PERGAMON

Expert Systems with Applications 25 (2003) 293–302

Expert Systems with Applications

www.elsevier.com/locate/eswa

Goal-oriented sequential pattern for network banking churn analysis

Ding-An Chiang[a], Yi-Fan Wang[b,*], Shao-Lun Lee[a], Cheng-Jung Lin[a]

[a]*Department of Information Engineering, Tamkang University, Tanshui, Taipei, Taiwan, ROC*
[b]*Department of Information Management, Chang Gung Institute of Technology, Kwei-Shan, Tao-Yuan, Taiwan, ROC*

Abstract

Discovering sequential patterns is one of the most important task in data mining. In this paper we propose an efficient algorithm, called Goal-oriented sequential pattern. It can provide enterprises warning signs soon before they are losing valuable customers and give them reference for decision making. Experiments comparing Apriori showed that Goal-oriented is more efficient, and performs reasonably well for the rules.

Keywords: Data mining; Association rule; Sequential pattern; Goal-oriented; Retention analysis

1. Introduction

The Pareto Principle or the 80:20 Rule is commonly employed in customer relationship management in which 80% of the company income comes from 20% of the major customers while 90% of the company income is generated from the basic customers. According to Don Peppers and Martha Rogers, the marketing experts, most enterprises average lost 25% customers annually. However, the cost of obtaining a new customer is five times higher than maintaining an existing customer (Gronroos, 1984; Reich & Benbasat). Nevertheless, the major concerns of the enterprises today are to maintain customers' loyalty and to find out the possible reasons for the failures of keeping customer loyalty (Reichheld, 1996). Further, to establish a system that can get alert before losing customers in order to take further actions to keep the customer loyalty is also the main consideration. It is very important for every company to fully understand the changes of numbers about their customers and how the changes affect the company. The prime concern for this research is to help the company understand the customer behaviors and list out the possible losing customers in order to retain customers.

In this research, we specify how to judge whether a customer is leaving and the retaining strategies. The Sequential Pattern cannot conduct a research function by a specific item; meanwhile, the regulations it figures are relatively irrelevant. This does not only increase the difficulties for interpreting the regulations, but also lack of efficiency. Therefore, to alter the flaw of being lack of efficiency, and the incapacity of searching a specific item, Goal-oriented pattern is adapted in this research to implement the searches for some particular items. We use the association analysis, comprising sequential patterns (Agrawal & Srikant, 1995; Han et al., 2000; Han, Pei, & Yin, 2000; Masseglia, Cathala, & Poncelet, 2001; Pei et al., 2001; Tseng & Hsu, 2001) and association rules (Agrawal, Imielinski, & Swami, 1993; Agrawal & Srikant, 1994; Park, Chen, & Yu, 1995; Srikant & Agrawal, 1995). The cause of leaving customers can be generated by sequential patterns. Consequently, the strategy planners can take proper actions to retain customers. We list the possible reasons of customer leaving by means of reverse sequence in which the original sequence is being reversed so that the main concerns of the item are arranged in front of the array. The contrasts between the goal items are then used to exclude the irrelevant items. Finally, the reversed sequence is re-reverse again to the original sequence in order to enhance the efficiency for searching the goal item rules.

* Corresponding author.
E-mail addresses: yfwang@mail.cgit.edu.tw (Y.-F. Wang), chiang@cs.tku.edu.tw (D.-A. Chiang).

0957-4174/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0957-4174(03)00073-3

2. Relative research

2.1. Association rules

Association rules are mainly used to find out the relations between items or features that happened synchronously in the database, such as generating the data about the groceries that are bought at the same time when people shop in the mall. For instance, 80% of people who merchandise milk buy bread as well. As soon as the decision maker fetches this information, strategies, such as setting the relevant counters nearby or held a promotion, can be generated from this aspect. Therefore, the main purpose of implementing association rules is to find out the synchronous relationship by analyzing the random data for the reference of making decisions.

The definition of association rule is as following: Make $I = \{i_1, i_2, \ldots, i_m\}$ as the itemset, in which each item represents a specific commodity. D stands for a trading database in which each transaction T represents a itemset. That is $T \subseteq I$. Each itemset is a non-empty sub-itemset of I, and the only identify code is *TID*. Each itemset $X \subset I$ has a measure standard—Support, to evaluate the statistical importance in D. Support(X, D) denotes the rate of merchandising X in transaction D.

The format of the association rule is $X \rightarrow Y$, in which $X, Y \subset I$, and $X \cap Y = \varnothing$. The interpretation of this association rule is that if X is purchased, Y can be bought at the same time. Each rule has a measuring standard called Confidence; i.e. Confidence$(X \rightarrow Y)$ = Support$(X \cup Y, D)$/Support(X, D). In this case, Confidence$(X \rightarrow Y)$ denotes if the merchandise including X, the chance of buying Y is relatively high.

There are two steps to find out the association rules. First, detecting the large itemset. Second, generating the association rules by utilizing the large itemset. Therefore, to explore the association rules also means to find out all the association rules of $X \rightarrow Y$ formats and meet the following conditions:

1. Support$(X \cup Y, D) \geq$ Minsup
2. Confidence$(X \rightarrow Y) \geq$ Minconf

The Minsup and Minconf are both set by the users. In general, the numbers of the transactions that comprising X is called the support of X, denoted by σx. Make Minsup the minimum value of support. If the support of X meets the condition, $\sigma x \geq$ Minsup, X is the large itemset.

