Zhang H, Zhao Y, Cao L *et al.* Customer activity sequence classification for debt prevention in social security. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 24(6): 1000–1009 Nov. 2009

Customer Activity Sequence Classification for Debt Prevention in Social Security

Huaifeng Zhang[1] (张淮风), *Member*, *IEEE*, Yanchang Zhao[2], *Member*, *IEEE*
Longbing Cao[2] (操龙兵), *Senior Member*, *IEEE*, Chengqi Zhang[2] (张成奇), *Senior Member*, *IEEE*
and Hans Bohlscheid[1]

[1] *Payment Reviews Branch, Business Integrity Division, Centrelink, Canberra, Australia*

[2] *Centre for Quantum Computation and Intelligent Systems (QCIS), University of Technology, Sydney, Australia*

E-mail: {huaifeng.zhang, hans.bohlscheid}@centrelink.gov.au; {yczhao, lbcao, chengqi}@it.uts.edu.au

Received February 28, 2009; revised July 16, 2009.

Abstract From a data mining perspective, sequence classification is to build a classifier using frequent sequential patterns. However, mining for a complete set of sequential patterns on a large dataset can be extremely time-consuming and the large number of patterns discovered also makes the pattern selection and classifier building very time-consuming. The fact is that, in sequence classification, it is much more important to discover discriminative patterns than a complete pattern set. In this paper, we propose a novel hierarchical algorithm to build sequential classifiers using discriminative sequential patterns. Firstly, we mine for the sequential patterns which are the most strongly correlated to each target class. In this step, an aggressive strategy is employed to select a small set of sequential patterns. Secondly, pattern pruning and serial coverage test are done on the mined patterns. The patterns that pass the serial test are used to build the sub-classifier at the first level of the final classifier. And thirdly, the training samples that cannot be covered are fed back to the sequential pattern mining stage with updated parameters. This process continues until predefined interestingness measure thresholds are reached, or all samples are covered. The patterns generated in each loop form the sub-classifier at each level of the final classifier. Within this framework, the searching space can be reduced dramatically while a good classification performance is achieved. The proposed algorithm is tested in a real-world business application for debt prevention in social security area. The novel sequence classification algorithm shows the effectiveness and efficiency for predicting debt occurrences based on customer activity sequence data.

Keywords sequential pattern mining, sequence classification, coverage test, interestingness measure

1 Introduction

Many real world applications involve data consisting of sequences of elements. In network intrusion detection, there are sequences of TCP/IPA packets. In text mining, the data is composed of sequences of words and delimiters. In customer basket analysis, the items purchased by each customer is made up of sequences of transactions. In bioinformatics, DNA, RNA and protein are all composed of sequences of molecule segments. The classification of sequence data is difficult because of the vast number of features used in describing each example.

Because of a wide range of applications and significant challenges, sequence classification has been studied for several decades. Since the early 1990's, along with the development of pattern recognition, data mining and bioinformatics, many sequence classification models have been proposed. However, there are two drawbacks in existing algorithms. One is that most of them are application-dependent in that they apply specific domain knowledge to build the classifier, for example the sequence classification models on speech recognition[1], text classification[2−3], bioinformatics[4−7], event analysis[8], customer behaviour predictions[9], and so on. Another drawback is that most algorithms cannot work on large datasets because of the complexity of the models.

In social security, each customer's transactional records form an activity sequence. From a business point of view, these sequences are closely correlated to the occurrence of debt. Here, a debt indicates a payment made by government to a customer who was not entitled to that payment. Because of the ongoing debt base over many years, debt prevention is a significant business goal of government departments and agencies.

Regular Paper
This work is supported by Australian Research Council Linkage Project under Grant No. LP0775041 and the Early Career Researcher Grant under Grant No. 2007002448 from University of Technology, Sydney, Australia.

In order to prevent customer debt, one critical task is to build debt predictive models based on the huge volume of customer activity sequences in social security area. For example, in Centrelink, the government agency which delivers a range of services to Australian community, there are more than 6.6 billion transactions filed annually on approximately 6.5 million customers. The conventional model-based sequence classification algorithms are very difficult to extend to process such a huge volume of data.

In recent years, because of the significant progress in data mining, there have been several researchers working on sequence classification[11−14] based on sequential pattern mining techniques[15−19]. These algorithms all follow a two-step strategy. The first step is sequential pattern mining in which a complete sequential pattern set is discovered given a minimum support. The second step is to select the discriminative patterns and build sequential classifiers based on the patterns. In the existing sequence classification algorithms, efficiency is the major bottleneck because of the following two issues. Firstly, sequential pattern mining is still very time-consuming. Suppose we have 150 distinct itemsets. If the aim is to mine for 10-item sequences, the number of candidate sequential patterns is more than

$$150^1 + 150^2 + 150^3 + \cdots + 150^{10} \approx 5 \times 10^{21}.$$

Even with efficient algorithms, the sequential pattern mining in the above example would still take weeks, or even months to complete if all the candidates are generated and processed. Secondly, a number of processes, such as pattern pruning[20−21] and coverage test[20−21], are to be applied to the sequential pattern set to build the sequential classifier. If the sequential pattern set contains a huge number 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 to find the complete rule set, but to discover the most discriminative rules[22−23]. Cheng *et al.*'s experimental results show that "redundant and non-discriminative patterns often overfit the model and deteriorate the classification accuracy"[23].

