Faculty of Engineering and Information Technology
University of Technology, Sydney

Debt Detection and Debt Recovery with Advanced Classification Techniques

A thesis submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

by

Shanshan Wu

October 2015

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

_____

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Dr. Ling Chen, for the continuous support of my Ph.D thesis, whose expertise, understanding, and patience added considerably to my thesis completion. I attribute the level of my degree to her encouragement and effort and without her this thesis, too, would not have been completed or written.

My sincere thanks also goes to my co-supervisor, Prof. Chengqi Zhang, for offering me the scholarship opportunities and leading me working on the business project in Centrelink. His advice on both research as well as on my career have been priceless. He has supported me thoughout my Ph.D study with his patience and knowledge whilst allowing me the room to work in a flexible time frame.

This thesis was supported by the Australian Research Council (ARC) Linkage Project between University of Technology, Sydney, Australia and Centrelink Australia. I worked on site in Centrelink Canberra for about one and half years. It is one of the most enjoyable working experience I've ever had. A very special thanks goes out to Dr. Yanchang Zhao and Dr. Huaifeng Zhang, who provided me with direction, shared their truthful and illuminating views on a number of issues and became more of a mentor and friend. I must also acknowledge Mr. Hans Bohlscheid, Mr. Peter Newbigin and Mr. Brett Clark from Business Integrity Programs Branch, Centrelink, Australia for their support of domain knowledge and helpful suggestions.

Last but not the least, I would like to thank my family for the support they provided me through my entire life. In particular, I must acknowledge

my mum and my husband, without whose love, encouragement and support, I would not have finished this thesis. At the end I would like express appreciation to my beloved son, Jayden, whose birth lights up my life and makes me more powerful to complete the thesis.

Contents

List of Figures

List of Tables

Abstract

My study is part of an ARC linkage project between University of Technology, Sydney and Centrelink Australia, which aims to applying data mining techniques to optimise the debt detection and debt recovery. A debt indicates an overpayment made by the government to a customer who is not entitled to that payment.

In social security, an interaction between a customer and the government department is recorded as an activity. Each customer's activities happen sequentially along the time, which can be regarded as a sequence. Based on the experience of debt detection experts, there are usually some patterns in the sequence of activities of customers who commit debts. The patterns indicating the customers' intention to be overpaid can thus be used to discover or predict debt occurrence. The development of debt detection and recovery over sequential transaction data, however, is a challenging problem due to following reasons. (1) The size of transaction data is vast, and the transaction data are being generated continuously as the business goes on. (2) Transaction data are always time stamped by the business system, and the temporal order of the transaction data is highly related to the business logic. (3) The patterns and relationships hidden behind the transaction data may be affected by a lot of factors. They are not only dependent on business domain knowledge, but also subject to seasonal and social factors outside the business. Based on a survey of existing methods on debt detection and recovery, data mining techniques are studied in this thesis to detect and recovery debt in an adaptive and efficient fashion.

Firstly, sequence data is used to model the evolvement of customer activities, and the sequential patterns generalize the trends of sequences. For long running sequence classification issues, even if the sequences come from the same source, the sequential patterns may vary from time to time. An adaptive sequential classification model is to be built to make the sequence classification adapt to the sequential pattern variation. The model is applied to 15,931 activity sequences from Centrelink which includes 849,831 activity records. The experimental results show that the proposed adaptive sequence classification framework performs effectively on the continuously arriving data.

Secondly, a new technique of sequence classification using both positive and negative patterns is to be studied, which is able to find the relationship between activity sequences and debt occurrences and also the impact of oncoming activities on the debt occurrence. The same dataset is used for the evaluation. The outcome shows if built with the same number of rules, in terms of recall, the classifier built with both positive and negative rules outperforms traditional classifiers with only positive rules under most conditions.

Finally, decision trees are to be built in the thesis to model debt recovery and predict the response of customers if contacted by phone. The customer contact strategy driven by the model aims to improve the efficiency of debt recovery process. The model is utilized in a real life pilot project for debt recovery in Centrelink. The pilot result outperforms the traditional random customer selection.

In summary, this thesis studies debt detection and debt recovery in social security using data mining techniques. The proposed models are novel and effective, showing potentials in real business.

Chapter 1

Introduction

1.1 Background

Centrelink Australia (http://www.centrelink.gov.au) is the Commonwealth Government agency responsible for delivering social security payments and benefits to the Australian community (*Centrelink annual report 2007-2008* 2008). It is one of the largest data intensive applications in Australia. In the financial year 2008-2009, Centrelink distributed approximately 85 billion dollars in social security payments to 6.5 million customers (*Centrelink annual report 2007-2008* 2008). In Centrelink, qualification for payment of an entitlement is assessed against a customer's personal circumstance. If all criteria are met, the payment to a customer continues until a change of the customer's circumstance precludes the customer from obtaining further benefit. However, for various reasons, customers on benefit payments sometimes get overpaid. A debt indicates an overpayment made by government to a customer who is not entitled to that payment. The overpayment collectively lead to a large amount of debt owed to Centrelink. There are generally two types of debts in social security. The first type is caused by fraud, which is committed when a person knowingly gives false and misleading information to get a payment that she/he should not be receiving or does not tell Centrelink the information that she/he is obliged to provide. Reviews, reports or

other methods could be used to prevent or reduce such kind of debts, so we call them *preventable debts*. The other type includes debts which are caused by the workflow or operator mistakes inside the social security system, such as benefit transfer, compensation charge and so on. We call them *unpreventable debts*. We focus on detecting and preventing preventable debts in this paper.

Data matching survey and random sample survey are the two existing methods to discover debts in Centrelink. Data matching survey compares the Centrelink records with the records from other related government agencies to reduce incorrect payments and avoid potentially large debts. For example, matching Centrelink customer records with data from ATO (short for Australian Taxation Office) could identify possible incorrect or fraudulent payments and detect customers who are either not declaring or under-declaring income. Random sample survey is the analysis of customer circumstances at some time points designed to establish whether a customer is being paid correctly. If some debts are incurred due to a reassessment of a customer's circumstances, the cases can be raised subject to recovery. However, the two methods are neither efficient nor accurate. Firstly, both of them can only discover the debts which already occurred. Neither of them is predictive of debt occurrences, which could help taking actions to prevent debt occurrences as early as possible. Secondly, they can only discover certain types of debts. For example, data matching with records from ATO mostly identify income-related debts. Lastly, both of them are not computationally efficient. Data matching between two large databases takes a lot of time and resources, and most of the efforts are wasted on matching data of non-debtors due to the very low portion of debtors. Also matching data from different organizations could not avoid uncertainty, e.g., matching names and addresses in many variations, in different formats and in different abbreviations. For random sampling, it has the same problem that the surveys are not targeted at high-risk customers, but a random sample of customers. Therefore, it also wastes a lot of efforts on reviewing non-debtors.

Once a debt is raised, a recovery procedure begins by sending a notification letter to the customer. If the customer does not respond within a defined period, a debt recovery team will attempt to contact the customer by out-bound phone calls to discuss debt repayment arrangements. Given that customer selection is largely random and that the human resources of the debt recovery teams are limited, this procedure is considered to be highly inefficient and obsolete. The fact that Centrelinks debt base is increasing rather than decreasing would seem to support this view.

Centrelink raises approximately one billion dollars of customer debt annually. Therefore, it is a very important task undertaken by Centrelink to identify and recover these debts as early as possible and as much as possible.

1.2 Research problem

My study is part of the ARC linkage project between University of Technology, Sydney and Centrelink Australia, which is aiming to applying data mining techniques so as to optimise the debt detection and debt recovery.

In social security, an interaction between a customer and the government department is recorded as an activity. Each customer's activities happen sequentially along the time and can be regarded as a sequence. Based on the experiences of debt detection experts, there are usually some patterns in the activities of customers who commit debts. The patterns indicate the customers' intention to overpayment which can be used to discover or predict debt occurrence. For example, the customer may change her/his date of birth or marital status so as to get a payment she/he is not entitled. So customers' activity sequence can be studied to predict the potential debt occurrence, and then actions can be taken to prevent debt before it happens. Given a list of a customer's activities interacting with Centrelink ordered by interaction timestamp, it is called a *debt* activity sequence if a debt is associated with the customer. Otherwise, it is a *normal* activity sequence. Therefore, debt detection can be modelled as sequence classification on activities sequences.

As mentioned previously, the customer selection in debt recovery is largely random, which is highly inefficient and obsolete. According to the experiences of debt collection experts, customers that have debt repayment history have better chances to place a debt repayment arrangement. Therefore, association analysis could be applied on customer characteristics and historical repayment data to identify the features which are highly related to the debt repayment. Our objective was to predict the response of customers contacted by phone for the purpose of putting in place debt repayment arrangements, thereby maximising debt recovery through out-bound telephone calls.

1.3 Contributions and structure

My research studies some novel data mining techniques on analysing customers activity sequences and demographic characteristics, so as to make the debt detection and debt recovery effective and efficient.

Sequence data records the time-varying information of the data source, and the sequential patterns generalize the trends in sequences. Even if the sequences come from the same source, the sequential patterns may vary from time to time. Therefore, the classifier built on a sequence dataset in the past may not work well on the current sequence dataset, not to mention future datasets. For example, based on our research work in Centrelink Australia, we found that the classifier built on transaction data generated from Jul. 2007 to Feb. 2008 does not work quite well on the new data generated from Mar. 2008 to Sep. 2008, due to the changes of policies, economic situation and other factors. An *adaptive sequence classification* model is proposed to accommodate sequential pattern variation.

The existing sequence classification methods based on sequential patterns consider only positive patterns. However, according to the domain knowledge in social security applications, some sequential patterns negatively related to debt occurrence. In this paper, we introduce negative sequential patterns and then propose a novel technique for *sequence classification using both*

negative and positive sequential patterns. The methodology effectively find the relationship between activity sequences and debt occurrences, and also find the impact of oncoming activities on the debt occurrence.

Decision trees are built on customer characteristics and historical repayment data to improve the debt recovery procedure. The model predicts the response of customers contacted by phone for the purpose of putting in place debt repayment arrangements. By ranking customers based on predicted scores, the results would greatly assist Centrelinks resource-limited debt recovery teams. Based on historical data, our results show that if the top 20 per cent of customers were contacted, approximately 50 per cent of recoveries would be achieved.

Here is the structure of the rest part of the thesis. A detailed review of related works is presented in Chapter 2. And then three data mining algorithms are presented in the following chapters to improve debt detection and debt recovery.

- A a novel adaptive sequence classification framework is proposed for long running sequence classification in the circumstance of time-varying sequential patterns. The proposed framework is to be applied to Centrelink real data to evaluate the effectiveness, comparing with the classification without using the proposed adaptive mechanism.

- A new technique for building sequence classifiers with both positive and negative sequential patterns is presented. This experiment is to test the performance of the classifiers using both positive and negative sequential patterns and compare it with the classifiers built with positive patterns only.

- A decision tree model based on historical data is developed to improve debt recovery in social security. The model will be applied in a debt recovery pilot project in Centrelink, in which test group and control group will be used to evaluate the effectiveness of the proposed model.

Finally, the conclusions and some future work will be discussed in Chapter 6.

Chapter 2

Related Works

In this chapter, the works related to the thesis are reviewed.

2.1 Sequence classification

Sequence classification is an important problem that arises in a wide range of real-world applications such as genomic analysis, information retrieval, health informatics, finance, and abnormal detection.Generally, a sequence is an ordered list of events. And a sequence may carry a class label. For example, a time series of ECG data may come from a healthy or ill person. A DNA sequence may belong to a gene coding area or a non-coding area. Given $\mathcal{L}$ as a set class labels, the task of $sequenceclassification$ (Xing, Pei & Keogh 2010) is to learn a sequence classifier C, which is a function mapping a sequence s to a class label $l \in \mathcal{L}$, written as, C: $s \rightarrow l$, $l \in \mathcal{L}$. There are some major challenges in sequence classification. First, different from the classification task on feature vectors, sequences do not have explicit features. Even with sophisticated feature selection techniques, the dimensionality of potential features may still be very high and the sequential nature of features is difficult to capture. Second, we may also want to get an interpretable classifier with business significance. Building an interpretable sequence classifier is difficult since there are no explicit features. The exist-

ing sequence classification methods can be divided into three large categories (Xing et al. 2010).

- The first category is feature based classification, which transforms a sequence into a feature vector and then apply conventional classification methods. Feature selection plays an important role in this kind of methods.
- The second category is sequence distance based classification. The distance function which measures the similarity between sequences determines the quality of the classification significantly.
- The third category is model based classification, such as using hidden markov model (HMM) and other statistical models to classify sequences.

In the rest of this section, some representative methods in the three categories will be reviewed.

2.1.1 Feature based classification

Conventional classification methods, such as decision trees and neural networks, are designed for classifying feature vectors. One way to solve the problem of sequence classification is to transform a sequence into a vector of features through feature selections.

Pattern-based feature selection is one of the commonly used methods. Lesh *et al.* (Lesh, Zaki & Ogihara 1999) proposed an algorithm for sequence classification using frequent patterns as features in the classifier. In their algorithm, subsequences are extracted and transformed into sets of features. The features are short sequence segments which satisfy the following criteria (1) frequent in at least one class (2) distinctive in at least one class and (3) not redundant. Criterion (2) means a feature should be significantly correlated with at least one class. The redundancy in Criterion (3) can be defined in the way of feature specification and feature generalization. An

efficient feature mining algorithm is proposed to mine features according to the criteria. After feature extraction, general classification algorithms such as Naïve Bayes, SVM or neural network can be used for classification. Their algorithm is the first try on the combination of classification and sequential pattern mining. However, a huge amount of sequential patterns are mined in the sequential mining procedure. Although pruning algorithm is used for the post-processing, there are still a large amount of sequential patterns constructing the feature space. Their experimental results show that using sequence patterns as features can improve the accuracy substantially.

Pattern-based sequence classification commonly follow a two-step strategy. The first step is sequential pattern mining in which a complete set of sequential pattern is discovered given a minimum support. The second step is to select the discriminative patterns as features and build classifiers based on the patterns. Among the existing sequence classification algorithms, efficiency is the major bottleneck because of the following two issues. Firstly, sequential pattern mining is very time-consuming. Suppose we have 150 distinct items. If our aim is to mine for 10-item sequences, the number of candidate sequential patterns is more than

$$150^1 + 150^2 + 150^3 + \ldots + 150^{10} \approx 5 * 10^{21}$$

Even if using efficient algorithms, the sequential pattern mining in the above example would still be quite time-consuming. Secondly, a number of processes, such as pattern pruning and coverage test, have to be applied to the sequential pattern set to build the classifier. If the sequential pattern set contains a huge amount of sequential patterns, the classifier building step can be also extremely time-consuming. In fact, the most important consideration in rule-based classification is not finding the complete rule set, but discovering the most discriminative rules (Cheng, Yan, Han & wei Hsu 2007) (Han, Cheng, Xin & Yan 2007). Cheng *et al.*'s experimental results show that redundant and non-discriminative patterns often overfit the model and deteriorate the classification accuracy (Han et al. 2007). (Shelke & Itkar n.d.) addressed that pattern growth approach is best suitable for further research

effort in this region due to divide and conquer policy, no candidate generation and compressed database.