As for the exploration of the association rules, many researchers take the Apriori algorithm (Agrawal & Srikant, 1994) supported by Agrawal et al. as the basic formulation. The Apriori algorithm forms the rules by way of simple and sequential progresses to lay out the relationship in the database. It is widely adopted to find out the large itemset, as demonstrated in Fig. 1. By means of this algorithm, each support can be calculated followed by scanning the transaction database. Further, judge if the support exceeds the minimum support so that the large itemset can be identified. In the following rounds, we use the itemset selected previously to be the seed itemset, which can be utilized to generate a new potential large itemset, called Candidate itemsets. Calculate the support of each candidate itemset to decide whether the itemsets can be the genuine large itemset and be the seed itemset for the next round. This algorithm can be ceased when no further candidate itemset can be generated.

```
L1={large 1-itemsets};
For (k=2; Lk-1≠∅; k++) do begin
    Ck = Apriori-gen (Lk-1);
    For all transactions t ∈ D do begin
        Ct = subset (Ck, t);
        For all candidates c ∈ Ct do
            c.count + +;
    End
    Lk = {c ∈ Ck | c.count □ Minsup}
End
Answer = ∪kLk
Apriori-gen()
{
Insert into Ck
Select p.item1, p.item2, … , p.itemk-1, q.itemk-1
From Lk-1 p, Lk-1 q
Where p.item1 = q.item1, … , p.itemk-2 = q.itemk-2,
p.itemk-1 < q.itemk-1;
Join step
For itemsets c ∈ Ck do
    For all (k-1)-subsets s of c do
        If (s ∉ Lk-1) then
            Delete c from Ck;}
Pruning step
```

Fig. 1. Apriori algorithm.

For example, in Fig. 2, database D is the original transaction database. Assume the minimum support is 2. First, calculate the number of each item that appears in the transaction database, which is to calculate the support and to evaluate whether the number is bigger than or equal to the minimal support and determine the Large 1-itemsets, L_1. Next, generate candidate 2-itemsets, C2 from $L_1 \times L_1$. Further, calculate the support of C_2 to create L_2. From $L_2 \times L_2$ brings about Candidate 3-itemsets, C_3. In the phase of

TransactionId	TransactionDate	CustomerId	Items
01	2001/02/03	ID0005	30
02	2001/02/25	ID0005	40,70
03	2001/03/26	ID0003	30,50
04	2001/03/27	ID0004	10,20
05	2001/04/02	ID0004	90
06	2001/04/20	ID0003	60
07	2001/07/22	ID0002	20
08	2001/08/06	ID0002	30
09	2001/08/16	ID0002	40,70,80
10	2001/10/02	ID0001	10
11	2001/11/06	ID0001	30
12	2001/11/15	ID0001	40,80

Fig. 2. Original transaction database.

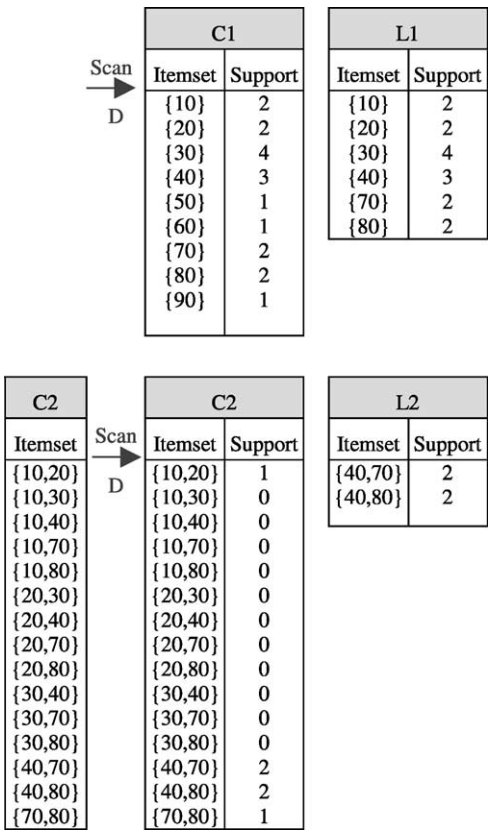


Fig. 3. Generation of candidate itemsets and large itemsets.

join, we have $\{40, 70, 80\}$. However, in the phase of pruning, the sub-itemsets $\{40, 70\}$, $\{40, 80\}$ of $\{40, 70, 80\}$ are both comprised in L_2, but $\{70, 80\}$ is not enclosed in L_2. Thus, it does not meet Candidate 3-itemset C_3. The algorithm, therefore, ceased. In consequence, the large itemsets generated are $L_1 = \{10\}, \{20\}, \{30\}, \{40\}, \{70\}, \{80\}$, $L_2 = \{40, 70\}, \{40, 80\}$ as demonstrated in Fig. 3.

In step 2, the association rules are created by use of the large itemsets. As the large itemsets figured in the previous step are $L_1 = \{10\}, \{20\}, \{30\}, \{40\}, \{70\}, \{80\}$; $L_2 = \{40, 70\}, \{40, 80\}$. Therefore, the association rules we intend to find out are $\{40\} \rightarrow \{70\}$, $\{70\} \rightarrow \{40\}$, $\{40\} \rightarrow \{80\}$ and $\{80\} \rightarrow \{40\}$.

2.2. Sequential patterns

The sequential patterns originate the association rules that occur frequently. It emphasizes the factor of time. What being concerned is the data that related on the base of time. We use this pattern to analyze the association of each individual event chronologically. The major method of mining sequential patterns are Apriori algorithm (Agrawal & Srikant, 1994), DHP algorithm (Park et al., 1995), FP-tree algorithm (Han et al., 2000), and FPL algorithm (Tseng & Hsu, 2001). The sequential pattern can be divided into Sequential Procurement (Agrawal & Srikant, 1995) and Cyclic Procurement (Ozden, Ramaswamy, & Silberschatz, 1998) by the sequence and the section of time. What sequential procurement concerns is the chronological pattern. It only analyses the association of data by the sequence of time. On the other hand, what cyclic procurement concerns are the events that happen in the same time section and whether they reoccur in other time section as well. It emphasizes the variation in different time section.