In this paper, we will not follow the conventional two-step sequence classification strategy. Instead, we propose a novel framework to build sequential classifier. The final classifier consists of a number of subclassifiers organised in a hierarchical way. Each of the sub-classifiers is built as follows. Firstly, in the sequential pattern mining stage, a small set of the sequential patterns strongly correlated to the target classes are discovered. Following pruning, the sequential pattern set is input for serial coverage test[20−21]. The sequential patterns that pass the coverage test form the first level of the sequential classifier. On the other hand, since we only select a small set of sequential patterns which are strongly correlated to the target classes, very often there are some samples not covered by the mined patterns. These samples are fed back for the training in next loop. This process continues until predefined thresholds are reached or all samples are covered. And in each loop, the patterns passed the coverage test are used to build the subclassifier at the corresponding level of the final sequential classifier.

The testing of the classifier is also implemented in a hierarchical way. That is, the sequential patterns of the first level are used for the testing first. If the testing instances cannot be covered by the patterns of the first level, they are tested using the patterns of the second level, and this continues until the last level of the classifier is attained. Following this strategy, at each level, only a small number of discriminative patterns have to be tested for classification. Hence the testing time is also reduced compared to conventional algorithms. Moreover, selecting the most discriminative patterns at each level makes the classifier work well with respect to the classification accuracy.

There are three main contributions in this paper. Firstly, we propose a novel hierarchical algorithm for sequence classification, which can reduce the running time while keeping the performance of the classification. Secondly, we select the most discriminative patterns using statistical test and class correlation ratio (CCR)[24]. And lastly, our algorithm is tested on a real-world application, which shows the efficiency and effectiveness of the proposed algorithm.

The structure of this paper is as follows. Section 2 introduces the previous work related to this paper. Section 3 provides the notation and description of the sequence classification problem. Section 4 introduces the discriminativeness measure in our algorithm. Section 5 is the outline of our algorithm. The case study is in Section 6, which is followed by the conclusion in Section 7.

2 Related Work

2.1 Associative Classification

Associative classification has a close relationship with sequence classification since they both use frequent patterns to build classifiers. In 1998, Liu *et al.*[20] combined association rule mining and classification to propose classification based on associations (CBA) algorithm, in which frequent itemsets instead of conventional features were used to build the classifier. The associative classifier shows good accuracy,

good scalability, and is easy to be interpreted. However, CBA only uses the highest ranking rule corresponding to the features of one instance, which is later proven to have serious overfitting problem. Li *et al.* proposed the algorithm Classification based on Multiple Association Rules (CMAR)[21]. In their algorithm, Chi-square test is used to measure how significantly one rule is positively correlated to a class. In order to tackle the overfitting problem in CBA algorithm, multiple rules instead of a single rule are employed to make decision on each sample. In CMAR, the ranking of the rule set is based on the weighted Chi-square of each rule. Antonie *et al.*[25] proposed a two-stage associative classification system (2SARC). The first stage is similar to that of the algorithms, i.e., mining for the complete set of class association rules. However, in the second stage, it is not to compare the scoring values of each class. Instead, a neural network was trained on a number of parameters of the association rules. The final classification decision is made by the trained neural network. Baralis *et al.*[26] argued that pruning classification rules should be only limited to rules resulting in incorrect classification. They proposed a lazy pruning algorithm which discards rules that incorrectly classify training objects and keep all the others. Normally the associative classification with lazy pruning generates many more rules in the classifier than the algorithms without lazy pruning. Cheng *et al.*[23] analysed the relationship between pattern frequency and its predictive power from information theory perspective. The theoretical analysis in their paper demonstrates that frequent patterns are high quality features and have good model generalisation ability. Their algorithms tackle the *Relevance Problem* and *Redundancy Problem*, which are believed to improve the performance of the classifier.

All of the above associative classification algorithms consist of two relatively independent parts, i.e., pattern mining and classifier modelling. In these algorithms, efficiency is the bottleneck to prevent them from possible deployment in real-world business. In 2005, Wang *et al.*[27] proposed the HARMONY algorithm, which improved the efficiency much, especially when min_sup is not very low. In their algorithm, only the rules with highest confidence covering each sample are kept while the others are pruned. However it still follows the two-stage framework but employs pruning in mining process. When the min_sup is set a low value, the searching space cannot be pruned much. Cheng *et al.*[28] proposed the DDPMine algorithm to directly build classifier from discriminative frequent patterns. The rules corresponding to greatest information gain are kept in mining process. Because the instances covered by the mined rule are eliminated from further process, the total searching space is much smaller than that of the conventional algorithms. In their algorithm, the database keeps varying during the whole mining process. Theoretically, it is not reasonable to equally treat each mined pattern in the classifier.