Ji *et al.* (Ji, Bailey & Dong 2005) propose an algorithm to mine distinctive subsequences with a maximal gap constraint. The algorithm, which uses bisect and boolean operations and a prefix growth framework, is efficient even with a low frequency threshold. Tseng and Lee (Tseng & hui Lee 2005) proposed a Classify-By-Sequence (CBS) algorithm to combine sequential pattern mining and classification. In their paper, two algorithms, CBS-Class and CBS-All were proposed. In CBS-Class, the database is divided into a number of sub-databases according to the class label of each instance. Then sequential pattern mining is implemented on each sub-database. In CBS-All, conventional sequential pattern mining algorithm is used on the whole dataset. Weighted scoring is used in both algorithms. Exarchos (Exarchos, Tsipouras, Papaloukas & Fotiadis 2008) proposed to combine sequential pattern mining and classification followed by an optimization algorithm. The accuracy of their algorithm is higher than that of CBS. However optimization is a very time-consuming procedure.

To identify the features of long sequences, Aggarwal (Aggarwal 2002) uses different wavelet decomposition coefficients to capture both the global and local features of sequences for the purpose of classification.

In contrast to pattern-based feature selection method, several k-gram based feature selection methods for sequence classifications can be found in (Dong 2009). A short sequence segment of k consecutive elements, called a k-gram, is usually selected as a feature. Given a set of k-grams, a sequence can be represented as a vector of the presence and the absence of the k-grams or as a vector of the frequencies of the k-grams. By using k-grams as features, sequences can be classified by a conventional classification method, such as SVM (Leslie, Eskin & Noble 2002)(Leslie, Kuang & Bennett 2004) and decision trees (Chuzhanova, Jones & Margetts 2010).The size of candidate features which are all k-grams where $1 \leq k \leq l$ is $2^l - 1$. If k is a large number, the size of the feature set can be huge. Since not all features are equally useful

for classification, Chuzhanova *et al.*. (Chuzhanova et al. 2010) proposed to use Gamma (or near-neighbour) test to select features from l-grams over the alphabet. The method was used to classify the large subunits rRNA, and the nearest-neighbor criterion was used to estimate the classification accuracy based on the selected features. Recently, (Raju & Varma n.d.)propose an algorithm CSpan for mining closed sequential patterns. CSpan uses a new pruning method called occurrence checking that allows the early detection of closed sequential patterns during the mining process. Closed sequential pattern mining extensively reduces the number of patterns produced and it can be utilized to obtain the complete set of sequential patterns.

As big data become an active and front edge research topic in data management area. Discovering frequent patterns hiding in a big dataset has application across a broad range of use cases. Sequential pattern mining has been introduced to industry products. In Spark 1.5, frequent pattern mining capabilities have been significantly improved by adding algorithms for association rule generation and sequential pattern mining (Liang, Zhang, Tu & Meng 2015).

2.1.2 Sequence distance based classification

Sequence distance based sequence classification defines a distance function to measure the similarity between a pair of sequences. Once such a distance function is obtained, existing classification methods, such as K nearest neighbor classifier(KNN) and SVM can be applied for sequence classification.

The Euclidean distance is a widely adopted distance function for sequence classification. Given two sequence s_1 and s_2, Euclidean distance is

$$dist(s_1, s_2) = \sqrt{\sum_{i=1}^{L}(s_1[i] - s_2[i])^2}$$

which requires the two sequence are of the same length. Given a labeled sequence dataset T, a positive integer k, and a new sequence s to be classified, the KNN classifier finds the k nearest neighbors of s in T and returns the

dominating class label in the k nearest neighbors as the label of s. Euclidean distance outperforms other more complex similarity measures when applying 1NN classifier on time series (Keogh & Kasetty 2003).

Dynamic time warping distance(DTW) (Keogh & Pazzani 2000) is a distance measure which does not require two sequences to be of the same length. DTW is to align two sequences and get the best distance by aligning. (Xi, Keogh, Shelton, Wei & Ratanamahatana 2006) shows that on small datasets, elastic measures such as DTW can be more accurate than Euclidean distance.

The idea of applying SVM on sequence data is to map a sequence into a feature space and find the maximum-margin hyperplane to separate two classes. SVM has been proved to be an effective method for sequence classification (Lodhi, Saunders, Shawe-Taylor, Cristianini & Watkins 2002) (Leslie et al. 2002) (Leslie et al. 2004). Given two sequences, s_1,s_2, some kernel functions, $K(s_1, s_2)$, can be viewed as the similarity between two sequences. The challenges of applying SVM to sequence classification is how to define feature spaces or kernel functions, and how to speed up the computation of kernel matrixes. Lei and Govindaraju (Lei & Govindaraju 2005) proposed to use an intuitive similarity measure, ER^2, for multi-dimensional sequence classification based on SVM. The measure is used to reduce classification computation and speed up the decision-making of multi-class SVM.

2.1.3 Model based classification

Another category of sequence classification is based on generative models, which assume sequences in a class are generated by an underlying model. Given a class of sequences, the probability distribution of the sequences in the class is captured by a model M. In the training stage, the parameters of model M are learned. In the classification stage, a new sequence is assigned to the class with the highest likelihood.

Naive Bayes sequence classifier has been widely used in text classification (Kim, Han, Rim & Myaeng 2006) and genomic sequences classification (BY, JG & J. 2005).

A discriminatively trained Markov Model (MM(k-1)) for sequence classification was proposed by Yakhnenko *et al.* (Yakhnenko, Silvescu & Honavar 2005). Their experimental results show that their classifiers are comparable in accuracy and more efficient than Support Vector Machines trained by k-gram representations of sequences.

Wu *et al.* (Wu, Berry, Fung & McLarty 1993) proposed a neural network classification method for molecular sequence classification. The molecular sequences are encoded into input vectors of a neural network classifier, by either an n-gram hashing method or a SVD (Singular Value Decomposition) method.

2.2 Fraud Detection

Applications similar to debt detection include fraud detection, terrorism detection, financial crime detection, network intrusion detection and spam detection. Different from transactional fraud detection which attempts to classify a transaction or event as being legal or fraud, our techniques try to predict the likelihood of a customer being fraud based on his past activities. It is at customer level instead of transaction level. Moreover, in our application, detection is only part of the work. Based on the patterns in transactional data, we also attempt to find out what measures can be taken to prevent debt occurrences. For example, if a customer is of high risk of debt, we also find out what can be done to reduce the risk of debt occurrence.

Whitrow *et al.* (Whitrow, Hand, Juszczak, Weston & Adams 2008) proposed transaction aggregation for credit card fraud detection, arguing that it is impractical to use all transactions in a fraud detection system with traditional transaction-level detection. By aggregating information over a succession of transactions or over a period of time, it may result in better fraud detection than transaction-level classification. They evaluated their technique on credit and debit card transactions from banks, which showed that the aggregation may lead to better fraud classification, but it is impor-

tant to select appropriate aggregation periods.

Fast *et al.* designed relational data pre-processing techniques to improve securities fraud detection (Fast, Friedland, Maier, Taylor, Jensen, Goldberg & Komoroske 2007). Their techniques try to find the associate individuals with branch office locations and infer professional associations by exploiting employment histories from data of the National Association of Securities Dealers, USA. This application exploits the relationship between dealers, not on customer transactions and activities.

Bonchi *et al.* (Bonchi, Giannotti, Mainetto & Pedreschi 1999) proposed a classification-based methodology for planning audit strategies in fraud detection and presented a case study to illustrate how classification techniques can be used to support the task of planning audit strategies. The models are constructed by analyzing historical audit data. Then, the models are used to plan effectively future audits for the detection of tax evasion. A decision tree algorithm, $C5.0$, was used in their case study. Although the target problem is similar with ours, the data used are different. What we used are transactional data which record activities related to customers. Because the time order in activities is important for predicting debt occurrences, sequence classifiers instead of decision trees are used in our application.

Fawcett and Provost designed an adaptive fraud detection system for automatically generating detectors for superimposition fraud and applied it to detect cellular cloning fraud in telecommunications industry (Fawcett & Provost 1997). A rule-learning program was used to find indicators of fraudulent behaviour from customer transactions, and monitors are created based on the indicators to profile legitimate customer behaviour. Then the outputs of monitors are used to generate alarms. The rules they generated are associations.

Rosset *et al.* (Rosset, Murad, Neumann, Idan & Pinkas 1999) studied the fraud detection in telecommunications and presented a two-stage system based on $C4.5$ to find fraud rules. They adapted the $C4.5$ algorithm for generating rules from bi-level data, i.e., customer data and behaviour-level

data. However, the behaviour data they used is the statistics in a short time frame, such as the number of international calls in a day and total duration of all calls in a day, which is different from the sequential patterns in our techniques.

Phua *et al.* proposed a technique for fraud detection where data distributions are skewed and tested it on automobile insurance data (Phua, Alahakoon & Lee 2004). Their method uses backpropagation, naive Bayesian and C4.5 algorithms on data partitions derived from minority oversampling.

Virdhagriswaran and Dakin designed a data mining system for camouflaged fraud detection and presented an application on accounting fraud by companies (Virdhagriswaran & Dakin 2006). Decision trees, logistic regression and k-means clustering are used in their system.

Julisch and Dacier (Julisch & Dacier 2002) used techniques of episode rules and conceptual clustering to mine historical alarms for network intrusion detection. Their episode rules are designed to predict the occurrence of certain alarms based on other alarms. Although it looks similar to the rules in our techniques, there are several critical differences as follows. First, our combined patterns are composed of not only episode rules, but also association rules, e.g., those rules built from demographic attributes. Second, negative relationship are also included in our techniques. Third, the impact of a single activity in a sequential rule is also evaluated in our techniques. Last, we built a classifier based on the learned rules.

2.3 Negative Sequential Pattern Mining

Since sequential pattern mining was first proposed in (Agrawal & Srikant 1995), a few sequential methods have been developed, such as GSP (Generalized Sequential Patterns) (Srikant & Agrawal 1996), FreeSpan (Han, Pei, Mortazavi-Asl, Chen, Dayal & Hsu 2000), PrefixSpan (Pei, Han, Mortazavi-Asl, Pinto, Chen, Dayal & Hsu 2001), SPADE (Zaki 2001) and SPAM (Ayres, Flannick, Gehrke & Yiu 2002). Most of the sequential pattern mining algo-

rithms focus on the patterns appearing in the sequences, i.e., the positively correlated patterns. However, the absence of some items in sequences may also be interesting in some scenarios. For example, in social welfare, the lack of follow-up examination after the address change of a customer may result in overpayment to the customer. Such kind of sequences with the non-occurrence of elements are negative sequential patterns. Limited studies have addressed this issue.

Sun *et al.* (Sun, Orlowska & Li 2004) proposed negative event-oriented patterns in the form of $\neg P \xrightarrow{T} e$, where e is a target event, P is a negative event-oriented pattern, and the occurrence of P is unexpectedly rare in T-sized intervals before target events. It is a special case of negative sequential pattern $\neg A \rightarrow B$ in our framework. In (Sun et al. 2004), P is supposed to be an "existence pattern" (i.e., a frequent itemset without time order), instead of a sequential pattern, although it is claimed that the discussion can be extended to sequential patterns.

Bannai *et al.* (Bannai, Hyyro, Shinohara, Takeda, Nakai & Miyano 2004) proposed a method for finding the optimal pairs of string patterns to discriminate between two sets of strings. The pairs are in the forms of $p' \wedge q'$ or $p' \vee q'$, where p' is either p or $\neg p$, q' is either q or $\neg q$, and p and q are two substrings. Their concern is whether p and q appear in a string s. The substring can be taken as a special case of sequential pattern, since the elements in substrings can only be continuous.

Ouyang and Huang proposed the notion of negative sequences (Ouyang & Huang 2007) as $(A, \neg B)$, $(\neg A, B)$ and $(\neg A, \neg B)$. Negative sequential patterns are derived from infrequent sequences. A drawback is that both frequent and infrequent sequences have to be found at the first stage, which demands a large amount of space.

Lin *et al.* (Lin, Chen & Hao 2007) designed an algorithm NSPM (Negative Sequential Patterns Mining) for mining negative sequential patterns. In their negative patterns, only the last element can be negative, and all other elements are positive.

Chapter 3

Adaptive Sequence Classification for Debt Detection

3.1 Introduction

From a data mining perspective, sequence classification is to build a classifier using frequent sequential patterns. Most of the conventional frequent pattern based classifications follow two steps. The first step is to mine a complete set of sequential patterns given a minimum support. The second step is to select a number of discriminative patterns to build a classifier. In most cases, mining a complete set of sequential patterns from a large dataset is extremely time-consuming. The discovered huge number of patterns make the pattern selection and classifier building very computationally expensive. In fact, finding the most discriminative patterns is more important than discovering the complete set of sequential patterns. In this chapter, a novel measure, *contribution weight*, is proposed to select the discriminative patterns. The discriminative power of frequent patterns are dynamically evaluated by applying the patterns to a set of evaluation data. The interestingness measure of frequent patterns are refined by contribution weight, so

as to let the discriminative patterns pop up.

Moreover, sequence data represent the evolvement of data sources, and the sequential patterns generalize the trends of sequences. For long running sequence classification issues, even if the sequences come from the same source, the sequential patterns may vary from time to time. Therefore, the classifier built on a sequence dataset in the past may not work well on the current sequence dataset, not to mention future datasets. For example, based on our previous research work in Centrelink Australia, we found out that the classifier built on transaction data generated from Jul. 2007 to Feb. 2008 does not work quite well on the new data generated from Mar. 2008 to Sep. 2008, due to the changes of policies, economic situation and other social influences. Therefore, it is significant to improve the sequence classification to make it adapt to the sequential pattern variation. The most direct way is to rebuild the classifier with the latest training dataset. However, the training is quite a time-consuming process. If the pattern variation is not so great, an incremental update would be much more efficient than rebuilding the classifier. In this chapter, an adaptive sequence classification framework is proposed to tackle the above problem. The adaptive model adapts the classifier in a timely fashion by adopting the proposed discriminative pattern boosting strategy, so as to catch up with the trends of sequential pattern variation and improve the classification accuracy.

There are three main contributions in this chapter. Firstly, a novel method to boost discriminative frequent patterns for sequence classification is proposed, which improves the accuracy of classifier. Secondly, an adaptive sequence classification model is suggested to improve the sequence classification performance on time-varying sequences. Lastly, the adaptive strategies are applied to a real-world application, which shows the efficiency and effectiveness of the proposed methods.

3.2 Problem Statement

Let $\mathcal{S}$ be a sequence database, in which each sequence is an ordered list of *elements*. These elements can be either *simple items* from a fixed set of items, or *itemsets*, that is, non-empty subsets of items. The list of elements of a data sequence s is denoted by $< s_1, s_2, \cdots, s_n >$, where s_i is the i^{th} element of s.

Consider two sequences $s =< s_1, s_2, \cdots, s_n >$ and $t =< t_1, t_2, \cdots, t_m >$. We say that s is a subsequence of t if s is a "projection" of t, derived by deleting elements and/or items from t. More formally, s is a subsequence of t if there exist integers $j_1 < j_2 < \ldots < j_n$ such that $s_1 \subseteq t_{j1}$, $s_2 \subseteq t_{j2}$, ..., $s_n \subseteq t_{jn}$. Note that for sequences of simple items the above condition translates to $s_1 = t_{j1}$, $s_2 = t_{j2}$, ..., $s_n = t_{jn}$. A sequence t is said to *contain* another sequence s if s is a subsequence of t, in the form of $s \subseteq t$.

3.2.1 Frequent Sequential Patterns

The number of sequences in a sequence database $\mathcal{S}$ containing sequence s is called the support of s, denoted as $sup(s)$. Given a positive integer min_sup as the support threshold, a sequence s is a frequent sequential pattern in sequence database $\mathcal{S}$ if $sup(s) \geq min\_sup$. The sequential pattern mining is to find the complete set of sequential patterns with respect to a given sequence database $\mathcal{S}$ and a support threshold min_sup.