The main concern in this research is on sequential procurement. In general, there are two phases to create the sequential pattern: Setting large itemset and establishing sequential pattern rules by large itemset. As the association rule, the meaning of sequential patterns is similar to the large itemset, which is to find out the association between items. The only difference is that in sequential patterns, the chronological pattern is one of the major concerns. In phase I, Apriori algorithm introduced in Fig. 1 is applied to find the large itemset. Whereas in phase II, there are five phases to generate sequential pattern rule from the large itemset.

Sort Phase. Customer series is the major figure while transaction time is the sub-major figure to arrange the raw data from the original transaction database. The raw transaction database is demonstrated in Fig. 2. The new database after sorting incrementally by time is showed in Fig. 4.

Large Itemset Phase. Find out all the large itemsets and denote each large itemset with a specific symbol.

Assume the minimum support is 2, from Section 2.2, we get the large itemset as $L_1 = \{10\}, \{20\}, \{30\}, \{40\}, \{70\}, \{80\}$; $L_2 = \{40, 70\}, \{40, 80\}$. Then code them with an integer as showed in Fig. 5.

Transformation Phase. Delete the non-large itemsets from the raw transaction database.

The items not in the large itemset from the raw transaction database are deleted and rearranged data by customers as shown in Fig. 6.

Sequential Phase. Delete the sequences that each contains two items from the data.

The sequences can be generated by algorithm similar to Apriori. The difference from finding the large itemset is in

TransactionId	TransactionDate	CustomerId	Items
10	2001/10/02	ID0001	10
11	2001/11/06	ID0001	30
12	2001/11/15	ID0001	40,80
07	2001/07/22	ID0002	20
08	2001/08/06	ID0002	30
09	2001/08/16	ID0002	40,70,80
03	2001/03/26	ID0003	30,50
06	2001/04/20	ID0003	60
04	2001/03/27	ID0004	10,20
05	2001/04/02	ID0004	90
01	2001/02/03	ID0005	30
02	2001/02/25	ID0005	40,70

Fig. 4. The customer sequence is ordered by increasing transaction time.

Large Itemsets	Mapped To
(10)	1
(20)	2
(30)	3
(40)	4
(70)	5
(80)	6
(40,70)	7
(40,80)	8

Fig. 5. Large itemsets.

this phases, the factor of sequence is being concerned. Hence, the combination of candidate sequence of two items is square of these of single item when we want to find the combination sequence from one item to two items. Similarly, we also can use Apriori-gen Algorithm to find out the candidate sequence if the items are above three. The LS_1 conducted by all large itemsets is shown in Fig. 7. In order to produce CS_2, we proceed $L_1 \times L_1$, which needs 64 combinations (8×8). We can get LS_2 by calculating supported level of CS_2. Similarly, we can get CS_3 via $LS_2 \times LS_2$. Although we generate ⟨3,4,6⟩, ⟨3,4,8⟩ and ⟨3,6,8⟩ in the merging phase, ⟨4,6⟩, ⟨4,8⟩, ⟨6,8⟩ (the subset of ⟨3,4,6⟩, ⟨3,4,8⟩ and ⟨3,6,8⟩, respectively) are not included within L_2 in pruning phase. This cannot satisfy the 3-items's candidate sequence (CS_3) and then algorithm will stop.

Maximal Phase. Find out the maximal sequences within the large sequences.

To link up sequence phase to reduce the time of counting non-maximal sequences. The sequence generated in the previous step is not contained in others is the largest sequence (Fig. 7). Sequence ⟨3,4⟩,⟨3,6⟩ and ⟨3,8⟩ are the largest sequence. Transforming the largest sequence to the original items, we can get our mining sequence pattern (⟨(30)(40)⟩, ⟨(30)(80)⟩, ⟨(30)(40,80)⟩).

3. Problem statement

At the present business field, creating a customer retention analysis model is the first step to start customer relation managements. Many businesses, therefore, hope to figure out the real cause of the loss of a customer, or even to be told that they are about to lose the customer by some clues before it occurs, and then they can propose or make some new sales strategies against the loss in advance. Although many businesses eager to create a customer-losing model to solve the problem of the loss of customers, they cannot find an effective way to solve the problem that has been confused them for a long time. This research, therefore, proposes a new method to solve such a problem, and according to the method, we can find out behavior patterns of losing customers or clues before they stop using some products through mining Sequential Patterns. In this section, we will take network banking services as an example to elaborate this new method, and moreover, to identify the real causes of the loss of customers in different trades.

3.1. Definition of loss

At first, we will make a simple definition for the analysis of the loss of customers in network banking services. We focus on each user who applied for network banking services to find out the periodicity of transaction time of the user. Generally, there are two ways for us to judge whether or not a customer is lost: One is judging by the average periodic transaction days of the customer in the past. If the interval between the present day and the last transaction day is longer than the average interval of transaction days in the past, we can tell that the customer has been lost. The other is judging by the longest interval from one transaction day to another in the past. We can also tell the loss of a customer, if the total of no transaction days since the last transaction day is longer than the longest interval between two transaction days.

3.2. Methodology

In this section, we offer a method to solve the constant loss of customers in network banking services, that is, to find out main reasons why customers stop using the network banking services through Sequential Patterns Algorithm and the definition of loss mentioned in Section 3.1. To identify the real cause of the loss of the customer, common sequential patterns cannot entirely be applied to it. Because if we put all of the transaction data into the program of sequential patterns, it will cost us lots of time to calculate to find out a rule, and even if we have found one, the chances are that it is useless rule.