2.2 Sequence Classification

There have been several researchers working on building sequence classifiers based on frequent sequential patterns. In 1999, Lesh *et al.*[11] 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. 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 that combines classification and sequential pattern mining. However, a huge number of sequential patterns are mined in the sequential mining procedure. Although pruning algorithm is used for the post-processing, there are still a large number of sequential patterns forming the feature space. Their algorithm does not tackle some important problems such as how to efficiently and effectively select discriminative features from a large feature space. Tseng and Lee[12] proposed a Classify-By-Sequence (CBS) algorithm to combine sequential pattern mining and classification. In their paper, two algorithms, CBS_Class and CBS_All are 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. The experimental results in their paper show that the performance of CBS_Class is better than the performance of CBS_All. Exarchos[14] proposed to combine sequential pattern mining and classification followed by an optimisation algorithm. The accuracy of their algorithm is higher than that of CBS. However optimisation is a very time-consuming procedure. In fact, in frequent pattern-based sequence classification, efficiency is the principal bottleneck problem to prevent it from being used in real business world. None of the above three sequence classification algorithms tackles this problem well.

3 Problem Statement

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

Consider two sequences $s = \langle s_1, s_2, \ldots, s_n\rangle$ and $t = \langle t_1, t_2, \ldots, t_m\rangle$. We say that s is a subsequence of t if there exist integers $j_1 < j_2 < \cdots < j_n$ such that $s_1 \subseteq t_{j1}, s_2 \subseteq t_{j2}, \ldots, s_n \subseteq t_{jn}$. Note that for sequences of simple items the above condition is translated to $s_1 = t_{j1}, s_2 = t_{j2}, \ldots, s_n = t_{jn}$. A sequence t is said to *contain* another sequence s if s is a subsequence of t, in the form of $s \subseteq t$.

3.1 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 $sup(s) > min\_sup$. 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 Classifiable Sequential Patterns

Let $\mathcal{T}$ be a finite set of *class labels*. A *sequential classifier* is a function

$$\mathcal{F} : \mathcal{S} \to \mathcal{T}. \tag{1}$$

In sequence classification, the classifier $\mathcal{F}$ is built on the base of frequent *classifiable sequential patterns* $\mathcal{P}$.

Definition 3.1 (Classifiable Sequential Pattern). *Classifiable Sequential Patterns* (*CSP*) *are frequent sequential patterns for the sequential 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 sequential classifier can be formulised as

$$\mathcal{F} : s \xrightarrow{\mathcal{P}} \tau. \tag{2}$$

That is, for each sequence $s \in \mathcal{S}$, $\mathcal{F}$ predicts the target class label of s based on the sequential classifier which is composed of 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 antecedent p_a of the classifiable sequential pattern p.

4 Interestingness Measure

Suppose there is a pattern $A \Rightarrow B$, there are three key properties (KP) and five other properties (OP) for a good interestingness measure (M)[29]. Basically, a good measure should satisfy the following three key properties:

- KP_1: $M = 0$ if A and B are statistically independent;
- KP_2: M monotonically increases with $P(A, B)$ when $P(A)$ and $P(B)$ remain the same; and
- KP_3: M monotonically decreases with $P(A)$ (or $P(B)$) when the rest of the parameters ($P(A, B)$ and $P(B)$ or $P(A)$) remain unchanged.

Furthermore, there are five other properties. A measure should satisfy them or not depending on different conditions. The five properties are:

- OP_1: symmetry under variable permutation;
- OP_2: row/column scaling invariance;
- OP_3: antisymmetry under row/column permutation;
- OP_4: inversion invariance; and
- OP_5: null invariance.

In this paper, we would like to find the sequential patterns that either positively or negatively relate to a class. To this end, the interestingness measures should satisfy three key properties plus OP_3, that is, they should distinguish positive and negative correlations of a table.

In this paper, *Class Correlation Ratio* (*CCR*)[24] is used as the principal interestingness measure since it meets the above requirements. The Class Correlation Ratio (*CCR*) can be defined given a contingency table shown in Table 1.

Table 1. 2 by 2 Feature-Class Contingency Table

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

Class correlation ratio is to measure how correlated the sequential pattern p_a is with the target class τ compared to negative class $\neg\tau$. It has the following formula:

$$CCR(p_a \to \tau) = \frac{\hat{corr}(p_a \to \tau)}{\hat{corr}(p_a \to \neg\tau)} \tag{3}$$

$$= \frac{a \cdot (c+d)}{c \cdot (a+b)}. \tag{4}$$

Here,

$$\hat{corr}(p_a \to \tau) = \frac{sup(p_a \cup \tau)}{sup(p_a) \cdot sup(\tau)} \tag{5}$$

$$= \frac{a \cdot n}{(a+c) \cdot (a+b)}, \tag{6}$$

is the correlation between p_a and the target class τ.

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

can distinguish whether a pattern is positively and negatively correlated to a target class. Also, it is a asymmetric measure to differentiate the target class from antecedent sequential patterns.

