attribute selection. *Proceedings of international conference on machine learning, New Brunswick, NJ* (pp. 28–36).

Chan, C. L., & Lan, C. H. (2001). A data mining technique combining fuzzy sets theory and Bayesian classifier—An application of auditing the health insurance fee. *Proceedings of the international conference on artificial intelligence* (pp. 402–408).

Cox, E. (1995). A fuzzy system for detecting anomalous behaviors in healthcare provider claims. *Intelligent system for finance and business* (pp. 111–134). New York, NY: Wiley.

Frieden, J. (1992). Health care fraud detection enters the information age. *Business and Health* , 29–32 June.

Friedman, C. P., & Wyatt, J. C. (1997). *Evaluation methods in medical informatics*. Berlin: Springer.

Guidelines to health care fraud (1991). *Guidelines to health care fraud*. National Health Care Anti-Fraud Association (NHCAA), NHCAA Board of Governors. http//www.nhcaa.org.

Hall, C. (1996). Intelligent data mining at IBM: New products and applications. *Intelligent Software Strategies*, *7*(5), 1–16.

He, H., Wang, J., Graco, W., & Hawkins, S. (1997). Application of neural networks to detection of medical fraud. *Expert Systems with Applications*, *13*(4), 329–336.

Health care fraud (2002). *Health care fraud: A serious and costly reality for all Americans*. National Health Care Anti-Fraud Association (NHCAA), *REPORT all_about_hcf* http//www.nhcaa.org.

Healy, W. L., Ayers, M. E., Iorio, R., Patch, D. A., Appleby, D., & Pfeifer, B. A. (1998). Impact of a clinical pathway and implant standardization on total hip arthroplasty: A clinical and economic study of short-term patient outcome. *The Journal of Arthroplasty*, *13*(3), 266–276.

Herb, W., & Tom, M. (1995). A scientific approach for detecting fraud. *Best's Review*, *95*(4), 78–81.

Hwang, S. Y., Wei, C. P., & Yang, W. S. (2004). Process mining: Discovery of temporal patterns from process instances. *Computers in Industry*, *53*(3), 345–364.

Ireson, C. L. (1997). Critical pathways: Effectiveness in achieving patient outcomes. *The Journal of Nursing Administration*, *27*(6), 16–23.

John, G., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. *Proceedings of international conference on machine learning* (pp. 121–129).

Kira, K., & Rendell, L. (1992). The feature selection problem: traditional methods and a new algorithm. *Proceedings of the conference on artificial intelligence, San Jose, CA* (pp. 129–134).

Koller, D., & Sahami, M. (1996). Toward optimal feature selection. *Proceedings of international conference on machine learning, Bari, Italy, July* (pp. 282–292).

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. *Proceedings of the AAAI symposium on relevance, Seattle, WA* (pp. 399–406).

Lassey, M., Lassey, W., & Jinks, M. (1997). *Health care systems around the world: Characteristics, issues, reforms*. Englewood Cliffs, NJ: Prentice-Hall.

Lavrac, N., & Lavrac, N. (1999). Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine*, *16*(1), 3–23.

Liu, B., Hsu, W., & Ma, Y. (1998). Integrating classification and association rule mining. *Proceedings of international conference on knowledge discovery and data mining, New York* (pp. 80–86).

Major, J., & Riedinger, D. (1995). *EFD: Heuristic statistics for insurance fraud detection Intelligent system for finance and business* (pp. 145–164). New York, NY: Wiley.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. Los Altos, CA: Morgan Kaufmann.

Pflaum, B. B., & Rivers, J. S. (1991). Employer strategies to combat health care plan fraud. *Benefits Quarterly*, *7*(1), 6–14.

Sokol, L., Garcia, B., Rodriguez, J., West, M., & Johnson, K. (2001). Using data mining to find fraud in HCFA health care claims. *Top Health Information Management*, *22*(1), 1–13.

Sokol, L., Garcia, B., West, M., Rodriguez, J., & Johnson, K. (2001). Precursory steps to mining HCFA health care claims. *Proceedings of the Hawaii international conference on system sciences, Hawaii* (pp. 6019).

United States General Accounting Office (1992). *Health insurance: Vulnerable payers lose billions to fraud and abuse*. GAO-HRD-92-69.

Wei, C. P., Hwang, S. Y., & Yang, W. S. (2000). Mining frequent temporal patterns in process databases. *Proceedings of international workshop on information technologies and systems, Australia: Brisbane* (pp. 175–180).