Customer Id	Original Customer Sequence	Transformed Customer Sequence	After Mapping
ID0001	<(10)(30)(40,80)>	<{(10)}{(30)}{(40)(80)(40,80)}>	<{1}{3}{4,6,8}>
ID0002	<(20)(30)(40,70,80)>	<{(20)}{(30)}{(40)(80)(40,80)}>	<{2}{3}{4,6,8}>
ID0003	<(30,50)(60)>	<{(30)}>	<{3}>
ID0004	<(10,20)(90)>	<{(10)}{(20)}>	<{1}{2}>
ID0005	<(30)(40,70)>	<{(30)}{(40)(70)(40,70)}>	<{3}{4,5,7}>

Fig. 6. Transformed database.

LS1	
Large 1-sequence	Support
<1>	2
<2>	2
<3>	4
<4>	3
<5>	2
<6>	2
<7>	2
<8>	2

LS2	
Large 2-sequence	Support
<3,4>	3
<3,6>	2
<3,8>	2

Scan D

CS2			
Candidate 2-sequence	Support	Candidate 2-sequence	Support
<1,1>	0	<5,1>	0
<1,2>	1	<5,2>	0
<1,3>	1	<5,3>	0
<1,4>	1	<5,4>	0
<1,5>	0	<5,5>	0
<1,6>	1	<5,6>	0
<1,7>	0	<5,7>	0
<1,8>	1	<5,8>	0
<2,1>	0	<6,1>	0
<2,2>	0	<6,2>	0
<2,3>	1	<6,3>	0
<2,4>	1	<6,4>	0
<2,5>	0	<6,5>	0
<2,6>	1	<6,6>	0
<2,7>	0	<6,7>	0
<2,8>	1	<6,8>	0
<3,1>	0	<7,1>	0
<3,2>	0	<7,2>	0
<3,3>	0	<7,3>	0
<3,4>	3	<7,4>	0
<3,5>	1	<7,5>	0
<3,6>	2	<7,6>	0
<3,7>	1	<7,7>	0
<3,8>	2	<7,8>	0
<4,1>	0	<8,1>	0
<4,2>	0	<8,2>	0
<4,3>	0	<8,3>	0
<4,4>	0	<8,4>	0
<4,5>	0	<8,5>	0
<4,6>	0	<8,6>	0
<4,7>	0	<8,7>	0
<4,8>	0	<8,8>	0

Fig. 7. Generation of candidate sequences and large sequences.

3.2.1. Time windows and normalization

For solving the problem described above, we use the concept of Time windows and Normalization to select the data we want. The size of the time window will affect many factors, in other words, the larger the window is, the more the transaction data we have to deal with, the more time we have to spend on calculation, and the more related rules we can find out. The supportability rating, therefore, may be raised. Conversely, the smaller the window is, the less or even none related rules would be found.

Firstly, we have to decide the size of the time window. For example, one year, six months, one-quarter, one month, or one week, etc. Next, in accordance of the characteristics of the problem, we use a way to set time windows that is different from other sequential patterns.

For instance, supposing that we set the time window size as one month, it means that we only select the transaction record of each customer in the last month from the database. It helps us a lot to differentiate the real cause of the loss of the customer, for the transaction activity of each customer before loss is what we are concerned, and these activities will reflect on the transaction data in the last month. Thus, by the concept of Time Windows and Normalization, we can get a better effect in finding a sequential pattern rule. After the time window was selected (e.g. one month), it is very common to select the transaction data for one month from the entire data. However, in order to find out the real factor of the loss of the customer, we use the idea of Normalization. We regard the last transaction date of each customer as a datum point, and so different customers have different datum points. We select the data that is traced back for one month from the datum point of each customer.

As shown in Fig. 8, we take the vertical axis as the customer, the horizontal axis as the transaction date, dark color as the selected data, and light color as the unselected data. The numbers in dark color represent the number of transaction times. When the time window is set as one month, the last transaction date of customer ID0001 is on 15/11/2001, then the data we selected are four transactions from 16/10/2001 to 15/11/2001; the last transaction date of customer ID0002 is on 16/08/2001, then the data we selected are the five transaction data from 17/07/2001 to 16/08/2001; the last transaction date of customer ID003 is on 20/04/2001, then the data we selected are the three transactions between 21/03/2001 and 20/04/2001, and so are the rests on analogy of this.

3.2.2. Data preprocessing and virtual labels

As among the four transactions of customer ID0001, ways of practicing general sequential patterns is regard

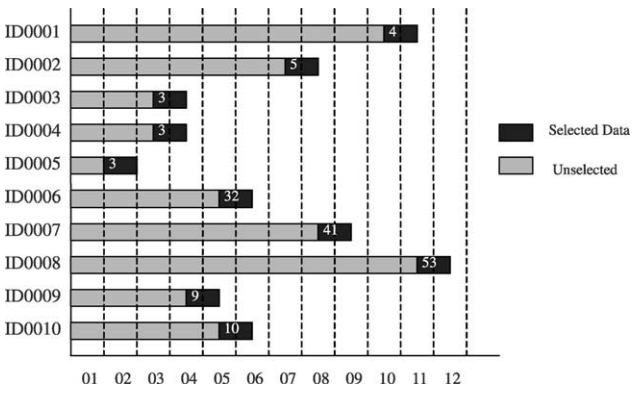


Fig. 8. Use time window to extract data.