5 Sequence Classification

Given a sequence database $\mathcal{S}$ and a set of target class $\mathcal{T}$, the conventional method to build sequential classifier is to mine for the complete set of patterns with respect to a given support threshold *min_sup*. Then, a number of processes are adopted to work on the large sequential pattern set to select the discriminative patterns for the classification.

In our algorithm, we do not follow the conventional algorithms to mine for the complete set of classifiable sequential pattern set. Instead, we build the sequential classifier in a hierarchical way as shown in Fig.1.

The outline of the hierarchical sequence classification algorithm is as follows.

1) Applying sequential pattern mining algorithm on the input dataset. Instead of calculating support and confidence of each candidate pattern, the algorithm in this paper calculates the frequency of each classifiable sequential pattern and the corresponding *CCR*. Since $CCR = 1$ means the antecedent is independent of the target class, only the patterns with $CCR > 1 + m_1$ or $CCR < 1 - m_2$ are selected as the candidate classifiable sequential patterns. Here m_1 and m_2 are predefined margins. Aggressive strategy is used in our frequent sequential pattern mining stage, which is illustrated in Subsection 5.1.

2) After the sequential patterns are discovered, pattern pruning is implemented on the classifiable sequential pattern set. We follow the pattern pruning algorithm in [21]. The only difference is, in our algorithm, *CCR* is used as the measure for pruning instead of *confidence*. The brief introduction of our pruning algorithm is shown in Subsection 5.2.

3) Conduct serial coverage test following the ideas in [20–21]. The patterns which can correctly cover at least one training sample in the serial coverage test form the first level of sequential classifier. Please see Subsection 5.3 for the description of the serial coverage test.

4) Since aggressive pattern mining strategy is used in sequential pattern mining step, only a small number of classifiable sequential patterns are discovered at the first level. Hence in the serial coverage test, a large portion of training samples may not be covered by the mined classifiable sequential patterns. These training samples are fed back to Step 1). With updated parameters, sequential pattern mining is again implemented. After pattern pruning and coverage test (Step 2) to Step 3)), the samples that still cannot be covered are fed back for sequential pattern mining until the predefined thresholds are reached or all samples are covered. The classifiable sequential patterns mined from each loop form the sequential classifier at each level.

5) The final sequential classifier is the hierarchical one consisting of the above sub-classifiers at different levels.

5.1 Sequential Pattern Mining

It is known that support is an anti-monotonic measure. The monotonics can dramatically reduce the

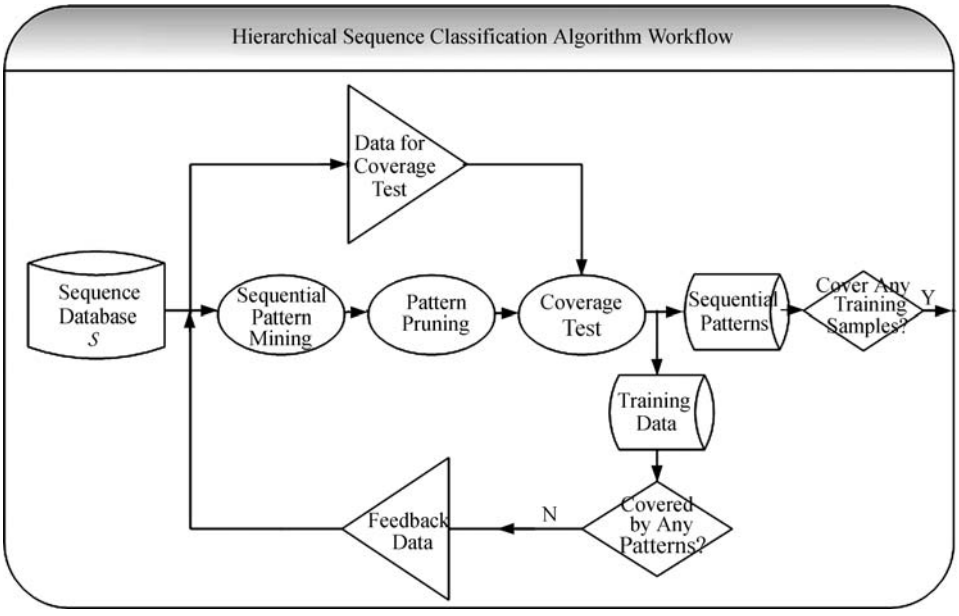


Fig.1. Hierarchical sequence classification algorithm.

searching space in frequent pattern mining algorithms. However, the general idea of a pattern being correlated to a target class is not anti-monotonic. To avoid examining the entire space, we use search strategies that ensure the concept of being potentially interesting is anti-monotonic. That is, $p_a \rightarrow c$ might be considered as potentially interesting if and only if all $\{p'_a \rightarrow c | p'_a \subset p_a\}$ have been found to be potentially interesting. In this algorithm, we select a new item in such a way that it makes a significant positive contribution to the pattern, when compared to its all generalisations. The pattern $p_a \rightarrow c$ is potentially interesting only if the test passes for all generalisations. This effectively tells us that, when compared to the generalisations, all of the items in the antecedent of the pattern make a significant positive contribution to the patterns associated with the target class. This technique prunes the search space most aggressively, as it performs $|p_a|$ tests per rule, where $|\cdot|$ is the length of a pattern.