3.2.2 Classifiable Sequential Patterns

Definition 3.1 (Sequence Classifier). *Let $\mathcal{T}$ be a finite set of* class labels. *A* sequence classifier *is a function*

$$\mathcal{F} : \mathcal{S} \rightarrow \mathcal{T} \tag{3.1}$$

In sequence classification, the classifier $\mathcal{F}$ is built on the base of *classifiable sequential patterns* $\mathcal{P}$, where $\mathcal{P}$ is a set of frequent patterns associated with labels given a certain min-support threshold.

Definition 3.2 (Classifiable Sequential Pattern). *Classifiable Sequential Patterns (CSP) are frequent sequential patterns for the sequence classifier in the form of $p_a \Rightarrow \tau$, where p_a is a frequent pattern in the sequence database $\mathcal{S}$.*

Based on the mined classifiable sequential patterns, a sequence classifier can be formulized as

$$\mathcal{F} : s \xrightarrow{\mathcal{P}} \tau. \tag{3.2}$$

That is, for each sequence $s \in \mathcal{S}$, $\mathcal{F}$ predicts the target class label of s based on the sequence classifier built with the classifiable sequential pattern set $\mathcal{P}$. Suppose we have a classifiable sequential pattern set $\mathcal{P}$. A sequence instance s is said to be *covered* by a classifiable sequential pattern $p \in \mathcal{P}$ if s contains the classifiable sequential pattern p.

3.2.3 Discriminative Sequential Patterns

Given a sequence database $\mathcal{S}$ and a set of target classes $\mathcal{T}$, we need to mine for a number of frequent classifiable sequential patterns to build a sequence classifier. Instead of using the complete pattern set, we select a small set of discriminative classifiable sequential patterns according to Class Correlation Ratio (CCR) (Verhein & Chawla 2007).

CCR measures how much the sequential pattern p_a is correlated with the target class τ compared to negative class $\neg\tau$. Based on the contingency table shown in Table 3.1, it is defined as

$$CCR(p_a \to \tau) = \frac{\hat{corr}(p_a \to \tau)}{\hat{corr}(p_a \to \neg\tau)} = \frac{a \cdot (c+d)}{c \cdot (a+b)} \tag{3.3}$$

where $\hat{corr}(p_a \to \tau)$ is the correlation between p_a and the target class τ, defined as

$$\hat{corr}(p_a \to \tau) = \frac{sup(p_a \cup \tau)}{sup(p_a) \cdot sup(\tau)} = \frac{a \cdot n}{(a+c) \cdot (a+b)}.$$

CCR falls in $[0, +\infty)$. $CCR = 1$ means the antecedent is independent of the target class. $CCR < 1$ indicates the antecedent is negatively correlated

Table 3.1: Feature-Class Contingency Table

	p_a	$\neg p_a$	$\sum$
τ	a	b	$a+b$
$\neg\tau$	c	d	$c+d$
$\sum$	$a+c$	$b+d$	$n=a+b+c+d$

with the target class, and $CCR > 1$ suggests a positive correlation between them.

In order to use the mined classifiable sequential patterns to build a classifier, we need a ranking (ordering) of the patterns that captures the ability of the pattern to make correct classification. The ranking is based on a weighted score

$$W_s = \begin{cases} CCR, & \text{if } CCR \geq 1 \\ \frac{1}{CCR}, & \text{if } 0 < CCR < 1 \\ M, & \text{if } CCR = 0 \end{cases}, \tag{3.4}$$

where M is the maximum CCR of all rules.

3.3 Discriminative Frequent Patterns Boosting

In order to find the most discriminative patterns rather than discover the complete set of sequential patterns, the concept of contribution weight is introduced in this section to describe the discriminative power of a classifiable sequential pattern on a given dataset.

Given a dataset, the more samples a pattern can correctly classify, the more discriminative the pattern is on the dataset. In other words, the more samples a pattern incorrectly classifies, the less discriminative the pattern is on the dataset. To make it more statistically significant, the definitions of *positive contribution ability* and *negative contribution ability* are given as follows.

Definition 3.3 (Positive Contribution Ability). *Given a dataset S, the Positive Contribution Ability (PCA) of pattern P is the proportion of samples that can be correctly classified by P out of all the samples in dataset S.*

Definition 3.4 (Negative Contribution Ability). *Given a dataset S, the Negative Contribution Ability (NCA) of pattern P is the proportion of samples that are incorrectly classified by P out of all the samples in the dataset S.*

For a classifiable sequential pattern P in the form of $p_a \Rightarrow \tau$, PCA of P on S can be denoted as

$$PCA_S(P) = \frac{\|\{s|p_a \subseteq s \wedge s \in S_\tau\}\|}{\|S\|}, \tag{3.5}$$

and NCA of pattern P on S can be denoted as

$$NCA_S(P) = \frac{\|\{s|p_a \subseteq s \wedge s \in S_{\neg\tau}\}\|}{\|S\|}, \tag{3.6}$$

where S_τ and $S_{\neg\tau}$ represent the subsets of S in which samples are of class τ and are not of class τ, respectively.

Above all, PCA and NCA describe the classification ability of patterns on a given dataset. In order to enhance classification performance, it is intuitive to boost the patterns with higher PCA and lower NCA, while depress those with lower PCA and higher NCA. Thereafter, a measure of *Contribution Weight* is proposed to measure the discriminative power that a pattern contributes to the classification on a dataset.

Definition 3.5 (Contribution Weight). *Given a dataset S, Contribution Weight of a classifiable sequential pattern P is the ratio of Positive Contribution Ability $PCA_S(P)$ on S and Negative Contribution Ability $NCA_S(P)$ on S. It can be denoted as*

$$CW_S(P) = \frac{PCA_S(P)}{NCA_S(P)} = \frac{\|\{s|p_a \subseteq s \wedge s \in S_\tau\}\|}{\|\{s|p_a \subseteq s \wedge s \in S_{\neg\tau}\}\|}. \tag{3.7}$$

The proposed measure of contribution weight tells the relative discriminative power of a classifiable sequential pattern on a given dataset, which is

based on the classification performance of the pattern on the dataset. According to the definition, contribution weight has following characters.

- The greater the value of contribution weight is, the more discriminative a pattern is on a given dataset, and vice versa.
- Contribution weight is a measure with regard to a dataset on which classification performance is evaluated.
- Contribution weight is independent of the algorithm that is used for classifiable sequential pattern mining, and it does not matter which interestingness measure is used for classification.

Therefore, contribution weight is introduced as a factor to boost the discriminative frequent patterns on a certain dataset. The term of *Boosted Interestingness Measure* is defined as follows.

Definition 3.6 (Boosted Interestingness Measure). *For a classifiable sequential pattern P with an interestingness measure R, the corresponding Boosted Interestingness Measure on dataset S is denoted as*

$$R_S^* = R \times CW_S(P). \tag{3.8}$$

In other words, boosted interestingness measure of a pattern can be regarded as a weighted interestingness measure, and the weight tells how much contribution the corresponding pattern can make to the classification on the given dataset. Patterns that are more discriminative on a given dataset are strongly boosted by higher contribution weights, and vice versa. From this point of view, boosted interestingness measure adjusts the original interestingness measure so as to make it indicating the discriminative ability of classifiable sequential patterns on the given dataset more vividly.

3.4 Adaptive Sequence Classification Framework

In order to catch up with the pattern variation over time, an adaptive sequence classification framework is introduced in this section. The main idea of the adaptive framework is to include the latest pattern into the classifier with the proposed boosted interestingness measure, so as to improve the classification performance on dataset of near future.

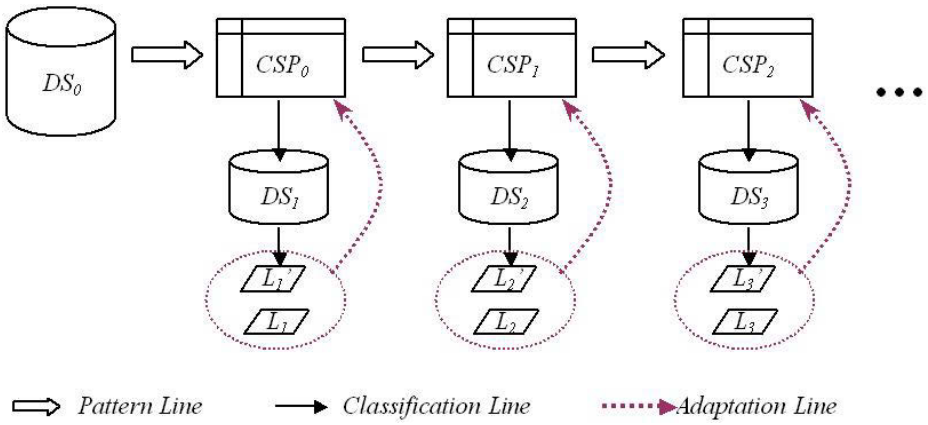

Figure 3.1: Architecture of Adaptive Sequence Classification

As illustrated in Figure 3.1, the initial classifiable sequential pattern set CSP_0 is extracted from the dataset DS_0, and then is used to perform prediction/classification on coming dataset DS_1 and get the predicted labels $L_1^{'}$. Once L_1, the real class labels of dataset DS_1 , is available, interestingness measure of the classifier CSP_0 could be refined and CSP_0 evolves into CSP_1 with boosted interestingness measure, which brings the timely trends of patterns in dataset DS_1 into the classification model. The boosted classifier will be applied to continuously coming dataset for prediction/classification. The procedure goes on as dataset updates all along, which is generalized in

Data: Dataset DS_i and corresponding real labels L_i that are available after classification/prediction, $i = 0, 1, ...$
Basic classification algorithm$F(F_1$:Classifier construction; F_2:Classifying)
Result: Predicted labels $L_i^{'}$, $i = 1, 2, ...$
Classifiers CSP_i, $i = 0, 1, 2, ...$

```
1  begin
2      CSP_0 = F_1(DS_0, L_0)
3      i = 1
4      while i do
5          L_i' = F_2(DS_i, CSP_{i-1})
6          Wait till L_i is available
7          Modify CSP_{i-1} with R*_{(DS_i,L_i)} to get CSP_i
8          i = i + 1
9      end
10 end
```

Algorithm 1: Adaptive classification model.

Algorithm 1. The boosted classifier CSP_i, $i = 1, 2, ...$ not only takes the latest pattern variation into the classification model, but also tracks the evolvement of the patterns ever since the initial classifier is built. Therefore, the performance of classification is expected to outperform that of the initial classifier.

Since the adaptive model is based on boosted interestingness measure, it inherits the properties of boosted interestingness measure congenitally. To be more precise, it is independent of interestingness measure and classifiable sequence mining method.

3.5 Case Study

The proposed algorithm has been applied in a real world business application in Centrelink, Australia. The purpose of the case study is to predict and

further prevent debt occurrence based on customer transactional activity data. In this section, the dataset used for debt prediction in Centrelink is described firstly. Then a pre-experiment is given to evaluate the effectiveness of discriminative pattern boosting strategy, followed by the experimental results of adaptive sequence classification framework.

3.5.1 Data Description

The dataset used for sequence classification is composed of customer activity data and debt data. In Centrelink, every single contact (e.g., a circumstance change) of a customer may trigger a sequence of activities running. As a result, large volumes of activity based transactions are encoded into 3-character "Activity Code" and are recorded in activity transactional files. In the original activity transactional table, each activity has 35 attributes, in which 4 attributes are used in the case study. These attributes are "CRN" (Customer Reference Number) of a customer, "Activity Code", "Activity Date" and "Activity Time", as shown in Table 3.2. We sort the activity data according to "Activity Date" and "Activity Time" to construct the activity sequence. The debt data consist of the "CRN" of the debtor and "Debt Transaction Date". In our case study, only the activities of a customer before the occurrence of his/her debt are kept for the sequence classification.

There are 155 different activity codes in sequences. Different from supermarket basket analysis, every transaction in this application is composed of one activity only. The activities in four months before a debt were believed by domain experts to be related to debt occurrence. If there were no debts for a customer during the period from July 2007 to February 2008, the activities in the first four months were taken as a sequence associated with no debts. After data cleaning and preprocessing, there are 15,931 activity sequences including 849,831 activity records used in this case study.

3.5.2 Effectiveness of Boosting Discriminative Patterns

In order to evaluate the effectiveness of discriminative patterns boosting, two groups of experiments are presented in this section. In both groups, we compare the performance of classification which uses discriminative pattern boosting strategy with that does not boost discriminative patterns. In group (a), the activity sequence data generated from Jul. 2007 to Oct. 2007 are used. After data cleaning, there are $6,920$ activity sequences including $210,457$ activity records used. The dataset is randomly divided into the following 3 portions.

- Training data(60%): To generate the initial classifier.
- Evaluation data(20%): To refine classifier.
- Test data(20%): To test the performance of classification.

While in group (b), some data generated in Nov. 2007 is added to the evaluation data and test data, expecting to include some pattern variation.

Table 3.2: Centrelink Data Sample

CRN	Act_Code	Act_Date	Act_Time
******002	DOC	20/08/07	14:24:13
******002	RPT	20/08/07	14:33:55
******002	DOC	05/09/07	10:13:47
******002	ADD	06/09/07	13:57:44
******002	RPR	12/09/07	13:08:27
******002	ADV	17/09/07	10:10:28
******002	REA	09/10/07	7:38:48
******002	DOC	11/10/07	8:34:36
******002	RCV	11/10/07	9:44:39
******002	FRV	11/10/07	10:18:46
******002	AAI	07/02/08	15:11:54

According to the property of contribution weight, the boosted interestingness measure is independent of basic classification. Therefore, we use the classification algorithm proposed in the work (Zhang, Zhao, Cao, Zhang & Bohlscheid 2009) to generate the initial classifier on the training dataset. And we use confidence as the base interestingness measure. For classification which uses boosting strategy, the evaluation dataset is used to refine the initial classifier, and the refined classifier is evaluated on the test dataset. While for the classification that does not boost discriminative patterns, we combine training data and evaluation data to generate the initial classifier, and then apply the initial classifier to the test dataset for debt prediction.

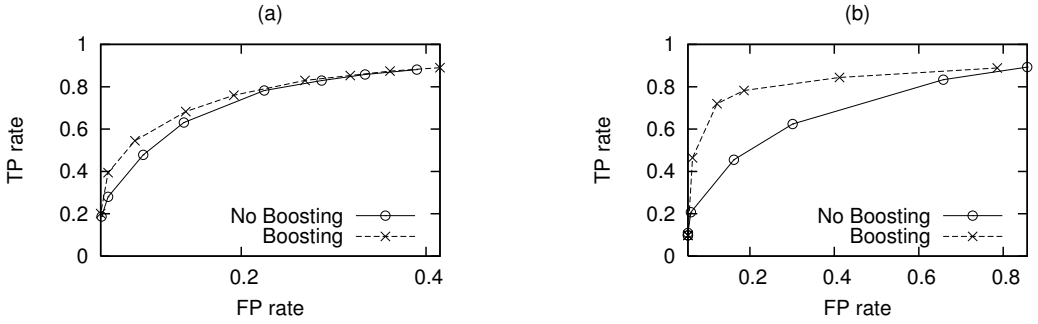

Figure 3.2: Effectiveness of Discriminative Patterns Boosting

ROC curve (Receiver Operating Characteristic) is used to plot the fraction of true positives *vs*. the fraction of false positives of each classifier. The best possible prediction method would yield a point in the upper left corner or coordinate (0,1) of the ROC space, representing 100% sensitivity (no false negatives) and 100% specificity(no false positives). Therefore, the more close to the upper left corner the curve is, the better the classification method is. As illustrated in Figure 3.2, the boosted classifier outperforms the classifier without boosting in both experiments. In group (a), training data, evaluation data and test data all come from the dataset generated in the same time period. By boosting discriminative patterns with evaluation data, classification power of initial classifier is refined by boosting discriminative patterns and depressing less discriminative patterns, so it outperforms the classifica-

tion without boosting. As for group (b), since some new data generated in different time period is added to the evaluation data and test data, some pattern variation might be included in the corresponding dataset. In this circumstance, the proposed boosting strategy notices the pattern variation in the updated dataset, refines the interestingness measure of the classifiers with evaluation data, and performs much better in the test data than the classifier without boosting.

