

行政院國家科學委員會專題研究計畫 成果報告

從交易資料庫中探勘時序性序列型樣

計畫類別：個別型計畫

計畫編號：NSC93-2213-E-032-012-

執行期間：93年08月01日至94年07月31日

執行單位：淡江大學資訊工程研究所

計畫主持人：蔣定安

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 94 年 9 月 9 日

Mining Interval Sequential Patterns

Ding-An Chiang,¹ Shao-Lun Lee,^{2,*} Chun-Chi Chen,¹ Ming-Hua Wang,³

¹*Department of Information Engineering, Tamkang University, Tan-Shui, Taipei, Taiwan, R.O.C.*

²*Department of Information Management, Oriental Institute of Technology, Pan-Chiao, Taipei, Taiwan, R.O.C.*

³*Department of Information Management, Nanya Institute of Technology, Jung-Li, Taoyuan, Taiwan, R.O.C.*

The main task of the *mining sequential patterns* is to analyze transaction database of a company in order to find out the priorities of items that most customers take when consuming. In this paper, we propose a new method—*ISP Algorithm*. With this method, we can not only find out the order of consumer items of each customer, but also offer the periodic interval of consumer items of each customer. What the difference, comparing with other previous *periodic association rules*, is that the period the algorithm provides is not the repeated purchases in a regular time, but the possible repurchases within a certain time frame. The algorithm utilizes transaction time interval of individual customers and that of all the customers to find out when and who will buy goods, and what items of goods they will buy.

1. INTRODUCTION

In the twenty-first century, business owners deeply understand the meaning of Charles Robert Darwin's famous saying "Survival of the fittest". The competition among businesses became white hot. We have "upgraded" from traditional information war to all-round intelligence war. The strategy "providing the most information to users" in the past to help users to make decisions no longer copes with unpredictable fluctuations on the customer market. How to handle the target market more precisely and promote enterprises' competitive intelligence becomes an indispensable weapon for the enterprises.

Moreover, as the coming of the new knowledge era, everything needs efficiency and effect. So does to a successful business. To make a decision, decision makers in the enterprise have to analyze lots of reports and information (both in the enterprise and out of the market). However, they need equivalent knowledge and experiences to handle because information analysis is a complicated and difficult work. When collecting information, one may need to gather it from many possible origins. According to different origins, the management information is analyzed by different computer programs that there are designed by programmers. The problem is how to find out unobvious but valuable information from database and provide it to decision makers for reference.

*Author to whom all correspondence should be addressed: e-mail: sllee@mail.oit.edu.tw.

The sequential pattern can be divided into Sequential Procurement² and Cyclic Procurement⁸ 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. It focuses on searching items or merchandise that customers keep purchasing, and then suggests business to promote new products of the kind to the customers. Lawrence et al. use Association Rule and Clustering to analyze past consuming data of customers, hoping to group customers properly. From analysis, business can understand consuming forms of each customer and then recommend suitable goods to customers who are interested in to raise the response rate of commercial merchandise.⁷ Kitts et al. use the concept of cross sale to calculate number and possibility of the customers who have bought certain goods and will buy some other merchandise afterwards. According to the purchase probability, then one can decide items of goods that recommended to the customers.⁶ 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. Generally, when examining the time window of the related rules that is found in the transactional database, we may find some rules appear at some certain time of a year, maybe in a few months or couples of weeks, and show their periodicities. Ozden et al. proposed Sequential Algorithm and Interleaved Algorithm to find out cyclic association rules.⁸ All the methods introduced above are helpful to decision makers to find out valuable information and it has practical values in the marketing management.

This paper uses the concept mentioned above, combining data mining skills and fundamentals of statistics to develop a new algorithm which we called Interval Sequential Pattern—ISP Algorithm. It cannot only find out the order of purchase items of each customer, but also the time interval of the periodic purchase items. By using the time interval information provided by this sequential pattern, marketing people can sale suitable products to suitable customers at the suitable sale time and reach enterprise competitive intelligence.^{4,10}

The ISP algorithm that we proposed in this paper is different with these two type methods mentioned above. The difference is order patterns only show the priority order of purchased items. For example, Rule A B C [0.20] represents the purchase order of the customer who, after buying item A, will buy item B and then item C in a row. There are 20% of the customers meet such purchase order, but it cannot show the time interval of the sequential pattern when happening, that is, the traditional sequential pattern cannot show how much time it takes from the first item A to the last item C; therefore, we cannot know when is the suitable time to recommend product C after the last time the customer bought product A.