5.2 Pattern Pruning

In this paper, two pattern pruning techniques are used to reduce the size of the mined sequential pattern set. The first technique is redundancy removal. We use general and high ranking patterns to prune more specific and low ranking patterns. Here the ranking is based on the weighted score which is defined as follows:

$$W_s = \begin{cases} CCR, & CCR > 1, \\ \dfrac{1}{CCR}, & CCR < 1, \\ M, & CCR = 0, \end{cases} \tag{7}$$

where M is a predefined integer for the maximum value of the CCR in the algorithm.

Suppose two sequences p_i and p_j are in the mined sequential pattern set $\mathcal{P}$. If the following conditions are met, the pattern p_j is pruned.

$$\begin{cases} p_i \subseteq p_j, \\ W_s^{p_i} > W_s^{p_j}. \end{cases}$$

The second pruning technique is significance testing. For each pattern $p_a \rightarrow c$, we test whether p_a is significantly correlated with c by χ^2 testing. Only the χ^2 value of a pattern greater than a threshold (in this paper, 3.84) is kept for further processing. All the other patterns are pruned.

5.3 Coverage Test

The serial coverage test is invoked after patterns have been ranked as above. Only the sequential patterns that cover at least one training sample not covered by a higher ranked pattern are kept for later classification. For each sorted sequential pattern starting from the top ranked one s_1, a pass over the training data set to find all objects that match s_1 is performed. Once training objects covered by s_1 are located, they are removed from the training data set and s_1 is inserted into the sub-classifier. The process repeats for the remaining ordered patterns until all training objects are covered or all sorted patterns have been checked. If a pattern is unable to cover at least a single training object, then it will be discarded.

5.4 Weighted Classifier

After serial coverage test, we have a set of sequential patterns to build the sub-classifier at each level of the final classifier. In this paper, we follow two strategies to build each sub-classifier as follows.

• Highest weighted score (CCR_{highest}). Given a sequence instance s, the class label corresponding to the classifiable sequential pattern with highest weighted score is assigned to s.

• Multiple weighted scores (CCR_{multi}). Given one sequence instance s, all the classifiable sequential patterns at one level covering s are extracted. It is not difficult to compute the sum of the weighted score corresponding to each target class. The class label corresponding to the largest weighted score sum is assigned to s.

5.5 Classifier Testing

When the hierarchical classifier is built, it can be used for the prediction of the target class on a sequence database, which is also conducted in a hierarchical way. Suppose there is a sequence instance s. Firstly, sub-classifier on the first level is used to test on s. If s can be covered by the sub-classifier, the target class of s is predicted following the proposed algorithms CCR_{highest} or CCR_{multi}. Otherwise s is input to the next level to test whether it is covered by the next sub-classifier until the last level. Since the prediction is also implemented in a hierarchical way, the coverage test is not needed to be done on the whole classifiable sequential pattern set. Hence the efficiency of prediction is also improved in our algorithm.

6 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 the customer transactional activity data.

The dataset used for the sequence classification is

composed of customer activity data and debt data. In Centrelink, every customer contact (e.g., a circumstance change) will trigger a sequence of activities. As a result, large volumes of activity-based transactions are recorded in activity transactional files. In the original activity transactional table, each activity has 35 attributes, of which 4 are used in the case study. These attributes are "CRN" (Customer Reference Number) of a customer, "Activity Code", "Activity Date" and "Activity Time" of each activity respectively shown in Table 2. We sort the activity data according to "Activity Date" and "Activity Time" to construct the activity code 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 a debt are kept for the sequence classification task. After data cleaning, there are 15 931 activity sequences including 849 831 activity records used.

Table 2. Samples of Centrelink Activity Data

CRN	Act_Code	Act_Date	Act_Time
******002	DOC	20/08/2007	14:24:13
******002	RPT	20/08/2007	14:33:55
******002	DOC	05/09/2007	10:13:47
******002	ADD	06/09/2007	13:57:44
******002	RPR	12/09/2007	13:08:27
******002	ADV	17/09/2007	10:10:28
******002	REA	09/10/2007	7:38:48
******002	DOC	11/10/2007	8:34:36
******002	RCV	11/10/2007	9:44:39
******002	FRV	11/10/2007	10:18:46
******002	AAI	07/02/2008	15:11:54

In this case study, we build a three-level hierarchical classifier. The thresholds of CCR at the three levels are as follows.

$$T^1 = \begin{cases} 2, & \text{if } CCR > 1, \\ 0.5, & \text{if } CCR < 1, \end{cases}$$

$$T^2 = \begin{cases} 1.2, & \text{if } CCR > 1, \\ 0.8, & \text{if } CCR < 1, \end{cases}$$

$$T^3 = \begin{cases} 1.05, & \text{if } CCR > 1, \\ 0.95, & \text{if } CCR < 1. \end{cases}$$

In order to evaluate the accuracy of the proposed algorithm, we implement another algorithm CCR_{CMAR} which is similar to CMAR[21]. CCR_{CMAR} is also implemented in our hierarchical framework. At each level, the sub-classifier is trained using a weighted χ^2 on multiple patterns. We compare the accuracy of CCR_{CMAR} to our proposed algorithms CCR_{highest} and CCR_{multi} at difference *min_sup* levels. The results are shown in Table 3. Note that in all of our experiments, 60% of the dataset is extracted as a training set while the remainder is used as a testing set. Maintaining the ratio of training and testing sets but randomly dividing them, we test the built classifier for five times.