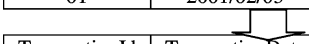
these four descriptions of the transactions as a single record, and then we induce rules by using sequential pattern algorithm. However, due to the characteristics of the problem, we add the concept of virtual labels before we delete duplicated descriptions of transactions. For general sequential patterns cannot find out number of times of every transaction activity when dealing with these four transaction data, it is surely that some information will be omitted by means of such calculation. Thus, in order to retain key information, we put some key descriptions of transaction different virtual labels. Take the customer as an example, if there are two repeated transactions described as 'login in failure', at the same period, we will combine these two consecutive data and label it a new description 'login in failure 2', which means two repeated attempts are made to sign on with incorrect PIN. Corresponding, if there are three repeated transaction descriptions of the customer, which are described as login in failure, at the same period, we will combine these three consecutive data and label it a new description 'login in failure 3', which means PIN is entered incorrectly three consecutive times. These descriptions like Login in failure 2 and Login in failure 3 are virtual labels that separately represent some important information such as the failures may be caused by the user who forgot the PIN or hackers who attempt to enter into the system.

3.3. Goal-oriented sequential patterns

Traditional sequential patterns cannot excavate specific goal items; however, for most of the decision makers, they have a need for finding out the happening order of some specific items. Therefore, we propose a method to deal with the problem that happens when we search the specific goal items. Generally, if the goal items appear at the top of the sequence, we only have to delete the large itemset that is not related to the goal items and then make the rules. Relatively, if the goal items appear at the bottom of the sequence, then we have to use the concept of 'the reversed sequence' to reverse the original sequence and make the items we concern at top of the whole sequence; after that, comparing the goal times and sifting the rules irrelevant to the goal items to increase the rule-finding efficiency. In addition, if the goal items appear at the middle of the sequence, we firstly delete the sequences after the goal items, and then follow the steps as mentioned in above.

As mentioned in Section 2.2, a general way of finding sequential patterns can be resolved into two steps. The first step is to find out large itemset. The second step is use the large itemset to conclude sequential patterns rules. At the first step, we use Apriori calculation to find out the large itemset, while at the second step as to use the large itemset to induce rule, we propose a new algorithm to coincide with the characteristics of the problem that is to find out the rules of loss. The algorithm can delete some rules that are irrelevant to the goal item in advance, and the efficiency of finding rules is increased by a big margin herein.

TransactionId	TransactionDate	CustomerId	Items
12	2001/11/15	ID0001	40,80
11	2001/11/06	ID0001	30
10	2001/10/02	ID0001	10
09	2001/08/16	ID0002	40,70,80
08	2001/08/06	ID0002	30
07	2001/07/22	ID0002	20
06	2001/04/20	ID0003	60
03	2001/03/26	ID0003	30,50
05	2001/04/02	ID0004	90
04	2001/03/27	ID0004	10,20
02	2001/02/25	ID0005	40,70
01	2001/02/03	ID0005	30



TransactionId	TransactionDate	CustomerId	Items
12	2001/11/15	ID0001	40,80
11	2001/11/06	ID0001	30
10	2001/10/02	ID0001	10
09	2001/08/16	ID0002	40,70,80
08	2001/08/06	ID0002	30
07	2001/07/22	ID0002	20

Fig. 9. The customer sequence is ordered by decreasing transaction time.

Although the five steps of the traditional sequential patterns mentioned above can also find out the rule of loss, it is a very inefficient way because we will find many rules, which are mostly useless ones that irrelevant to the loss. Because the purpose of this research is to find out the rule of loss, in other words, to find out the transaction activity of the customer before losing, to deal with the problem like this, we propose a new sequential patterns called Goal-oriented algorithm to solve such a problem towards a specific goal. The problem of mining Goal-oriented sequential patterns is split into six phases.

Descend Sort Phase. The sort phase converts the original transaction database into a database of customer sequences. The customer sequence is a list of transactions ordered by decreasing transaction time. Delete the sequences when it was not beginning with goal item.

We can get a new database (Fig. 9) from decreasing sorting raw one (Fig. 2) by time, if we set the target as 80 and omit the sequence that does not start from the target item.

Large Itemset Phase. Find out all the large itemsets and denote each large itemset with a specific symbol.

Assume the minimum support is 2, we get the large itemset as $L_1 = \{30\}, \{40\}, \{80\}; L_2 = \{40, 80\}$. Then code them with an integer as shown in Fig. 10.

Large Itemsets	Mapped To
(30)	1
(40)	2
(80)	3
(40,80)	4

Fig. 10. Large itemsets.

Customer Id	Original Customer Sequence	Transformed Customer Sequence	After Mapping
ID0001	<(40,80)(30)(10)>	<{(40)(80)(40,80)}{(30)}>	<{2,3,4}{1}>
ID0002	<(40,70,80)(30)(20)>	<{(40)(80)(40,80)}{(30)}>	<{2,3,4}{1}>

Fig. 11. Transformed database.

Transformation Phase. Delete the non-large itemsets from the raw transaction database.

The items not in the large itemset from the raw transaction database are deleted and rearranged data by customers as shown in Fig. 11.

Sequential Phase. Delete the sequences that each contains two items from the data.

Whereas in phase II, the numbers of large itemsets, which contains target item (80) are 2. One is large itemset (80), the mapping value is 3, the others is (40,80), which mapping value is 4. The LS1 conducted by all large itemsets. In order to produce CS_2, we proceed $L_1 \times L_1$. Because of the numbers of large itemsets, which contain target item (80) are 2, so it only needs eight combinations ($4 \times 2 = 8$). The result is shown in Fig. 12. Comparison with traditional method (Section 2.2), it needs compute 64 times, there is a major difference between them. And then we can get LS_2 by calculating supported of CS_2. Similarly, we can get CS_3 via $LS_2 \times LS_2$, because we cannot generate CS_3 in the merging phase, the algorithm will stop.

Maximal Phase. Find out the maximal sequences within the large sequences.