In all, the discriminative pattern boosting strategy improves the classification performance, especially when the sequence data evolves with pattern variation.

3.5.3 Performance of Adaptive Sequence Classification Framework

In this subsection, the adaptive sequence classification framework will be evaluated on the sequence datasets obtained with a sliding window applied on the activity sequence data. After applying sliding window on the sequences generated from Jul. 2007 to Aug. 2008, we get 11 windows listed in Table 3.3.

Following the framework proposed in Section 3.4, the classification in the work (Zhang et al. 2009) is firstly applied on W_0 and the initial classifier CSP_0 is generated. By discriminative pattern boosting with W_1, CSP_0 is refined to CSP_1 and then is applied to make debt prediction on W_2. Here we still use confidence as the base interestingness measure. The debt prediction performance on W_2 is illustrated in the first graph in Figure 3.3. Thereafter, CSP_1 is boosted with sequence data in W_2, and the generated CSP_2 is applied on W_3 to predict debt occurrence. As the procedure goes on continuously, the debt prediction performance on all the following windows are listed in Figure 3.3, which is represented by the ROC curves labelled *Adaptation_all_along*. In order to evaluate the performance of adaptive sequence classification framework, debt prediction on each window is also performed with initial classifier CSP_0, whose performance is denoted by

the ROC curves labelled *No_adaptation*. According to Figure 3.3, we can tell that the proposed adaptive framework outperforms the initial classifier in debt prediction on continuously coming datasets. Since the classifier is continuously updated with the latest data, it captures the pattern variation in the new dataset and then works well on the debt prediction on the on-coming dataset. Meanwhile, we apply CSP_1, which is boosted once based on initial classifier, to each of the windows and get the performance denoted by the curves labelled *Adaptation_once*. The classifier boosted once still outperforms the initial classifier. While it does not contains the pattern information in the latest datasets, its performance is always worse than that of *Adaptation_all_along* strategy.

Above all, conclusion could be drawn that the proposed adaptive sequence classification framework updates the classifier with new data, includes the sequence pattern variation in the new data, and performs effectively on the continuously arriving data.

Table 3.3: Data Windows

Window	Start Date	End Date
W0	02/07/07	31/10/07
W1	01/08/07	30/11/07
W2	01/09/07	31/12/07
W3	01/10/07	31/01/08
W4	01/11/07	29/02/08
W5	01/12/07	31/03/08
W6	01/01/08	30/04/08
W7	01/02/08	31/05/08
W8	01/03/08	30/06/08
W9	01/04/08	31/07/08
W10	01/05/08	31/08/08

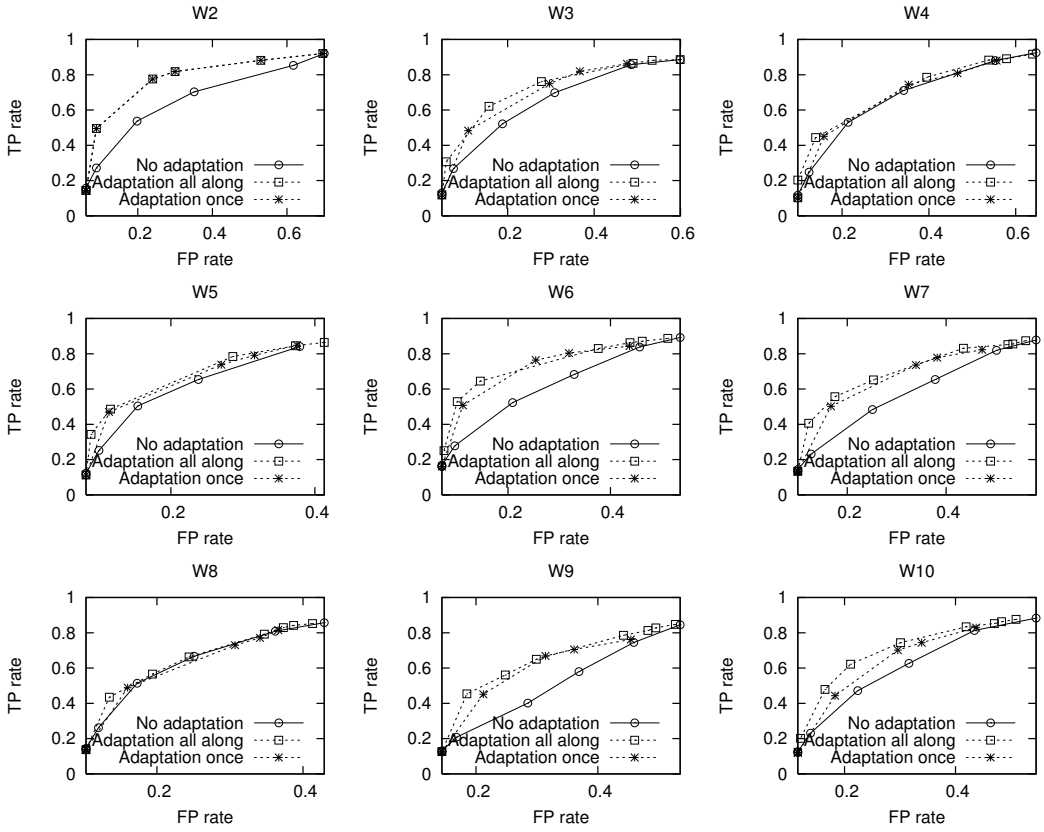

Figure 3.3: RoC curves of Adaptive Sequence Classification Framework

3.6 Conclusion and Future Work

In this chapter a novel adaptive sequence classification framework was proposed for long running sequence classification in the circumstance of time-varying sequence patterns. In order to make the classifier catch up with the latest sequence pattern variation, a discriminative pattern boosting strategy which boosts discriminative patterns and depresses less discriminative patterns based on the latest sequential data was introduced. The proposed methods were tested on a real-world dataset, and the case study showed the effectiveness of the proposed strategy.

The adaptive framework refines the classifier round by round, and in each round the adaptation is based on the classifier generated in the previous round. Though it tracks the evolvement of sequential patterns, the latest

pattern variation is given the same consideration as the previous ones. In the future work, we will study how to assign tilted weight to the historical data, which may include more latest sequence pattern characteristics into the classification model.

Chapter 4

Debt Detection by Sequence Classification Using Both Positive and Negative Patterns

4.1 Introduction

Debt detection can be generally modelled as a fraud detection problem. Therefore, we can adopt a classification approach. All transactions about a customer form a transaction sequence. If no debt happens to a customer, the sequence is labelled as normal (i.e., no-debt). If a debt happens to a customer, the corresponding customer sequence is labelled as debt. We can collect a training set containing both no-debt and debt sequences and learn a sequence classifier. The classifier can be applied to new customer sequences to detect possible debts.

A classifier needs to extract features for classification. Since sequences are the data objects in debt detection, it is natural to use sequential patterns, i.e., subsequences that are frequent in customer sequences, as features. The traditional techniques for sequential pattern based classifiers consider only positive patterns, where a positive pattern captures a set of positively correlated events. Moreover, to detect debt at an early stage and prevent debt

occurrence, a classification model is needed to predict the likelihood of debt occurrence based on the transactional activity data. Nevertheless, to the best of our knowledge, there are no techniques for building classifiers based on negative sequential patterns like $A \rightarrow \neg B$, $\neg A \rightarrow B$ and $\neg A \rightarrow \neg B$, where A and B and positive sequential patterns.

To tackle the above problems, based on the work on *negative sequential patterns* (Zhao, Zhang, Cao, Zhang & Bohlscheid 2008), we have designed a new technique, *sequence classifiers using both positive and negative patterns*, to find the relationship between activity sequences and debt occurrences, and also find the impact of oncoming activities on the debt occurrence. The contributions of this chapter are:

- A new technique of *sequence classification using both positive and negative sequential patterns*;
- An application in social security, demonstrating the effectiveness of our new technique on sequence classification with both positive and negative sequential patterns.

4.2 Sequence Classification Using Both Positive and Negative Sequential Patterns

From data mining perspective, sequence classification is to build classifiers using sequential patterns. To the best of our knowledge, all of the existing sequence classification algorithms use positive sequential patterns only. However, the sequential patterns negatively related to debt occurrence are very important in debt detection as well. In this section, we first introduce negative sequential patterns and then propose a novel technique for sequence classification using both negative and positive sequential patterns.

4.2.1 Negative Sequential Patterns

Traditional sequential pattern mining deals with positive correlation between sequential patterns only, without considering negative relationship between them. To find the above negative relationship in sequences, Zhao *et al.* previously designed a notion of *negative sequential rules* (Zhao et al. 2008) as follows.

Definition 4.1 (Negative Sequential Rule). *A negative sequential rule (NSR) is in the form of* $A \rightarrow \neg B$, $\neg A \rightarrow B$ *or* $\neg A \rightarrow \neg B$, *where* A *and* B *are positive sequential patterns composed of items in time order.*

Based on the above definition, there are four types of sequential rules, including the tradition positive sequential rules (see Type I).

- Type I: $A \rightarrow B$, which means that pattern A is followed by pattern B;
- Type II: $A \rightarrow \neg B$, which means that pattern A is not followed by pattern B;
- Type III: $\neg A \rightarrow B$, which means that if pattern A does not appear, then pattern B will occur; and
- Type IV: $\neg A \rightarrow \neg B$, which means that if pattern A doesn't appear, then pattern B will not occur.

For types III and IV whose left sides are the negation of a sequence, the meaning of the rules are: if A doesn't occur in a sequence, then B will (type III) or will not (type IV) occur in the sequence. That is to say, there is no time order between the left side and the right side. Note that A and B themselves are sequential patterns, which makes them different from negative association rules. The supports, confidences and lifts of the four types of sequential rules are shown in Table 4.1. In the table, $P(A\&B)$ denotes the probability of the concurrence of A and B in a sequence, no matter which one occurs first, or whether they are interwoven with each other.

Table 4.1: Supports, Confidences and Lifts of Four Types of Sequential Rules

	Rules	Support	Confidence	Lift
I	$A \rightarrow B$	$P(AB)$	$\frac{P(AB)}{P(A)}$	$\frac{P(AB)}{P(A)P(B)}$
II	$A \rightarrow \neg B$	$P(A)-P(AB)$	$\frac{P(A)-P(AB)}{P(A)}$	$\frac{P(A)-P(AB)}{P(A)(1-P(B))}$
III	$\neg A \rightarrow B$	$P(B)-P(A\&B)$	$\frac{P(B)-P(A\&B)}{1-P(A)}$	$\frac{P(B)-P(A\&B)}{P(B)(1-P(A))}$
IV	$\neg A \rightarrow \neg B$	$1-P(A)-P(B)+P(A\&B)$	$\frac{1-P(A)-P(B)+P(A\&B)}{1-P(A)}$	$\frac{1-P(A)-P(B)+P(A\&B)}{(1-P(A))(1-P(B))}$

4.2.2 An Algorithm for Building Sequence Classifiers

Our algorithm for building a sequence classifier with both positive and negative sequential patterns is composed of five steps.

1) Obtaining negative and positive sequential patterns from a sequential pattern mining algorithm, such as the technique (Zhao et al. 2008).

2) Calculating the frequency, chi-square and CCR of every classifiable sequential pattern, and only those patterns meeting *support*, *significance* (measured by chi-square) and *CCR* criteria are extracted into the classifiable sequential pattern set $\mathcal{P}$.

3) Pruning patterns in the obtained classifiable sequential pattern set with the pattern pruning algorithm in (Li, Han & Pei 2001). The only difference is that, in our algorithm, *CCR*, instead of confidence, is used as the measure for pruning.

4) Conducting serial coverage test by following the ideas in (Liu, Hsu & Ma 1998) and (Li et al. 2001). The patterns which can correctly cover one or more training samples in the serial coverage test are kept for building a sequence classifier.

5) Ranking selected patterns with W_s and building the classifier as follows. Given one sequence instance s, all the classifiable sequential patterns having covered s are extracted. The sum of the weighted score corresponding to each target class is computed and then s is assigned with the class label corresponding to the largest sum.

The Algorithm 2 is illustrated as below. The complexity of the algorithm is $O(mn)$ where m is the sample size chosen and n is the training sample size.

4.3 Experimental Results

Our technique was applied in social security to study the relationship between transactional activity patterns and debt occurrences, and then to build sequence classifiers for the detection of debt occurrences.

The data we used are the debts and activity transactions of 10,069 Centrelink customers from July 2007 to February 2008, which is the same dataset used in Chapter 3.

4.3.1 Results of Negative Sequential Pattern Mining

The technique on *negative sequential rules* (Zhao et al. 2008) was used to find both positive and negative sequential patterns from the above data. By setting minimum support to 0.05, that is, 797 out of 15,931 sequences, 2,173,691 patterns were generated and the longest pattern has 16 activities. From the patterns, 3,233,871 positive and negative rules were derived. Some selected sequential rules are given in Table 4.2. The rules marked with "Type I" are positive sequential rules, while others are negative ones. "DEB" standing for debt and the other codes are activities.

4.3.2 Evaluation of Sequence Classification

This experiment is to test the performance of the classifiers using both positive and negative sequential patterns and compare it with the classifiers built with positive patterns only.

Note that in the binary classification problem in our case study, $A \rightarrow \neg DEB$ can be taken as a positive rule $A \rightarrow c_2$, where c_2 denotes "no debt". Therefore, in this experiment, we treat Type I and Type II patterns as posi-

tive ones and Type III and Type IV as negative ones. That is, in the results shown in Tables 4.4–4.7, the traditional classifiers (labelled as "Positive") were built with both Type I and II rules, while our new classifiers (labelled as "Neg& Pos") were built with all four types of rules. However, in applications where there are multiple classes, Type II rules are negative rules.

By setting the minimum support to 0.05 and 0.1 respectively, we got two sets of sequential patterns, "PS05" and "PS10", The numbers of the four types of patterns are shown in Table 4.3. There are $775,175$ patterns in "PS10" and $3,233,871$ patterns in "PS05". It is prohibitively time consuming to do coverage test and build classifiers on so large sets of patterns. In this experiment, we ranked the patterns according to their CCRs and extracted the top $4,000$ and $8,000$ patterns from "PS05" and "PS10" and referred to them as "PS05-4K", "PS05-8K", "PS10-4K" and "PS10-8K" respectively.

After that, two groups of classifiers were built. One group are classifiers built with both negative and positive patterns (i.e., all four types of rules), and the other are built with positive patterns (i.e., Type I and II rules) only. In order to compare the two groups of classifiers, we select various numbers of patterns used in the final classifiers from the patterns passing coverage test, and the results are shown in Tables 4.4–4.7. In the four tables, the first rows show the number of patterns used in the classifiers. In Tables 4.6 and 4.7, some results are not available for pattern number as 200, because there are less than 200 patterns remaining after coverage test.

From the four tables, we can see that, if built with the same number of rules, in terms of recall, our classifiers built with both positive and negative rules outperforms traditional classifiers with only positive rules under most conditions. It means that, with negative rules involved, our classifiers can predict more debt occurrences.

For the results on "PS05-4K" (see Table 4.4), our classifiers is superior than traditional classifiers with 80, 100 and 150 rules in recall, accuracy and precision.

From the results on "PS05-8K" shown in Table 4.5, we can see that,

our classifiers with both positive and negatives rules outperforms traditional classifiers with only positive rules in accuracy, recall and precision in most of our experiments.