In Table 3, at all *min_sup* levels, CCR_{multi} outperforms CCR_{highest}. This result again verifies that the classifier constructed from only using the highest ranking pattern for one instance suffers from overfitting. Between the two algorithms both using multiple patterns for one instance, CCR_{multi} and CCR_{CMAR}, we can see that CCR_{multi} outperforms CCR_{CMAR} at all *min_sup* levels. When *min_sup* becomes greater, the difference between the two algorithms increases, which means our algorithm is more robust than CCR_{CMAR} when less patterns are discovered for classification.

Table 3. Performance of Different Algorithms

min_sup (%)	No. Patterns	CCR_{CMAR} (%)	CCR_{highest} (%)	CCR_{multi} (%)
1	39 220	75.0	72.7	75.2
2	10 254	74.4	71.8	74.9
5	1 116	69.4	70.9	72.4
10	208	64.2	61.0	66.7

In order to compare the efficiency of our algorithm and conventional algorithms, we also implement the standard sequential mining algorithm using SPAM[17]. In our case study, SPAM takes too long time if *min_sup* $< 5\%$. So we mined for two sets of sequential patterns, with *min_sup* $= 5\%$ and min_sup $= 10\%$, which are called "PS05" and "PS10" respectively. When *min_sup* $= 5\%$, the number of the mined patterns is 2 173 691. In the coverage test, we would check whether a pattern covering each sample. Suppose we have 15 931 sequences. The total number of the possible checking between the sequence data and the mined sequential patterns with *min_sup* $= 5\%$ is $2173691 \times 15931 \times 0.05 = 1.73 \times 10^9$. When *min_sup* $= 10\%$, the number of the mined patterns is 773 724. And the total number of possible matching between the sequence data and the sequential patterns is $773724 \times 15931 \times 0.1 = 1.12 \times 10^9$. Even after pattern pruning, it is still inhibitorily time-consuming to implement serial coverage test and build classifier on such a large set of patterns. In this experiment, we ranked the patterns according to their CCRs and extracted the first 4000 and 8000 patterns from "PS10" and "PS05" and called them "PS10-4K", "PS05-4K", "PS10-8K" and "PS05-8K" respectively. Hence we have four classifiers built on the above four sequential pattern sets. The comparison to our classifiers CCR_{multi} at *min_sup* $= 5\%$ and at *min_sup* $= 10\%$ is shown in Table 4. In Table 4, the "No. Patterns" is the number of the patterns obtained from sequential pattern mining stage. The "Accuracy" is the accuracy of each classifier on the same testing dataset.

Table 4. Comparison of the Proposed Algorithm to Conventional Algorithm

min_sup	CCR_{multi}		CCR_{SPAM}			
(%)	No. Patterns	Accuracy (%)	No. Patterns	Accuracy (%)	No. Patterns	Accuracy (%)
10	208	66.7	4 000	64.7	8 000	65.7
5	1 116	72.4	4 000	69.9	8 000	70.6

For the convenience of presentation, we call the algorithm based on SPAM "CCR_{SPAM}". We have the following two findings from Table 4. Firstly, the accuracy of classifier increases when *min_sup* decreases and the number of patterns increases. This finding happens on both our algorithm and CCR_{SPAM}. Actually this finding is also proven by Table 3. When the *min_sup* decreases from 10% to 1%, the accuracy of the classifiers will increase monotonically. Secondly, the proposed algorithm outperforms CCR_{SPAM} even though it uses much less patterns than CCR_{SPAM}. When $min\_sup = 10\%$, there are only 208 patterns mined in our algorithm while the accuracy is 66.7%. Even 8000 patterns are selected for building the sequential classifier in CCR_{SPAM}, the accuracy is 65.7% while the accuracy decreases to 64.7% when input pattern number is 4000. When $min\_sup = 5\%$, we have the similar finding with a little bigger difference in the classifier accuracy.

From this experiment we can see that our algorithm outperforms CCR_{SPAM} in both efficiency and accuracy. We believe that one of the reasons for the improvement in accuracy is that our algorithm uses less redundant sequential patterns.

7 Conclusion and Future Work

In this paper we propose a novel hierarchical algorithm for sequence classification. In the final classifier, the sequential patterns are organised at different levels and only a small set of sequential patterns are used for training or testing on each level. Hence the searching space in our algorithm is much smaller than that of conventional algorithms. In order to select discriminative sequential patterns at each level, we employ cross correlation ratio as the principal interestingness measure. Pattern pruning and coverage test are also used to select the sequential patterns for the final classifier. Our algorithm is tested on a real-world database and the case study shows its efficiency and effectiveness.