To link up sequence phase to reduce the time of counting non-maximal sequences. The sequence of ⟨3,1⟩ and ⟨4,1⟩ is the large sequences (Fig. 12). Transforming the largest sequence to the original items, we can get our mining sequence pattern ⟨(80)(30)⟩ and ⟨(40,80)(30)⟩.

Reverse Phase. Reverse the maximal sequences.

Reverse the maximal sequences ⟨(80)(30)⟩ and ⟨(40,80)(30)⟩, and we get the sequences ⟨(30)(80)⟩ and ⟨(30)(40,80)⟩. Therefore, the goal oriented sequential pattern rules we intend to find out are ⟨(30)(80)⟩ and ⟨(30)(40,80)⟩.

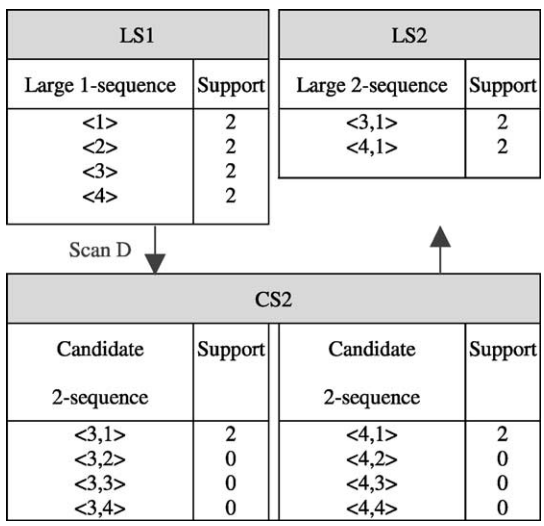


Fig. 12. Generation of candidate sequences and large sequences.

4. The experiment results

To examine the executing efficiency of the Goal-oriented sequential pattern algorithm, we separately design some experiments to examine our method, and compare it with Apriori algorithm. At last, we will explain the result of our experiment. In Section 4.1, we will introduce experimental environments, including the experiment desktop, the practicing program language and information resources. In Section 4.2, we will introduce the experiment results and compare the executing efficiency of the Goal-oriented sequential patterns algorithm. In Section 4.3, we will introduce the customer loss analysis result.

4.1. Experimental environments

1. Experimental platform
   - CPU: Pentium III 600 MHz
   - Memory: 392 MB RAM'13.9 GB HD
   - Operating System: Window 2000
   - Database: SQL Server 2000
2. Implementation Language
   - Visual Basic 6.0
3. Experimental Data Source
   - We analyze the customers' data from a famous network banking in Taiwan, total more than one million data. These data are banks' customers more than 6 months transaction data.

4.2. The executing efficiency analysis

Fig. 13 shows the executing frame of Apriori algorithm, and we donot have to restrict its items of producing rules, while the executing frame of Goal-oriented algorithm as shown in Fig. 14, we have to restrict its items of producing rules. The program offers two choices: One is to begin at one target item; the other is to end at one target item. The customer loss model offered in the thesis utilizes the latter choice. We focus on the target item 'Termination' as the ending rule, so that we can know what activities happened before the termination.

In order to compare the executing efficiency between these two algorithms, we utilize four different supports 0.1, 0.2, 0.3, 0.4, and examine them under the condition of

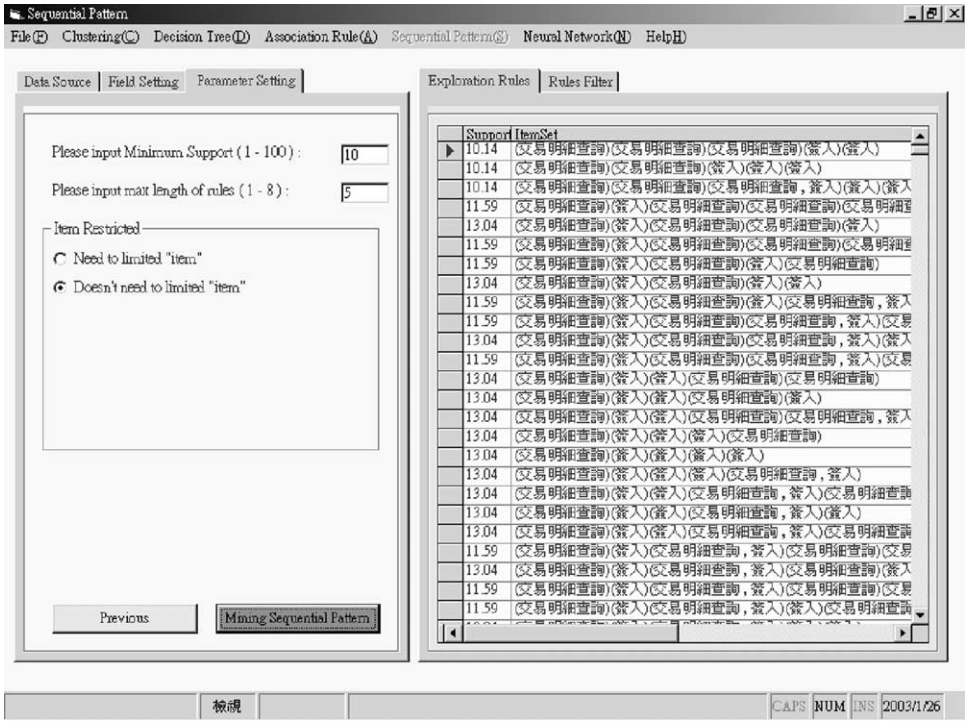


Fig. 13. The executing frame of apriori algorithm.

allowing 5 as the max ruling length. The result was shown in Fig. 15. Compare with these two algorithms, in the same minimum support 0.1 and different data set, the result was shown in Fig. 16. It indicates that the executing efficiency of the Goal-oriented algorithm is evidently better than that of the Apriori algorithm, and it is more apparently as there are more activities and numbers of item and the minimum support comes to less. The numbers of executing efficiency between these two algorithms can up to multi-ten times difference.