As shown by Tables 4.6 and 4.7, our classifiers have higher recall with 80, 100 and 150 rules. Moreover, our best classifier is the one with 60 rules, where accuracy=0.760, specificity=0.907 and precision=0.514. In both Table 4.6 and Table 4.7, our classifier with 60 rules achieved higher values in the three measures than all the traditional classifiers.

One interesting thing we found is that, the number of negative patterns used for building our classifiers is very small, compared with that of positive patterns. Especially for "PS05-4K" and "PS05-8K", the two pattern sets chosen from the mined patterns with minimum support=0.05, there are only 4 and 7 negative patterns, respectively. However, these several negative patterns do make a difference when building classifiers. Three of them are given as follows.

- $\neg ADV \rightarrow \neg DEB$ (CCR=1.99, conf=0.85)
- $\neg(STM, REA, DOC) \rightarrow \neg DEB$ (CCR=1.86, conf=0.84)
- $\neg(RPR, DOC) \rightarrow \neg DEB$ (CCR=1.71, conf=0.83)

Three examples of positive rules are also given as follows.

- $STM, RPR, REA, EAD \rightarrow DEB$ (CCR=18.1)
- $REA, CCO, EAD \rightarrow DEB$ (CCR=17.8)
- $CCO, MND \rightarrow \neg DEB$ (CCR=2.38)

4.4 Conclusions and Discussion

We have presented a new technique for building sequence classifiers with both positive and negative sequential patterns. We have also presented an

application for debt detection in the domain of social security, which shows the effectiveness of the proposed technique.

One limitation is that an element in a sequence is assumed to be a single event, based on the transaction data in the social security application. However, in many other applications, an element may be composed of multiple items. Therefore, to extend our techniques to such general sequence data will be part of our future work.

Another limitation is that time constraints are only partly involved in our techniques. What we have done is to set the time window for a pattern to be less than 4 months, based on domain experts' suggestions. Nevertheless, we haven't set any other time constraints, such as the time interval between adjacent elements. In other applications, it may be interesting to find patterns with the above constraints and use them to build sequence classifiers.

A third limitation is that, in real world applications, there are different costs associated with correct prediction, false positives and false negatives, and it will be more fair and more useful when measuring the performance of classifiers by taking different costs into consideration.

Data: Sequence Set S

Training sample set TS

Result: Classifiers CSP_i, $i = 0, 1, 2, ...$

begin

Get positive sequential patterns from S with sequential pattern mining algorithm from (Zhao et al. 2008), as P_1, P_2,...,P_n; Get negative sequential patterns from S with sequential pattern mining algorithm from (Zhao et al. 2008), as N_1, N_2,...,N_m;

$i = 1$

while i **do**

if P_i) $>T_{sup}$ and Chi-squareP_i $>T_{chi}$ and CCR(P_i) $>T_{CCR}$ then output P_i to pool C; $i = i + 1$

end

$i = 1$

while i **do**

if P(N_i) $>T_{sup}$ and Chi-squareN_i $>T_(chi)$ and CCR(N_i) $>T_{CCR}$ then output N_i to pool C; $i = i + 1$

end

Sort pool C by CCR in descending order

For each item in C

set its cover-count to 0;

while TS is not empty and C is not empty do

if c from C covers t from TS then

output c

increase $t's$ cover-count by $t's$ CCR

if $t's$ cover-count $>T_\theta$ then delete t

end

Rank the selected patterns by their weighted score. This is CSP

Given one sequence instance s, all the classifiable sequential patterns from CSP that can covered s are extracted. The sum of the weighted score corresponding to each target class is computed and then s is assigned with the class label corresponding to the largest sum.

end

Algorithm 2: Sequence Classifiers with Both Positive and Negative Patterns.

Table 4.2: Selected Positive and Negative Sequential Rules

Type	Rule	Support	Confidence	Lift
I	REA ADV ADV→DEB	0.103	0.53	2.02
I	DOC DOC REA REA ANO→DEB	0.101	0.33	1.28
I	RPR ANO→DEB	0.111	0.33	1.25
I	RPR STM STM RPR→DEB	0.137	0.32	1.22
I	MCV→DEB	0.104	0.31	1.19
I	ANO→DEB	0.139	0.31	1.19
I	STM PYI→DEB	0.106	0.30	1.16
II	STM PYR RPR REA RPT→ ¬DEB	0.166	0.86	1.16
II	MND→ ¬DEB	0.116	0.85	1.15
II	STM PYR RPR DOC RPT→ ¬DEB	0.120	0.84	1.14
II	STM PYR RPR REA PLN→ ¬DEB	0.132	0.84	1.14
II	REA PYR RPR RPT→ ¬DEB	0.176	0.84	1.14
II	REA DOC REA CPI→ ¬DEB	0.083	0.83	1.12
II	REA CRT DLY→ ¬DEB	0.091	0.83	1.12
II	REA CPI→ ¬DEB	0.109	0.83	1.12
III	¬{PYR RPR REA STM}→DEB	0.169	0.33	1.26
III	¬{PYR CCO}→DEB	0.165	0.32	1.24
III	¬{STM RPR REA RPT}→DEB	0.184	0.29	1.13
III	¬{RPT RPR REA RPT}→DEB	0.213	0.29	1.12
III	¬{CCO RPT}→DEB	0.171	0.29	1.11
III	¬{CCO PLN}→DEB	0.187	0.28	1.09
III	¬{PLN RPT}→DEB	0.212	0.28	1.08
IV	¬{ADV REA ADV}→ ¬DEB	0.648	0.80	1.08
IV	¬{STM EAN}→ ¬DEB	0.651	0.79	1.07
IV	¬{REA EAN}→ ¬DEB	0.650	0.79	1.07
IV	¬{DOC FRV}→ ¬DEB	0.677	0.78	1.06
IV	¬{DOC DOC STM EAN}→ ¬DEB	0.673	0.78	1.06
IV	¬{CCO EAN}→ ¬DEB	0.681	0.78	1.05

Table 4.3: The Pattern Numbers of Each Type in PS10 and PS05

	PS10 ($min\_sup = 0.1$)		PS05 ($min\_sup = 0.05$)	
	Number	Percent(%)	Number	Percent(%)
Type I	93,382	12.05	127,174	3.93
Type II	45,821	5.91	942,498	29.14
Type III	79,481	10.25	1,317,588	40.74
Type IV	556,491	71.79	846,611	26.18
Total	775,175	100	3,233,871	100

Table 4.4: The Classifiers on Various Number of Patterns (PS05-4K)

Pattern Number		40	60	80	100	150	200	300
Neg&Pos	Recall	.438	.416	.286	.281	.422	.492	.659
	Precision	.340	.352	.505	.520	.503	.474	.433
	Accuracy	.655	.670	.757	.761	.757	.742	.705
	Specificity	.726	.752	.909	.916	.865	.823	.720
Positive	Recall	.130	.124	.141	.135	.151	.400	.605
	Precision	.533	.523	.546	.472	.491	.490	.483
	Accuracy	.760	.758	.749	.752	.754	.752	.745
	Specificity	.963	.963	.946	.951	.949	.865	.790

Table 4.5: The Classifiers on Various Number of Patterns (PS05-8K)

Pattern Number		40	60	80	100	150	200	300
Neg&Pos	Recall	.168	.162	.205	.162	.173	.341	.557
	Precision	.620	.652	.603	.625	.615	.568	.512
	Accuracy	.771	.774	.773	.771	.771	.775	.762
	Specificity	.967	.972	.956	.969	.965	.916	.829
Positive	Recall	.141	.103	.092	.092	.108	.130	.314
	Precision	.542	.576	.548	.548	.488	.480	.513
	Accuracy	.761	.762	.760	.760	.754	.753	.760
	Specificity	.962	.976	.976	.976	.963	.955	.904

Table 4.6: Classification Results with Pattern Set (PS10-4K)

Pattern Number		40	60	80	100	150
Neg&Pos	Recall	0	.303	.465	.535	.584
	Precision	0	.514	.360	.352	.362
	Accurary	.756	.760	.667	.646	.647
	Specificity	1	.907	.733	.682	.668
Positive	Recall	.373	.319	.254	.216	.319
	Precision	.451	.421	.435	.430	.492
	Accuracy	.736	.727	.737	.738	.753
	Specificity	.853	.858	.893	.907	.893

Table 4.7: The Classifiers on Various Number of Patterns (PS10-8K)

Pattern Number		40	60	80	100	150	200
Neg&Pos	Recall	0	.303	.465	.535	.584	N/A
	Precision	0	.514	.360	.352	.362	N/A
	Accuracy	.756	.760	.667	.646	.647	N/A
	Specificity	1	.907	.733	.682	.668	N/A
Positive	Recall	.459	.427	.400	.378	.281	.373
	Precision	.385	.397	.430	.438	.464	.500
	Accuracy	.688	.701	.724	.729	.745	.756
	Specificity	.762	.790	.829	.843	.895	.879

Chapter 5

Optimising Debt Recovery with Decision Trees

5.1 Introduction

In Centrelink, once a debt is raised, a recovery procedure begins by sending a notification letter to the customer. If the customer does not respond within a defined period, a debt recovery team will attempt to contact the customer by phone to discuss debt repayment arrangements. Given that customer selection is largely random and that the human resources of the debt recovery teams are limited, this procedure is considered to be highly inefficient. The fact that Centrelink's debt base is increasing rather than decreasing, seems to support this view.

In this chapter, the application of data mining techniques to optimise customer selection for debt recovery in Centrelink is presented. Decision trees are built to model debt recovery based on customer characteristics and historical repayment data. The objective is to predict the response of customers contacted by phone for the purpose of putting in place debt repayment arrangements. By ranking customers based on predicted scores, the results will greatly assist Centrelink's resource-limited debt recovery teams. Based on historical data, the results show that if the top 20 percent of customers are

contacted, approximately 50 percent of recoveries will be achieved.

Some previous works have also focused on debts in social security (Zhao, Cao, Morrow, Ou, Ni & Zhang 2006) (Wu, Zhao, Zhang, Zhang, Cao & Bohlscheid 2009) (Zhao, Zhang, Cao, Zhang & Bohlscheid 2009), but those works have mainly concerned debt detection and debt prevention, rather than debt recovery. However, in one particular piece of earlier research, a number of combined association rules for debt recovery (Zhao, Zhang, Figueiredo, Cao & Zhang 2007) is presented. During that research-oriented project, combined association rules are built with demographic patterns and repayment patterns. Association rules with identical demographic patterns and different repayment patterns are discovered, which are used to assist in moving customers from a slow payback arrangement to a quicker repayment schedule. The principal differences between that earlier work and our research are:

- The research objectives are different. The previous work focuses on moving customers from a slow-payer group to a quick-payer group, while our research identifies customers with a higher potential to enter into a successful debt repayment arrangement, thereby maximising debt recovery through out-bound telephone calls.

- The customer populations are different. The previous work involves all Centrelink customers with debt, while our research is concerned only with customers who have had debt for a certain time frame while without a recovery arrangement in place.

- The techniques are different. This project builds decision trees on customer demographics and historical debts and repayments, while the previous research studies hybrid rules by combining frequent patterns from transactional data and demographic data.

5.2 Population and Data Preprocessing

This project targets individuals who are non-current Centrelink customers; have a total debt balance of 1500 to 5000 dollars; have no debt repayment arrangement in place; and have any debts that are at least 44 days overdue. Exclusions from the project population are organisations, deceased customers, Indigenous Australians and those customers whose files are classified as "deny access". In this project, all data are extracted from Teradata Enterprise Data Warehouse (EDW). The data used are demographics, assets and income, benefit payment, debt details, debt recovery history including arrangements and repayments from 1 April 2009 to 31 December 2009. We assume that the first three months of 2010 is a response period following out-bound phone calls by debt recovery teams, and the debt recovery history in that period is used to work out how much repayment made by customers. There are some repayments initiated by customers, such as repayments made by B-pay or cash, which are tagged as voluntary repayments. Other repayments, such as repayments via an agent, are in a great degree from out-of-control customers and are tagged as non-voluntary repayments. Voluntary repayments show the intention of individuals to make arrangements and/or repayments and are used in modelling to rank customers. Similarly, debt recovery arrangements also fall into two categories which distinguish the willingness of customers to repay their debts. After the extraction from EDW, the data are preprocessed with SAS Enterprise Guide and then used to build decision trees using R[1]. The C4.5 algorithm is used in building the decision tree. The current and historical information of customers are aggregated so that each customer has only one record to feed to decision trees. For example, the historical debts of a customer are aggregated into the amount and the number of debts in the past three years and in the last 12 months, the maximum amount of debts, the median amount of debts and so on. The historical repayments are also aggregated as the amount and the number of

[1]A free software environment for statistical computing and graphics. http: //www.r-project.org/

repayments in the past one or three years and the maximum/median amount of repayments. After preprocessing, there are 20,708 customer records, each having 192 variables.

5.3 A Predictive Model for Customer Ranking

The general aim of the analysis is to use the information known about customers "before they fell into the population scope of this project" to predict their debt recovery behaviours in the coming months. The target variable, that is, the variable to predict, is the total amount of voluntary repayments initiated by customers, rather than the amount automatically set by Centrelink or collected by Mercantile Agents (who are debt collection agents contracted by Centrelink). The technique of decision trees has been used to build predictive models and several decision trees with a variation of parameters are built. The one with the best performance on test dataset is chosen and the selected decision tree is illustrated in Figure 5.1. The customers with variables leading them to a leaf node fall into a level of predicted target value. The box-plot under each node illustrates the distribution of repayments for the leaf node. For example, consider the right-most path (i.e. node 1 $\rightarrow$ node 29 $\rightarrow$ node 41) in Figure 5.1. The observations in the training dataset are split into two groups according to whether or not the customer's current outstanding debts have ever been referred to a Mercantile Agent. The customers whose current outstanding debts have never been referred to a Mercantile Agent go to the left branch, while all the other customers go to the right. For the customers falling to the right sub-tree, if the total amount of debts they have repaid in a voluntary manner in the last 12 months is more than 250 dollars, they have an average repayment of 198 dollars in the following 12 weeks.