It is not difficult to see that the algorithm proposed in our paper is a closed-loop system while most of the existing ones are open-loop. There are two main advantages with our closed-loop algorithm. Firstly, there is feedback from the classification to the frequent pattern mining. Hence, only small sets of discriminative patterns are discovered in the pattern mining stage. Secondly, since the samples are fed back to generate more discriminative patterns in each loop, the performance of the system is much better in both efficiency and effectiveness. The formulisation of our closed-loop system and the updating of the system parameters will constitute part of our future work.

Acknowledge We are grateful to Mr. Peter Newbigin and Mr. Brett Clark from Payment Reviews Branch, Business Integrity Division, Centrelink, Australia, for extracting data and providing domain knowledge.

References

[1] Juang B H, Chou W, Lee C H. Minimum classification error rate methods for speech recognition. *IEEE Trans. Speech and Audio Signal Processing*, May 1997, 5(3): 257–265.

[2] Lodhi H, Saunders C, Shawe-Taylor J, Cristianini N, Watkins C. Text classification using string kernels. *Journal of Machine Learning Research*, 2002, 2: 419–444.

[3] Baker L D, McCallum A K. Distributional clustering of words for text classification. In *Proc. the 21st ACM SIGIR International Conference on Research and Development in Information Retrieval*, Melbourne, Australia, August 24–28, 1998, pp.96–103.

[4] Wu C, Berry M, Shivakumar S, McLarty J. Neural networks for full-scale protein sequence classification: Sequence encoding with singular value decomposition. *Machine Learning*, October, 1995, 21(1/2): 177–193.

[5] Chuzhanova N A, Jones A J, Margetts S. Feature selection for genetic sequence classification. *Bioinformatics*, 1998, 14(2): 139–143.

[6] She R, Chen F, Wang K, Ester M, Gardy J L, Brinkman F S L. Frequent-subsequence-based prediction of outer membrane proteins. In *Proc. the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD 2003*), Washington DC, USA, August 24–27, 2003, pp.436–445.

[7] Sonnenburg S, Rätsch G, Schäfer C. Learning interpretable SVMs for biological sequence classification. In *Proc. Research in Computational Molecular Biology* (*RECOMB 2005*), Cambridge, USA, May 14–18, 2005, pp.389–407.

[8] Hakeem A, Sheikh Y, Shah M. CASE^E: A hierarchical event representation for the analysis of videos. In *Proc. the Nineteenth National Conference on Artificial Intelligence* (*AAAI 2004*), San Jose, USA., July 25–29, 2004, pp.263–268.

[9] Eichinger F, Nauck D D, Klawonn F. Sequence mining for customer behaviour predictions in telecommunications. In *Proc. the Workshop on Practical Data Mining at ECML/PKDD*, Berlin, Germany, September 18–22, 2006, pp.3–10.

[10] Centrelink Annual Report 2007-2008. Technical Report, Centrelink, 2008.

[11] Lesh N, Zaki M J, Ogihara M. Mining features for sequence classification. In *Proc. the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, California, USA, August 15–18, 1999, pp.342–346.

[12] Tseng V S M, Lee C-H. CBS: A new classification method by using sequential patterns. In *Proc. SIAM International*

Conference on Data Mining (*SDM 2005*), Newport Beach, USA, April 21–23, 2005, pp.596–600.

[13] Xing Z, Pei J, Dong G, Yu P S. Mining sequence classifiers for early prediction. In *Proc. SIAM International Conference on Data Mining* (*SDM 2008*), Atlanta, USA, April 24–26, 2008, pp.644–655.

[14] Exarchos T P, Tsipouras M G, Papaloukas C, Fotiadis D I. A two-stage methodology for sequence classification based on sequential pattern mining and optimization. *Data & Knowledge Engineering*, September 2008, 66(3): 467–487.

[15] Agrawal R, Srikant R. Mining sequential patterns. In *Proc. the Eleventh IEEE International Conference on Data Engineering* (*ICDE 1995*), Taipei, China, March 6–10, 1995, pp.3–14.

[16] Pei J, Han J, Mortazavi-Asl B, Pinto H, Chen Q, Dayal U, Hsu MC. PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. In *Proc. the 17th IEEE International Conference on Data Engineering* (*ICDE 2001*), Heidelberg, Germany, April 2–6, 2001, pp.215–224.

[17] Ayres J, Flannick J, Gehrke J, Yiu T. Sequential pattern mining using a bitmap representation. In *Proc. the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD 2002*), Edmonton, Canada, July 23–26, 2002, pp.429–435.

[18] Yan X, Han J, Afshar R. Clospan: Mining closed sequential patterns in large datasets. In *Proc. SIAM International Conference on Data Mining* (*SDM 2003*), San Francisco, USA, May 1–3, 2003, pp.166–177.

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

[20] Liu B, Hsu W, Ma Y. Integrating classification and association rule mining. In *Proc. the 4th ACM International Conference on Knowledge Discovery and Data Mining* (*KDD 1998*), Menlo Park, USA, August 27–31, 1998, pp.80–86.