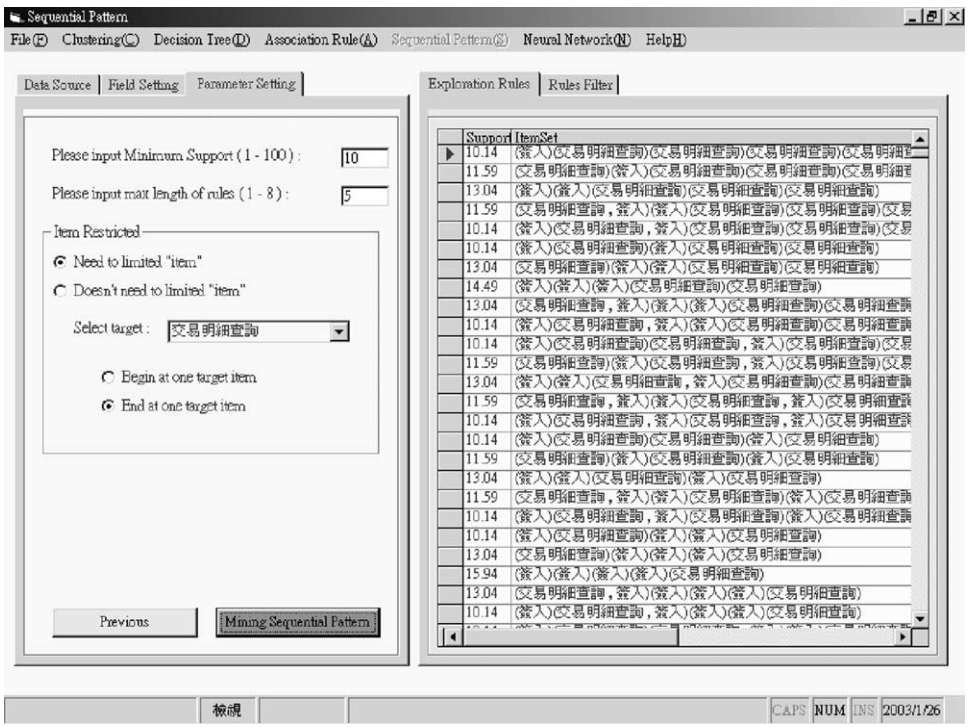


Fig. 14. The executing frame of goal-oriented algorithm.

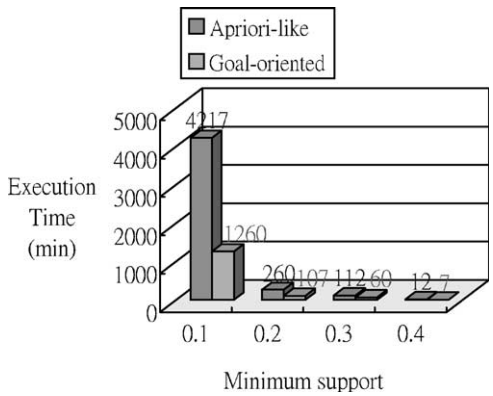


Fig. 15. Execution time (same data set and different minimum support).

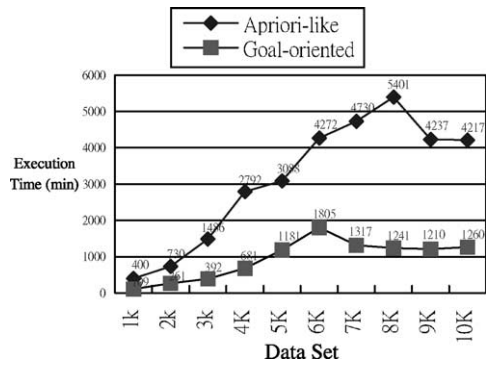


Fig. 16. Execution time (same minimum support and different data set).

As in Fig. 17, which shows the numbers of rules these two algorithms generated, the Apriori algorithm produces more rules though, it makes users not so easy to read and check. Compare with these two algorithms, in the same minimum support 0.1 and different data set, the result was shown in Fig. 18. As in Fig. 13, if we do not restrict an item, it contains all the rules, including the rules we concern and those we do not concern. It is difficult for us to read. Contrarily, the numbers of rules that are produced

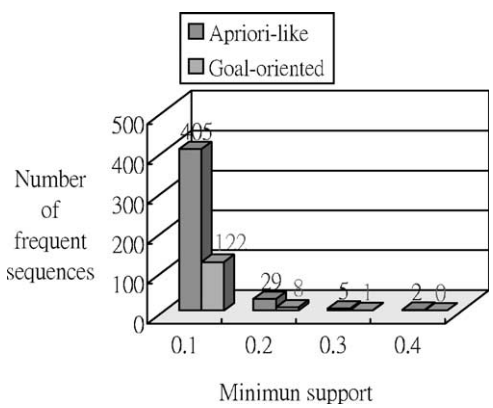


Fig. 17. Number of rules (same data set and different minimum support).