Based on the decision tree, those variables which make a significant difference with regard to a customers' intention to make repayments are:

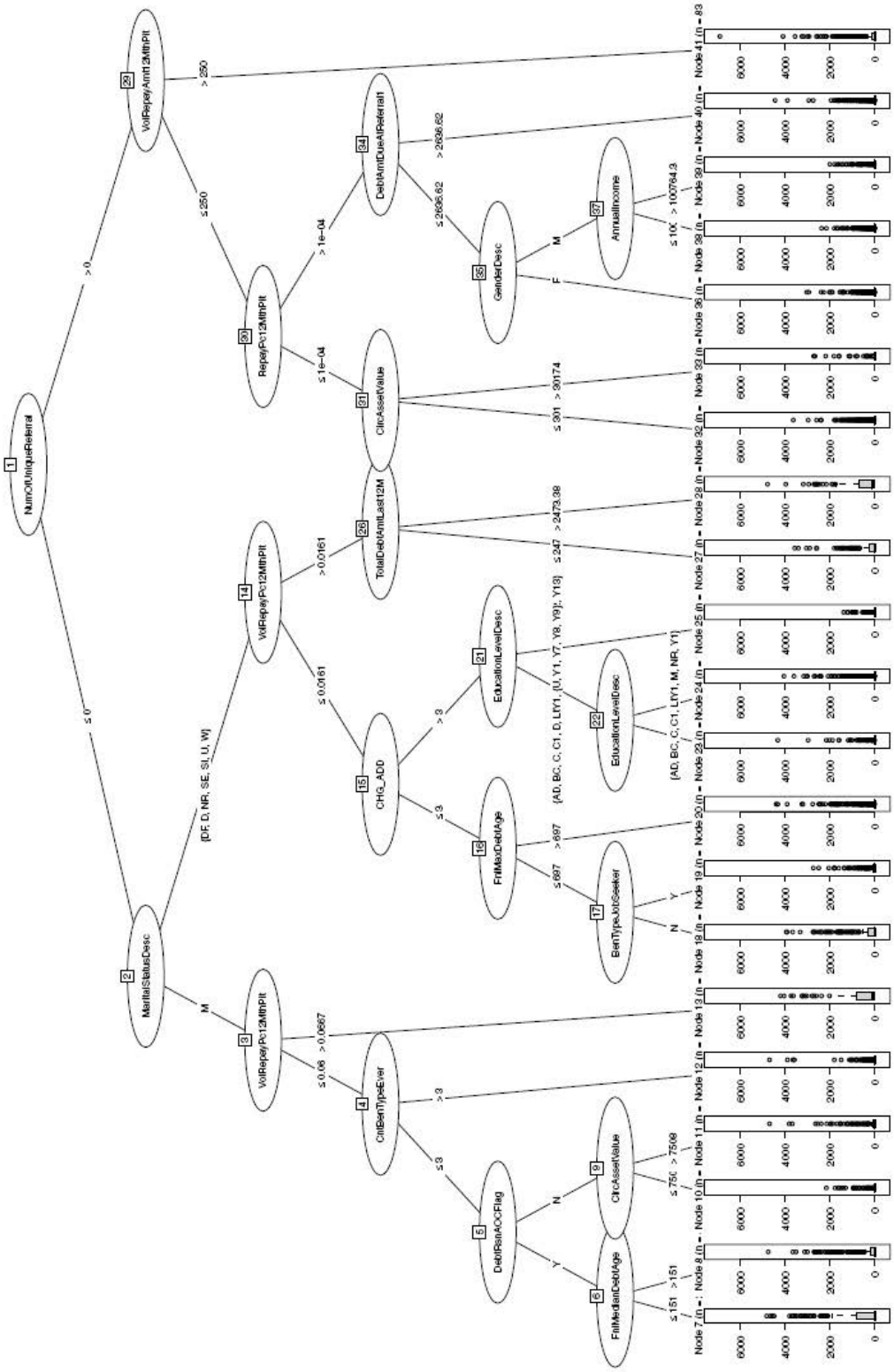

Figure 5.1: Decision Tree

- whether the customer has ever been referred to a Mercantile Agent;
- marital status;
- the amount of voluntary repayments in the past 12 months;
- the proportion of voluntary/total repayments in the customers total debt amount;
- the number of benefit types the customer has ever received;
- the times of address change;
- total debt amount in the past 12 months;
- asset value;
- the amount of debt due at the first referral;
- whether the customer has ever had a debt with debt reason "AOC";
- the age of the longest finalised debt, and the median of the age of all finalised debts;
- education level;
- gender;
- whether the customer is on jobseeker benefit; and
- the customers annual income.

The model is currently being used to generate a list of customers across all payment types, ranked according to the amount of repayments they are likely to make if a repayment arrangement is put in place. The list provides the basis for a 12-week out-bound call trial and a fresh list will be generated each week during the trial period. Ranking customers will enable debt recovery teams to focus on those customers who are likely to repay debts, rather than

the current process which contacts customers at random. The model will increase the amount of debt under arrangement, which in turn will reduce the outstanding debt base and the average age of debts outstanding.

5.4 Model Evaluation

5.4.1 Evaluation Criteria

Since the output of the decision tree is used by Centrelink debt recovery teams to select customer to contact by phone, the repayments collected over a 12-week period are used to evaluate the performance of the model. Results are presented similar to Figure 5.2, where the horizontal axis represents the percentage of (ranked) customers contacted and the vertical axis shows the percentage of repayments that could be collected over 12 weeks. A model is expected to recover more debts with fewer phone calls. Therefore, the closer to the up-left corner is a curve, the better is a model.

5.4.2 Comparison with Commercial Software

Using the default settings, a decision tree is built using a commercial software (referred to as EM). Again using the same settings, that is, MinSplit (the minimum number of instances in a node in order to be considered for splitting), MinBusket (the minimum number of instances in a terminal node), MaxSurrogate (the number of surrogate splits to evaluate) and MaxDepth (the maximum depth of the tree), a second decision tree is built using R, with *ctree* function in package party (Hothorn, Hornik & Zeileis 2006). Each model s run ten times and average results are shown in Figure 5.2 – the solid line shows the results achieved for R, while the dotted line is results for EM. Although results are similar, the model built with R was better on each occasion, expect when the phone call percentile is in 0%–3% and 20%–25%. Therefore *ctree* from R has been chosen to build the models.

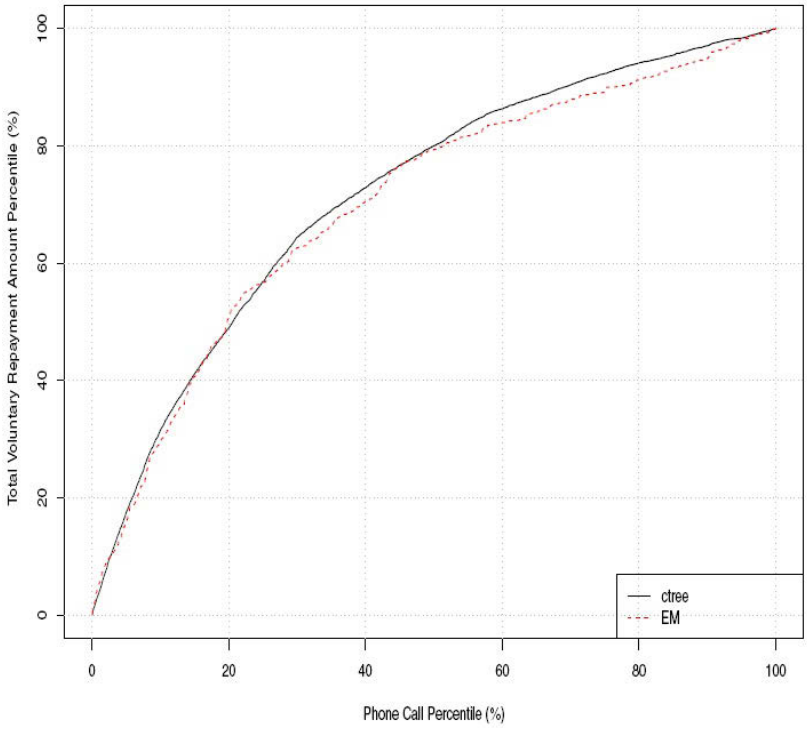

Figure 5.2: Comparison with A Commercial Software (EM)

5.4.3 Experimental Results on Different Parameters

The decision trees generated with *ctree* with six sets of different parameters are also tested. The average results of running each setting 10 times are given in Figure 5.3. The labels in the legend show the values of MinSplit, MinBusket, MaxSurrogate and MaxDepth used in the six sets of parameters. For example, with the first setting "100-40-4-4", MinSplit is set to 100, MinBusket is 40, and both MaxSurrogate and MaxDepth are 4. The six settings are "100-40-4-4", "100-40-10-6", "100-40-20-8", "200-100-15-8", "300-150-15-8" and "300-150-15-6". Figure 5.3 shows that results with depth 6 are better than depth 4. However, there is little difference in results between trees with depth 6 and 8. There is also very little difference in results between trees with "MinSplit=100, MinBusket=40", "MinSplit=200, MinBusket=100" and "MinSplit=300, MinBusket=150". Generally speaking, the smaller the MinSplit and MinBusket, and the greater the MaxDepth, the

more likely to be over-fit is a resulting decision tree. Therefore, "300-150-15-6" is chosen as the setting on which to build our final model, which produces very good results and is less likely to over-fit, compared with models with smaller node sizes and bigger trees.

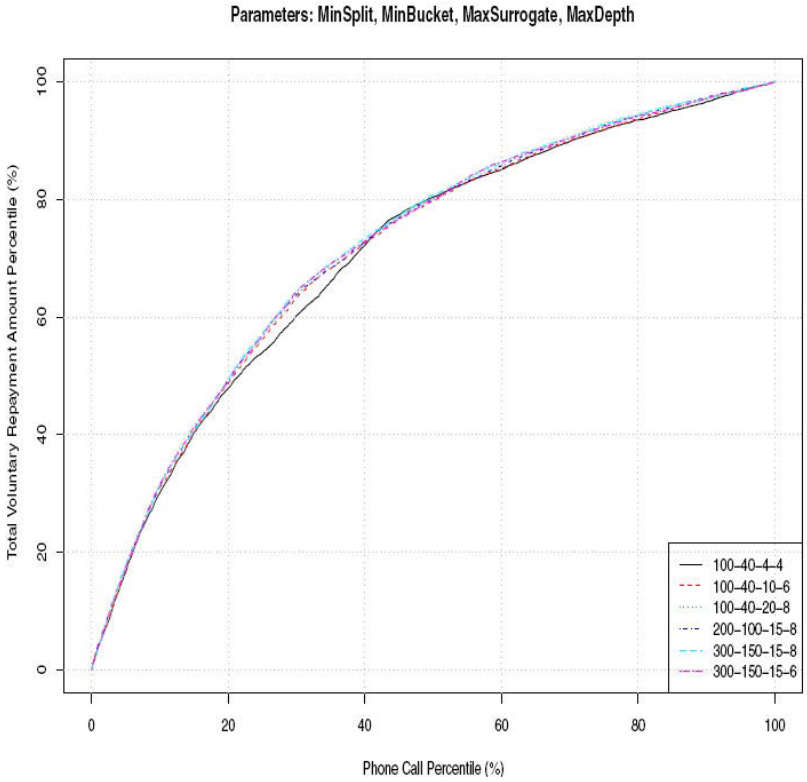

Figure 5.3: Comparison on Different Parameters

5.4.4 Evaluation Results

The decision tree model has been evaluated on historical data. In the modelling procedure, the historical data are randomly partitioned into two sets: training data (70%) and test data (30%). In order to test the stability of the modelling methodology, random partitioning is performed ten times and the model built with training data in each run is evaluated on the corresponding

test data. With the decision tree model, the customers are ranked in descending order based on the predicted amount of voluntary repayments they would repay over a 12-week period. Once ranked, phone calls are made to customers starting from the highest ranking. In Figure 5.4, the black solid line illustrates the average performance of all ten runs, while the other lines are the performance of individual runs. The fact that there is little variation over 10 runs indicates that the performance of the decision tree modelling is stable.

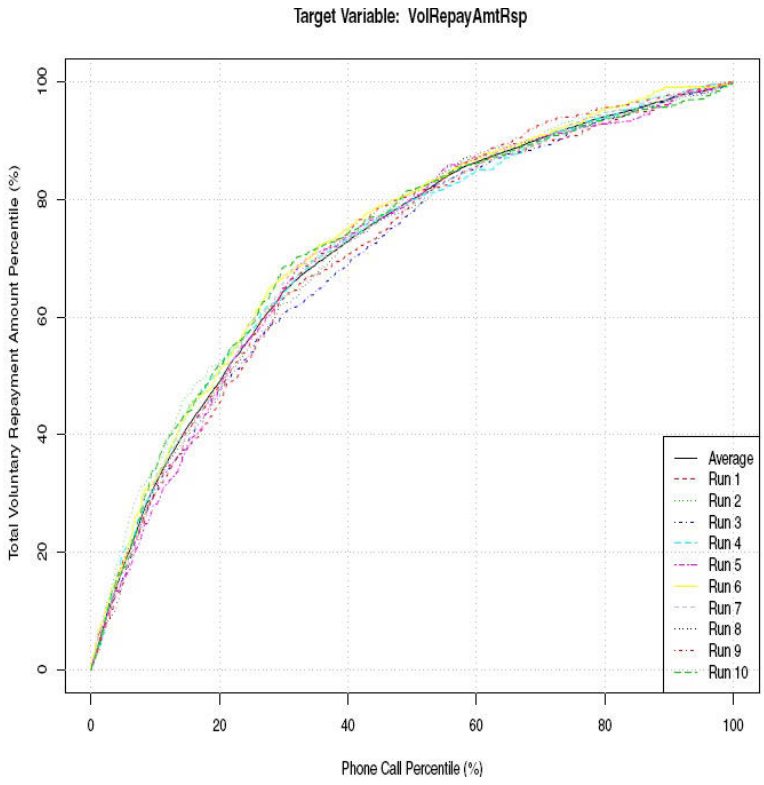

Figure 5.4: Total Voluntary Amount Collected in 12 Weeks (10 runs)

Figure 5.5 shows average result of the above ten runs, where the solid line shows the percentage of repayments collected and the dotted line shows the average voluntary repayment amount repaid by the customers contacted. In Figure 5.5, we see that contacting the highest ranked 20% of customers will result in the collection of approximately 50% of total voluntary repayments

over a 12-week period. This is 2.5 times as good as random customer selection. If 50% customers are contacted, almost 80% of the total voluntary repayment could be collected over the same period. The average voluntary repayment amount per customer contacted increases, in the left of the chart, and then decreases when more customers are contacted. Therefore, the model is effective in capturing in its top-ranked list the customers who would make more repayments.

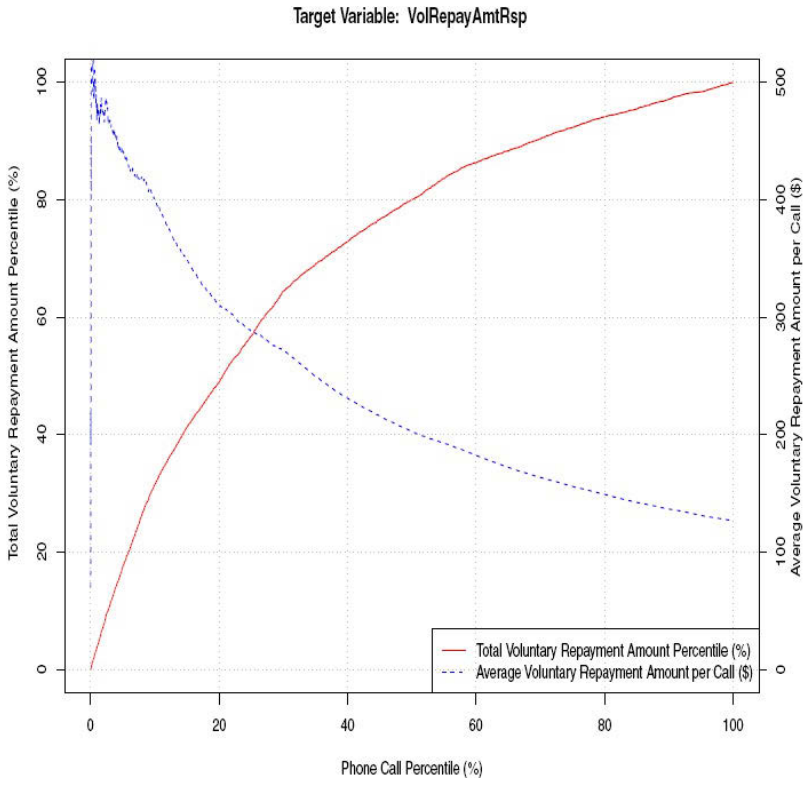

Figure 5.5: Total Voluntary Amount Collected in 12 weeks & Average Voluntary Repayment Amount per Call (Average of 10 runs)

Figure 5.6 shows the total repayment amounts collected (including both voluntary and non-voluntary repayments) of ten runs, while Figure 5.7 shows the average result. Once again, Figure 5.6 validates the stability of the built model, and Figure 5.7 clearly shows that the highest ranked customers would make greater repayments than lower ranked customers.