[21] Li W, Han J, Pei J. CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proc. the First IEEE International Conference on Data Mining* (*ICDM 2001*), Los Alamitos, USA, Nov. 29–Dec.2, 2001, pp.369–376.

[22] Cheng H, Yan X, Han J, Hsu C-W. Discriminative frequent pattern analysis for effective classification. In *Proc. 23rd IEEE International Conference on Data Engineering* (*ICDE 2007*), Istanbul, Turkey, April 17–20, 2007, pp.716–725.

[23] Han J, Cheng H, Xin D, Yan X. Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery*, August 2007, 15(1): 55–86.

[24] Verhein F, Chawla S. Using significant, positively associated and relatively class correlated rules for associative classification of imbalanced datasets. In *Proc. the Seventh IEEE International Conference on Data Mining* (*ICDM 2007*), Omaha, USA, Oct. 28–31, 2007, pp.679–684.

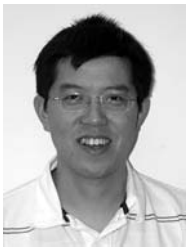
[25] Antonie M L, Zaiane O R, Holte R C. Learning to use a learned model: A two-stage approach to classification. In *Proc. the Sixth International Conference on Data Mining* (*ICDM 2006*), Hong Kong, China, Dec. 18–22, 2006, pp.33–42.

[26] Baralis E, Garza P. A lazy approach to pruning classification rules. In *Proc. the Second IEEE International Conference on Data Mining* (*ICDM 2002*), Maebashi City, Japan, Dec. 9–12, 2002, pp.35–42.

[27] Wang J, Karypis G. Harmony: Efficiently mining the best rules for classification. In *Proc. SIAM International Conference on Data Mining* (*SDM 2005*), Newport Beach, USA, April 21–23, 2005, pp.205–216.

[28] Cheng H, Yan X, Han J, Yu P S. Direct discriminative pattern mining for effective classification. In *Proc. the 24th IEEE International Conference on Data Engineering* (*ICDE 2008*), Cancun, Mexico, April 7–12, 2008, pp.169–178.

[29] Tan P-N, Kumar V, Srivastava J. Selecting the right interestingness measure for association patterns. In *Proc. the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD 2002*), Edmonton, Canada, July 23–26, 2002, pp.32–41.



Huaifeng Zhang is a senior Data Mining specialist in Data Mining Section within Centrelink, Australia. Dr. Zhang was awarded the Ph.D. degree from Chinese Academy of Sciences (CAS) in 2004. He has more than 40 publications in the previous five years, including one book published by Springer, eight articles in journals, three chapters in edited books. His research interests include combined pattern mining, sequence classification, behaviour analysis and modeling, etc.

Yanchang Zhao is a postdoctoral research fellow in Data Sciences & Knowledge Discovery Research Lab, Faculty of Engineering & IT, University of Technology, Sydney, Australia. His research interests are association rules, sequential patterns, clustering and post-mining. He has published more than 30 papers on the above topics, including six journal articles, one edited book and three book chapters. He has served as chair of two international workshops, program committee member for 14 international conferences and reviewer for 9 international journals and over a dozen of other international conferences.

Longbing Cao is an associate professor in Faculty of Engineering & IT, University of Technology, Sydney, Australia. He is the director of Data Sciences & Knowledge Discovery Research Lab. His research interest focuses on domain driven data mining, multi-agents, and the integration of agent and data mining. He is a chief investigator of three ARC (Australian Research Council) Discovery projects and two ARC Linkage projects. He has over 50 publications, including one monograph, two edited books and 10 journal articles. He is a program co-chair of 11 international conferences.

Chengqi Zhang is a research professor in Faculty of Engineering & IT, University of Technology, Sydney, Australia. He is the director of UTS Research Centre for Quantum Computation and Intelligent Systems and a chief investigator in Data Mining Program for Australian Capital Markets on Cooperative Research Centre. He has been a chief investigator of eight research projects. His research interests include data mining and multi-agent systems. He is a co-author of three monographs, a co-editor of nine books, and an author or co-author of more than 150 research papers. He is the chair of the ACS (Australian Computer Society) National Committee for Artificial Intelligence and Expert Systems, a chair/member of the steering committee for three international conferences.

Hans Bohlscheid is an executive in the Australian Public Service, Hans' present role as business manager for the data mining was preceded by a long career in education where he held a number of teaching and principal positions. For the last four years he has been responsible for the development and implementation of Commonwealth Budget initiatives based on changes to legislation and policy. During this period he has managed a considerable suite of projects, however it is his recent involvement in a pilot which sought to determine the effectiveness of data mining as a predictive and debt prevention tool, that has shifted his focus to research and analysis. In addition to his government responsibilities, Hans is currently managing a 3-year University of Technology Sydney research project which is funded through an Australian Research Council Linkage Grant in partnership with the Commonwealth. He is a partnership Investigator and industry advisor to the University's Data Sciences and Knowledge Discovery Laboratory, and he has co-authored a number of publications and book chapters relating to data mining. His personal research involves an examination of project management methodology for actionable knowledge delivery.