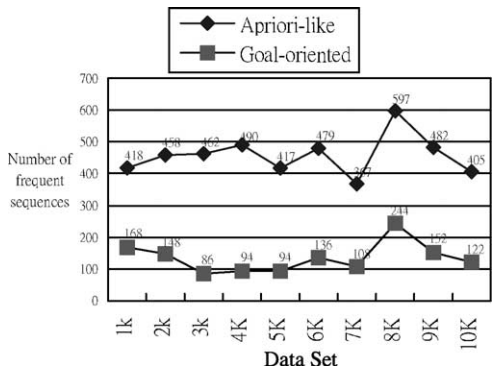


Fig. 18. Number of rules (same minimum support and different data set).

by the Goal-oriented sequential patterns focus on the rules relate to the target item, and therefore have a better readability. In Fig. 14, we restrict each rule to end up with the description ‘transaction inquiry’, and therefore, the last description of each rule is ‘transaction inquiry’. It gets us easy to read.

4.3. The customer loss analysis result

After we use Goal-oriented sequential patterns algorithm to process the customer loss analysis, the rule of loss will be produced as shown in Fig. 19; take the model login in failure 2 as an example, 77.894% of the customers who accord with the situation do not enter into the system over 30 days, that is to say that it is very possible for them to stop using the service. Login in failure 2 indicates the meaning of entering incorrect PIN two times, and due to the agreements of the banking services, the bank will disable the service if three consecutive attempts are made to sign on with incorrect PIN, such a regulation causes the customers choose to terminate the services. According to the loss analysis data, we can find out the customers of this type, if the bank agent who is responsible for the customer service can take the initiative in contacting the customer and solving the problem in time, then the efficiency of constant use of the services is increased by a big margin, it also applies to other customers in other models. While the bank is blindly promoting its sales to get more new users, it does not realize that the existing customers are losing gradually. For the bank, such a gradual loss is larger than the benefits of getting new customers. Fig. 19 shows the main reasons for the customer loss after the analysis. We list the first 20 rules in proportion to the termination rate, and we give the customer center the name lists of the customers who are classified by the causes. The center then can trace the losing customers, and because the customer activity is subject to change, we can use the system to add new information and renew the rules of loss as we find out the rate of the causes of these rules is decreased by a big margin.

Rule	Rate of Interruption	Transaction Behavior	Secondary Transaction Behavior
1	77.89%	Login in failure 2	
2	77.09%	Login in failure 3	
3	65.85%	Login in failure 4	
4	53.79%	Login in failure	Change password
5	50.06%	Change password	
6	44.95%	Login in failure	
7	41.50%	Redemption of fund 2a	
8	41.50%	Redemption of fund 1	Redemption of fund 2a
9	39.43%	Fund account inquiry	Redemption of fund 2a
10	39.43%	Fund account inquiry	Redemption of fund 1
11	35.67%	Redemption of fund 1	
12	33.48%	Fund account inquiry	Redemption of fund 1

Fig. 19. Main losing mode.

5. Conclusion

In this paper, in order to set up the customer-losing pattern, we present a novel method to find out the activities of the losing customers by the excavating sequential pattern algorithm. Moreover, we propose another new algorithm called the Global-oriented algorithm in search of specific goal items. The most distinct difference between this algorithm and the traditional one is that we use the concept of reversed sequence. We firstly reverse the original sequence, and then sort the items we concerns and put them to the top of the whole sequence, next, comparing goal items to sift the rules which are irrelevant to the goal items. The algorithm makes the rule-finding efficiency increased by a big margin. Besides, the rules concluded by the algorithm are simple, which result from the goal items we concern; and the rules, therefore, are easier to read. Finally, we attempt to find out the main reasons why the customers terminate the services through the Goal-oriented algorithm. The results show that up to 80% of the terminated users, the cause of the termination is just because they repeated entered incorrect PIN for two times. The customers stop using the network banking services or become a static account in consequence. After obtaining such information, with some timely solutions, businesses can effectively prevent the loss from happening.

References

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large database. *Proceedings of the ACM SIGMOD international conference on management of data*, 207–216.

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th international conference on very large data bases*, 478–499.

Agrawal, R., & Srikant, R. (1995). Mining sequential patterns. *Proceedings of the 11th international conference on data engineering*, 3–14.

Gronroos, C. (1984). A service quality model and its marketing implication. *European Journal of Marketing*, *18*(4), 36–44.

Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & Hsu, M.-C. (2000a). Freespan: frequent pattern-projected sequential pattern mining. *Proceedings of the international conference of data mining and knowledge discovery*, 355–359.

Han, J., Pei, J., & Yin, Y. (2000b). Mining frequent patterns without candidate generation. *Proceedings of the ACM SIGMOD conference on management of data*, 241–250.

Masseglia, F., Cathala, F., & Poncelet, P. (1998). The PSP approach for mining sequential patterns. *Proceedings of the second european symposium on principles of data mining and knowledge discovery*, *1510*, 176–184.

Ozden, B., Ramaswamy, S., & Silberschatz, A. (1998). Cyclic association rules. *In Proceedings of the 14th international conference on data engineering*, 412–421.

Park, J. S., Chen, M. S., & Yu, P. S. (1995). An effective hash based algorithm for mining association rules. *Proceedings of the ACM SIGMOD conference on management of data*, 175–186.

Pei, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., & Hsu, M.-C. (2001). Prefixspan: mining sequential patterns efficiently by prefix projected pattern growth. *Proceedings of the international conference of data engineering*, 215–224.

Reich, B. H., & Benbasat, I (2003). An empirical investigation of factors influencing the success of customer-oriented strategic system. *Information Systems Research, 1*(3) 325–347

Reichheld, F. (1996). *The loyalty effect: The hidden force behind growth, profit and lasting value*. Cambridge: Harvard Business School Press.

Srikant, R., & Agrawal, R. (1995). Mining generalized association rules. *Proceedings of the 21th international conference on very large data bases*, 407–419.

Tseng, F. C., & Hsu, C. C. (2001). Generating frequent patterns with the frequent pattern list. *Proceedings of the asia pacific conference of data mining and knowledge discovery*, 376–386.