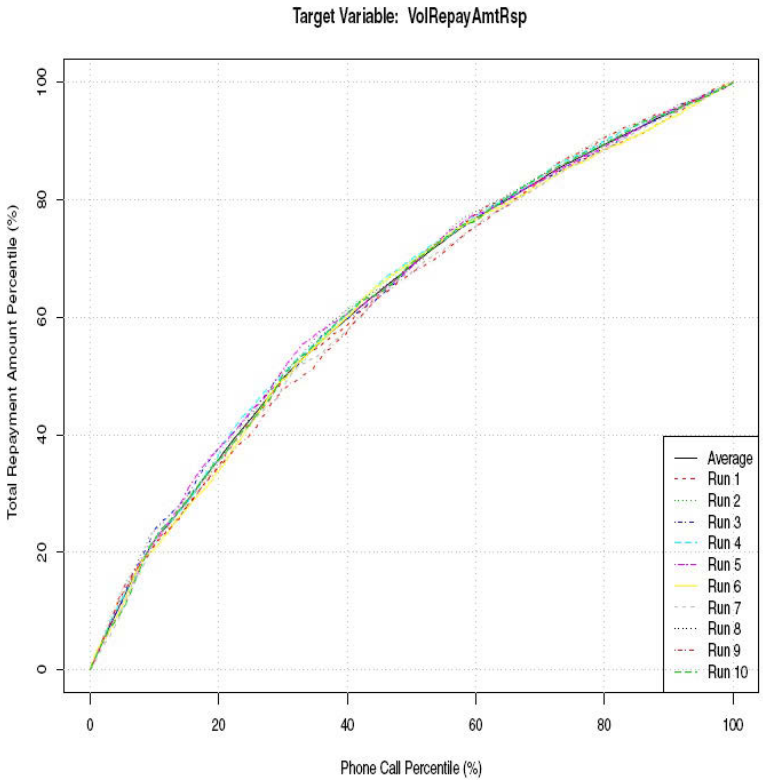

Figure 5.6: Total Repayment Amount Collected in 12 weeks (10 runs)

5.5 Conclusions and Future Work

A decision tree model based on historical data has been developed to improve debt recovery in social security. The test shows that it is very effective to target customers with high intention to repay their debts and recovery debts as much as possible.

The model developed has been applied to real life data to generate a ranking for new customers as they enter the pool. This customer ranking was used in a debt recovery pilot project in Centrelink in 2010. The ranking provided recommendation to determine which customer will be contacted and a fresh list of customers were generated weekly over a 12-week period. Out bound calls were made to the selected customers in the pilot project. The outcome of the pilot was very positive and the debts collected over this

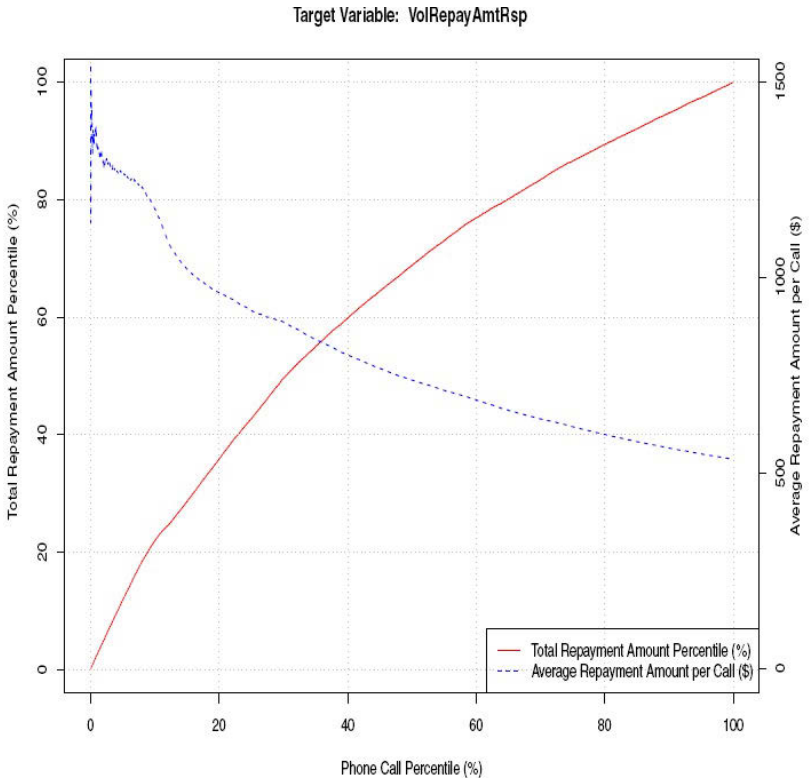

Figure 5.7: Total Repayment Amount Collected in 12 weeks & Average Repayment Amount per Call (Average of 10 Runs)

period outperformed against previous random customer selection for debt recovery. The result will be used to further evaluate the effectiveness of our model and improve the decision tree to make it adaptive to changes and emerging trends.

A further target variable on which to rank customers selected for an outbound phone call is the number of days before a customers first repayment. We have also built decision trees for this variable and it would be interesting to build a predictive model in the future by using two or more target variables at the same time. Another possible future work is to investigate treatment options for customers who are less likely to repay debt by analysing and modelling the difference between high-ranked and low-ranked customers.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Debt detection involves monitoring the behaviour of customers in order to estimate, detect or avoid debt occurrence as quickly as possible once it has been perpetrated. Debt prevention describes measures to stop debt from occurring in a timely fashion. Ideally, debt detection and debt prevention should work together to reduce the debt occurrences. In practice, debt detection must be used continuously, as one will typically be unaware that debt prevention has failed. This thesis applied data mining techniques to help detect and prevent the debt occurrence in social security.

Debt detection is important for improving payment accuracy in social security. Debt detection from customer transaction data can be generally modelled as a sequence classification problem. For long-running debt detections, the patterns in the transaction sequences may exhibit variation from time to time, which makes it imperative to adapt classification to the pattern variation. This thesis proposed a novel *adaptive sequence classification* framework for debt detection in a social security application. The main technique is to capture the pattern variation by boosting discriminative patterns and depressing less discriminative ones according to the latest sequence data. The application of the proposed adaptive sequence classification framework

in a real-world application demonstrates its efficiency and effectiveness.

The existing sequence classification methods based on sequential patterns consider only positive patterns. However, according to our study on a large social security application, negative patterns are very useful in accurate debt detection. In this thesis, a new technique of *sequence classifiers using both positive and negative patterns* was proposed to find the relationship between activity sequences and debt occurrences, and also the impact of oncoming activities on the debt occurrence. An application for debt detection in the domain of social security was conducted, which showed the effectiveness of the proposed technique.

The traditional method of contacting a customer for the purpose of putting in place a debt recovery schedule has been an out-bound phone call. By and large, customers are chosen at random. This obsolete and inefficient method of selecting customers for debt recovery purposes has been existing for years. In order to improve this process, decision trees were built in the thesis to model debt recovery and predict the response of customers if contacted by phone. Test results on historical data showed that, the built model was effective to rank customers in their likelihood of entering into a successful debt recovery repayment schedule. If contacting the top 20 per cent of customers in debt, instead of contacting all of them, approximately 50 per cent of repayments would be received.

6.2 Future work

Although this research has made some progress in debt detection and debt recovery in social security with data mining techniques, there are still many open issues in this area. Our future work will consist of the following parts.

6.2.1 A life-time model for debt detection and prevention

The debt detection and prevention techniques proposed in this thesis aim at capturing customers' behaviour feature in a local fashion. In the continuously long-running terms, behaviours corresponding to a subject is continuous and evolving. Let's take Centrelink's case as an example. Ever since a person becomes a customer of Centrelink, he/she could be qualified to get Austudy Payment or Youth Allowance when undertaking qualified study or apprenticeship. After study, the Newstart Allowance applies in case of unemployment. At this stage, if he/she provides fake information about employment and applies for Newstart Allowance, a fraud is raised. Later, when he/she gets married and has children, he/she could apply Child Care Benefit from Centrelink, and many other family benefits if conditions are satisfied. Suppose he/she gets devoiced unfortunately. Then the Single Parent Payment applies. When he/she is over the age eligible for age pension (e.g. 65), he/she may apply Age Pension from Centrelink. The sequential data in each age group may share some common patterns, while patterns may differ dramatically between age groups. Therefore, it would be valuable to target at a lifetime debt detection and prevention strategy.

For a long sequence containing all the activities in the whole lifetime of a subject, it may span several temporal stages. If the sequence could be divided into several sub-sequences corresponding to the temporal stages, and each sub-sequence is long enough to be analysed as a traditional sequence, then we call the long sequence as a multi-granularity (Bettini, Wang & Jajodia 1998) sequence. Formally, for a sequence $S =< s_{1,1}, s_{1,2}, ..., s_{1,l_1}, s_{2,1}, s_{2,2}, ..., s_{2,l_2}, ..., s_{n,1}, s_{n,2}, ..., s_{n,l_n} >$, if $S_i =< s_{i,1}, s_{i,2}, ..., s_{i,l_i} >$, where $i = 1, 2, ..., n$ is of business significance and l_i is long enough, then we call S as a multi-granularity sequence. In particular, $S =< S_1, S_2, ..., S_n >$ is termed as a macro-granular sequence, and $S_i =< s_{i,1}, s_{i,2}, ..., s_{i,l_i} >$ is called a micro-granular sequence.

Each micro-granular sequence of a multi-granularity sequence is a traditional sequence. For a set of multi-granularity sequences, if they are parti-

tioned according to the same temporal stages, the micro-granular sequences in the same stage must share some common patterns, no matter normal patterns or patterns associated with fraud. Therefore, we can conduct traditional sequence pattern mining and sequence classification techniques over micro-granular sequences in the same time span. This can be referred to as *intra-stage sequence analysis*. According to the results of intra-stage sequence analysis, micro-granular sequences in each temporal stage can be assigned a class label, which represents the subject's behavior in the corresponding stage. Thereafter, we can look at each multi-granularity sequence from a macro-granular point of view, by replacing micro-granular sequences with their class labels in each stage. Since what happened previously in a sequence may have impact on the subject's later behaviour, *inter-stage sequence analysis* could be performed with Finite Automaton. Taking classes in each stage as a state of a subject in corresponding temporal stage, a finite automaton along the whole sequences can be built as in Figure 6.1. With the finite automaton of a set of multi-granularity sequences, we can predict a subject's behaviour in a certain stage, by considering how he behaves in previous stages.

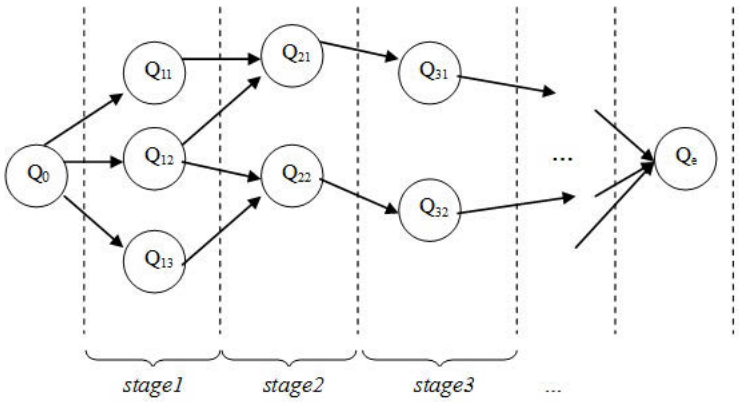

Figure 6.1: FA of a set of multi-granularity sequences

In order to capture the features in each stage, an idea of sequence analysis

across stages is conceived. After intra-stage sequence analysis, each sequential pattern appearing in several stages could be chosen for post-mining across stages. Each pattern's variation of confidence, support, and lift in different stages is to be studied. For those patterns behave relatively stable across stages are regard as common features, while those vary a lot across stages could be used to catch up with the behaviour in different stages.

The challenges in building a lifetime model for debt detection and prevention include 1) How to decide the boundary of temporal stages. Business knowledge may apply. 2) How to give micro-granular sequences in each temporal stage proper class labels.

6.2.2 Recommendation for debt detection and prevention

Recommendation for debt detection and prevention is to recommend some activity in a sequence, either push in some activities or change some activities, so as to reduce the occurrence likelihood of some predicted activity.

The patterns identified in sequential pattern mining could be further studied on their impacts in various situations, to give recommendation for debt detection and debt prevention. Over a sequence dataset, we find patterns like $P \rightarrow T$, $P \| A_1 \rightarrow T$, $P \| A_2 \rightarrow T$ and $P \| A_3 \rightarrow T$, where P is a sequence of activities, A_1, A_2 and A_3 are activities, T denotes for debt occurrence and $\|$ stands for $followed\ by$. If the confidence of $P \| A_1 \rightarrow T$ is greater than the other patterns, then 1) A_1 is good for fraud detection after P, if $P \| A_1$ is designed for detecting fraud; or 2) A_1 should be avoided after P if possible, when $P \| A_1$ is not designed for fraud detection. Also, it is important to get advice from business experts to select the patterns for recommendation on whether and when to conduct an activity for fraud detection or for fraud prevention.

Post-mining (Zhao, Zhao, Zhang & Cao 2009) techniques could be used to find patterns like $P \rightarrow T$, $P \| A_1 \rightarrow T$, $P \| A_2 \rightarrow T$ and $P \| A_3 \rightarrow T$, where a) the confidences of the patterns are significantly different, and b) A_1, A_2

and A_3 are "debt detectors" activities (e.g., a review) that are used to find out the existence of debt.

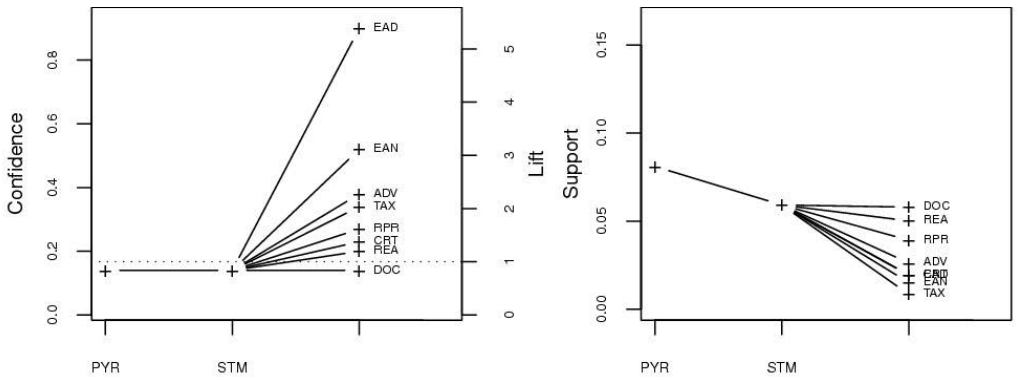

Figure 6.2: The impacts of various activities following "PYR STM", showing that EAD, EAN, ADV and TAX are highly associated with debt occurrences

Based on the activity patterns we generated in Centrelink application, Figure 6.2 shows a set of activity patterns starting with "PYR STM" and the third activities are respectively EAD, EAN, ADV, TAX, RPR, CRT, REA and DOC. In the left chart, the horizontal dotted line shows the average likelihood of debt occurrence when we know nothing about the data. The left vertical axis shows confidence (i.e., the likelihood of debt occurrence) and the right one shows lift (i.e., the ratio of the likelihood of debt occurrence to the average case). The right chart shows the supports of patterns (i.e., the proportion of sequences that contain the pattern). Every "+" shows the corresponding value of the pattern composed of the activities from the first one to the current one. Specifically, with "PYR STM" occurring first, EAD is most likely to be followed by a debt, with a confidence of 0.9 and a lift of 5.4. It means that 90% of all sequences containing "PYR STM EAD" are followed by debt, which is 4.4 times higher than the average case. Its support is 0.02, that is, "PYR STM EAD DEB" appears in 2% sequences. Since "EAD" denotes the activity earnings verification which is always a part of review, it could be recommended to detect the potential debt. As another example,

"DOC" is not designed to detect debt and it maintains the probability of debt occurrence. So "DOC" could be the one to be recommended to avoid debts. For activities that are not designed to detect debt, if their occurrence will increase the probability of debt occurrence, they should be avoided after "PYR STM".

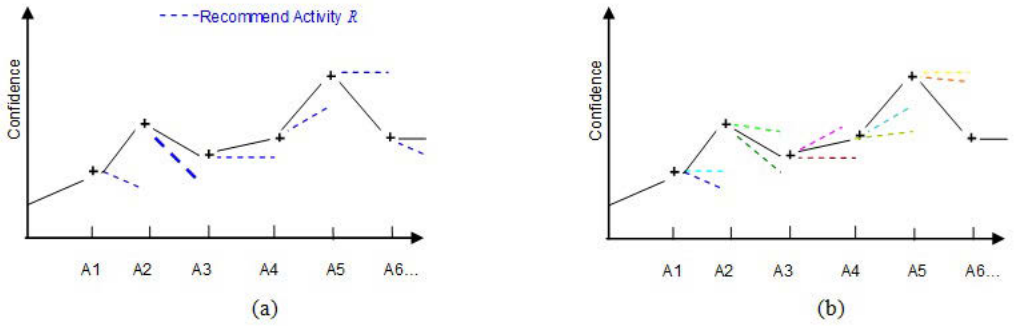

Figure 6.3: Fraud prevention recommendations

A direct and simple way to decide the time point to do recommendation is to observe the probability of fraud occurrence during the sequence matching procedure for fraud detection. When detecting fraud from a sequence, we match the sequence with classifiable sequential patterns. According to the pattern matching result to the present, if the probability of fraud occurrence turns beyond a user specified threshold, then actions should be taken to reduce the probability of fraud occurrence as soon as possible. Firstly, we choose a set of activities that are highly negatively related to fraud as recommendation candidates. For each of the candidate activity, we insert it into the patterns that are positively related to fraud at different positions, so as to find out under what circumstance the candidate could reduce the fraud occurrence probability greatly. And then such activities are included into the patterns at the best recommendation point as a "spreading branch" for future recommendation to prevent fraud. The idea of spreading branch is illustrated by dotted lines in Figure 6.3. From Figure 6.3(a), we observe that, among all the recommendation points, the inclusion of the activity R just after the activity A_2 will reduce the confidence of fraud occurrence

most greatly. Therefore, the recommendation branch of R is added as the bold dashed line. Moreover, we can study the impact of several activities on fraud occurrence at each possible recommendation point, as illustrated in Figure 6.3(b).

The challenging issues about recommendation for fraud detection and prevention are: 1) Decision on the time point to do recommendation; 2) Decision on which activity to be recommended.

Debt discovery and debt prevention in social security area is an important and ongoing topic due to business integrity and productivity reasons. As addressed in the chapter, there are still a lot of challenging issues and future work to discovery. Our study opens a promising future for this area by using data mining techniques.

Appendix A

List of Publications

Below listed include my research papers that have been finished and published during my PhD study at the University of Technology, Sydney:

- Shanshan Wu, Yanchang Zhao, Huaifeng Zhang, Chengqi Zhang, Longbing Cao, Hans Bohlscheid: Debt Detection in Social Security by Adaptive Sequence Classification. KSEM 2009: 192-203
- Yanchang Zhao, Huaifeng Zhang, Shanshan Wu, Jian Pei, Longbing Cao, Chengqi Zhang, Hans Bohlscheid: Debt Detection in Social Security by Sequence Classification Using Both Positive and Negative Patterns. ECML/PKDD (2) 2009: 648-663
- Yanchang Zhao, Hans Bohlscheid, Shanshan Wu, Longbing Cao: Less Effort, More Outcomes: Optimising Debt Recovery with Decision Trees. ICDM Workshops 2010: 655-660

Bibliography

Aggarwal, C. C. (2002), On effective classification of strings with wavelets, *in* 'Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '02, ACM, New York, NY, USA, pp. 163–172.

Agrawal, R. & Srikant, R. (1995), Mining sequential patterns, *in* 'Proc. of the 11th International Conference on Data Engineering', IEEE Computer Society Press, Taipei, Taiwan, pp. 3–14.

Ayres, J., Flannick, J., Gehrke, J. & Yiu, T. (2002), Sequential pattern mining using a bitmap representation, *in* 'KDD'02: Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM, New York, NY, USA, pp. 429–435.

Bannai, H., Hyyro, H., Shinohara, A., Takeda, M., Nakai, K. & Miyano, S. (2004), Finding optimal pairs of patterns, *in* 'Proc. of the 4th Workshop on Algorithms in Bioinformatics (WABI'04)', Bergen, Norway, pp. 450–462.

Bettini, C., Wang, X. S. & Jajodia, S. (1998), 'Mining temporal relationships with multiple granularities in time sequences', *Data Engineering Bulletin* **21**, 32–38.

Bonchi, F., Giannotti, F., Mainetto, G. & Pedreschi, D. (1999), A classification-based methodology for planning audit strategies in fraud detection, *in* 'Proc. of the 5th ACM SIGKDD International Conference

on Knowledge Discovery and Data Mining, ACM Press', San Diego, CA, USA, pp. 175–184.

BY, C., JG, C. & J., K.-S. (2005), 'Protein classification based on text document classification techniques', *Proteins* **1**(58), 855–970.

Centrelink annual report 2007-2008 (2008), Technical report.

Cheng, H., Yan, X., Han, J. & wei Hsu, C. (2007), Discriminative frequent pattern analysis for effective classification, *in* 'In ICDE', pp. 716–725.

Chuzhanova, N. A., Jones, A. J. & Margetts, S. (2010), 'Feature selection for genetic sequence classification', *Bioinformatics* **12**(1), 40–48.

Dong, G. (2009), *Sequence Data Mining*, Springer-Verlag, Berlin, Heidelberg.

Exarchos, T. P., Tsipouras, M. G., Papaloukas, C. & Fotiadis, D. I. (2008), 'A two-stage methodology for sequence classification based on sequential pattern mining and optimization', *Data Knowl. Eng.* **66**(3), 467–487.

Fast, A., Friedland, L., Maier, M., Taylor, B., Jensen, D., Goldberg, H. G. & Komoroske, J. (2007), Relational data pre-processing techniques for improved securities fraud detection, *in* 'Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', ACM Press, San Jose, California, USA, pp. 941–949.

Fawcett, T. & Provost, F. (1997), 'Adaptive fraud detection', *Data Mining and Knowledge Discovery* **1**, 291–316.

Han, J., Cheng, H., Xin, D. & Yan, X. (2007), 'Frequent pattern mining: Current status and future directions', *Data Min. Knowl. Discov.* **15**(1), 55–86.

Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U. & Hsu, M.-C. (2000), Freespan: frequent pattern-projected sequential pattern mining, *in* 'KDD '00: Proc. of the 6th ACM SIGKDD international conference on

Knowledge discovery and data mining', ACM, Boston, Massachusetts, USA, pp. 355–359.

Hothorn, T., Hornik, K. & Zeileis, A. (2006), 'party: A laboratory for recursive partytioning'.

Ji, X., Bailey, J. & Dong, G. (2005), Mining minimal distinguishing subsequence patterns with gap constraints, *in* 'In ICDM', pp. 194–201.

Julisch, K. & Dacier, M. (2002), Mining intrusion detection alarms for actionable knowledge, *in* 'Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', Edmonton, Alberta, Canada, pp. 366–375.

Keogh, E. J. & Pazzani, M. J. (2000), Scaling up dynamic time warping for datamining applications, *in* 'In Proc. 6th Int. Conf. on Knowledge Discovery and Data Mining', pp. 285–289.

Keogh, E. & Kasetty, S. (2003), 'On the need for time series data mining benchmarks: A survey and empirical demonstration', *Data Min. Knowl. Discov.* **7**(4), 349–371.

Kim, S.-B., Han, K.-S., Rim, H.-C. & Myaeng, S. H. (2006), 'Some effective techniques for naive bayes text classification', *IEEE Transactions on Knowledge and Data Engineering* **18**(11), 1457–1466.

Lei, H. & Govindaraju, V. (2005), 'Similarity-driven sequence classification based on support vector machines', *2013 12th International Conference on Document Analysis and Recognition* **0**, 252–261.

Lesh, N., Zaki, M. J. & Ogihara, M. (1999), Mining features for sequence classification, *in* 'Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '99, ACM, New York, NY, USA, pp. 342–346.

Leslie, C., Kuang, R. & Bennett, K. (2004), 'Fast string kernels using inexact matching for protein sequences', *Journal of Machine Learning Research* **5**, 1435–1455.

Leslie, C. S., Eskin, E. & Noble, W. S. (2002), The spectrum kernel: A string kernel for svm protein classification., *in* 'Pacific Symposium on Biocomputing', pp. 566–575.

Li, W., Han, J. & Pei, J. (2001), Cmar: Accurate and efficient classification based on multiple class-association rules, *in* 'ICDM '01: Proc. of the 2001 IEEE International Conference on Data Mining', IEEE Computer Society, Washington, DC, USA, pp. 369–376.

Liang, F., Zhang, J. Z., Tu, D. & Meng, X. (2015), 'Improved frequent pattern mining in spark 1.5: Association rules and sequential patterns'.

Lin, N. P., Chen, H.-J. & Hao, W.-H. (2007), Mining negative sequential patterns, *in* 'Proc. of the 6th WSEAS International Conference on Applied Computer Science', Hangzhou, China, pp. 654–658.

Liu, B., Hsu, W. & Ma, Y. (1998), Integrating classification and association rule mining, *in* 'KDD'98: Proc. of the 4th International Conference on Knowledge Discovery and Data Mining', AAAI Press, pp. 80–86.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. & Watkins, C. (2002), 'Text classification using string kernels', *J. Mach. Learn. Res.* **2**, 419–444.

Ouyang, W.-M. & Huang, Q.-H. (2007), Mining negative sequential patterns in transaction databases, *in* 'Proc. of 2007 International Conference on Machine Learning and Cybernetics', Hong Kong, China, pp. 830–834.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. & Hsu, M.-C. (2001), Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth, *in* 'ICDE '01: Proc. of the 17th International

Conference on Data Engineering', IEEE Computer Society, Washington, DC, USA, pp. 215–224.

Phua, C., Alahakoon, D. & Lee, V. (2004), 'Minority report in fraud detection: classification of skewed data', *SIGKDD Explorations* **6**(1), 50–59.

Raju, V. & Varma, G. (n.d.), 'Mining closed sequential patterns in large sequence databases', *International Journal of Database Management Systems (IJDMS)* **7**(1), 29–39.

Rosset, S., Murad, U., Neumann, E., Idan, Y. & Pinkas, G. (1999), Discovery of fraud rules for telecommunications - challenges and solutions, *in* 'Proc. of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', San Diego, CA, USA, pp. 409–413.

Shelke, S. & Itkar, A. (n.d.), 'A review on sequential pattern mining algorithms', *IInternational Journal of Electrical, Electronics and Computer Engineering* **4**(1), 14–19.

Srikant, R. & Agrawal, R. (1996), Mining sequential patterns: Generalizations and performance improvements, *in* 'EDBT '96: Proc. of the 5th International Conference on Extending Database Technology', Springer-Verlag, London, UK, pp. 3–17.

Sun, X., Orlowska, M. E. & Li, X. (2004), Finding negative event-oriented patterns in long temporal sequences, *in* 'PAKDD'04: Proc. of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining', Sydney, Australia, pp. 212–221.

Tseng, V. S. M. & hui Lee, C. (2005), C.-h.: Cbs: A new classification method by using sequential patterns, *in* 'In: SDM 2005: Proc. of the 2005 SIAM International Data Mining Conference', pp. 596–600.

Verhein, F. & Chawla, S. (2007), Using significant, positively associated and relatively class correlated rules for associative classification of imbal-

anced datasets, *in* 'ICDM'07: Proc. of the 7th IEEE International Conference on Data Mining', pp. 679–684.

Virdhagriswaran, S. & Dakin, G. (2006), Camouflaged fraud detection in domains with complex relationships, *in* 'Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', Philadelphia, PA, USA, pp. 941–947.

Whitrow, C., Hand, D., Juszczak, P., Weston, D. & Adams, N. (2008), 'Transaction aggregation as a strategy for credit card fraud detection', *Data Mining and Knowledge Discovery* .

Wu, C. H., Berry, M. W., Fung, Y.-S. & McLarty, J. (1993), Neural networks for molecular sequence classification., *in* 'ISMB', pp. 429–437.

Wu, S., Zhao, Y., Zhang, H., Zhang, C., Cao, L. & Bohlscheid, H. (2009), Debt detection in social security by adaptive sequence classification, *in* 'Knowledge Science, Engineering and Management, Third International Conference, KSEM 2009, Vienna, Austria, November 25-27, 2009. Proceedings', pp. 192–203.

Xi, X., Keogh, E., Shelton, C., Wei, L. & Ratanamahatana, C. A. (2006), Fast time series classification using numerosity reduction, *in* 'Proceedings of the 23rd International Conference on Machine Learning', ICML '06, ACM, New York, NY, USA, pp. 1033–1040.

Xing, Z., Pei, J. & Keogh, E. (2010), 'A brief survey on sequence classification', *SIGKDD Explor. Newsl.* **12**(1), 40–48.

Yakhnenko, O., Silvescu, A. & Honavar, V. (2005), Discriminatively trained markov model for sequence classification, *in* 'Proceedings of the Fifth IEEE International Conference on Data Mining', ICDM '05, IEEE Computer Society, Washington, DC, USA, pp. 498–505.

Zaki, M. J. (2001), 'Spade: An efficient algorithm for mining frequent sequences', *Machine Learning* **42**(1-2), 31–60.

Zhang, H., Zhao, Y., Cao, L., Zhang, C. & Bohlscheid, H. (2009), 'Customer activity sequence classification for debt prevention in social security.', *J. Comput. Sci. Technol.* **24**(6), 1000–1009.

Zhao, Y., Cao, L., Morrow, Y., Ou, Y., Ni, J. & Zhang, C. (2006), Discovering debtor patterns of centrelink customers, *in* C. Peter, P. J. Kennedy, J. Li, S. J. Simoff & G. J. Williams, eds, 'Fifth Australasian Data Mining Conference (AusDM2006)', Vol. 61 of *CRPIT*, ACS, Sydney, Australia, pp. 135–144.

Zhao, Y., Zhang, H., Cao, L., Zhang, C. & Bohlscheid, H. (2008), Efficient mining of event-oriented negative sequential rules, *in* 'WI'08: Proc. of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence', Sydney, Australia, pp. 336–342.

Zhao, Y., Zhang, H., Cao, L., Zhang, C. & Bohlscheid, H. (2009), Mining both positive and negative impact-oriented sequential rules from transactional data, *in* 'Advances in Knowledge Discovery and Data Mining, 13th Pacific-Asia Conference, PAKDD 2009, Bangkok, Thailand, April 27-30, 2009, Proceedings', pp. 656–663.

Zhao, Y., Zhang, H., Figueiredo, F., Cao, L. & Zhang, C. (2007), Mining for combined association rules on multiple datasets, *in* 'Proceedings of the 2007 International Workshop on Domain Driven Data Mining', DDDM '07, ACM, New York, NY, USA, pp. 18–23.

Zhao, Y., Zhao, Y., Zhang, C. & Cao, L. (2009), *Post-mining of Association Rules: Techniques for Effective Knowledge Extraction*, Information Science Reference - Imprint of: IGI Publishing, Hershey, PA.