What different with cyclic association rule is the sequential pattern that is found is not merely periodic purchase items; therefore, we named the algorithm Interval Sequential Pattern instead of emphasize whether or not it is a periodic purchase. The algorithm uses the fundamentals of statistics to calculate the shortest time interval of purchase, the longest time interval of purchase, and the average time interval of purchase in each rule. It helps marketing people to recommend suitable merchandise to customers at the suitable time. Besides, ISP Algorithm separately

calculates the shortest time interval, the longest time interval, and the average time interval in the transactional sequential pattern of each individual along with those of overall customers. It shows the different consuming behavior between individuals and the whole, and separates and groups different customers further. So we can use the rule we found to analyze the model of customers' consuming behavior and to help us to make a further profitable decision. According to this result, we can compare the different consuming behaviors between individual and overall customers, and provide more information to help marketing people to make decisions in advance. Enterprises that get such information can be master of the likes of customers more precisely and reach target marketing.

2. RELATED RESEARCHES

As point the out by Agrawal et al,² the associated definition of the Mining Sequential Pattern, a sequence has multi-itemsets, and is arranged by some certain order. We represent a sequence as $\langle s_1, s_2, \dots, s_n \rangle$, and s_i is an itemset in it. If two sequences $\langle a_1, a_2, \dots, a_n \rangle$ and $\langle b_1, b_2, \dots, b_m \rangle$ have an order integer $i_1 < i_2 < \dots < i_n, 1 \leq i_k \leq m$, and make $a_1 \subseteq b_{i_1}, \dots, a_n \subseteq b_{i_n}$, then we called sequence $\langle b_1, b_2, \dots, b_m \rangle$ include sequence $\langle a_1, a_2, \dots, a_n \rangle$, and at the same time, $\langle a_1, a_2, \dots, a_n \rangle$ is $\langle b_1, b_2, \dots, b_m \rangle$ and the length of it is subsequence of n . In sequence set, if a sequence s is the maximal sequence, then sequence s is not included by other sequences. A customer sequence is the whole itemsets of a customer, the sequence that is sorted by the order of priority of transaction time. If sequence s is included by a customer sequence, then we say it as the customer supports sequence s . A support for a sequence is the number of the overall customers that support the sequence. If the support of a sequence fits the minimal support, then this sequence is called frequent sequence. On the contrary, it is called non-frequent sequence. The length of a sequence consists of the number of itemset of that sequence. We stand k -sequence for the sequence of the length k , and a frequent sequence of length k is indicated as k -frequent sequence.

In general, there are two phases to create the sequential pattern : Setting large itemset and establishing sequential pattern rules by large itemset.^{5,9} As the association rule,^{1,3} 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 is applied to find the large itemset. Whereas in phase II, there are five phases to generate sequential pattern rule from the large itemset.

1. 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 new database after sorting incrementally by time is showed in Table I.

Table I. The customer sequence is ordered by increasing transaction time

Transaction Id	Transaction Date	Customer Id	Items
10	2001/10/02	ID0001	10
11	2001/11/06	ID0001	30
12	2001/11/15	ID0001	40,80
13	2001/11/20	ID0001	20,30
15	2001/12/02	ID0001	30,80
17	2001/12/04	ID0001	40
07	2001/07/22	ID0002	20
08	2001/08/06	ID0002	30
09	2001/08/16	ID0002	40,70,80

14	2001/11/25	ID0002	30
16	2001/12/20	ID0002	40,70
18	2001/12/25	ID0002	30,70
19	2001/12/28	ID0002	40
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

2. Large Itemset Phase: Find out all the large itemsets and denote each large itemset with a specific symbol.
3. Transformation Phase: Delete the non-large itemsets from the raw transaction database. That is, the items not in the large itemset from the raw transaction database are deleted and rearranged data by customers.
4. Sequential Phase: Delete the sequences that each contains two items from the data. The sequences can be generated by algorithm similar to Apriori.
5. Maximal Phase: Find out the maximal sequences within the frequent 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.

3. INTERVAL SEQUENTIAL PATTERNS

3.1 Calculating transaction time interval

In the original transactional database, we need to discuss different transactional behavior model of each customer separately; therefore, according to transaction sequence of each customer, we have to calculate separately the shortest time interval, the longest time interval and the average time interval of each customer. By doing this, we can provide marketing people the most suitable time to sale that merchandise. The following are the associated definitions and the formula for calculation:

Definition 3.1: Transaction time interval

Let a customer sequence be $C = \langle C_1, C_2, \dots, C_i, \dots, C_m \rangle$, and the transaction dates be $D = \langle D_1, D_2, \dots, D_i, \dots, D_m \rangle$, $1 \leq i \leq m$. If there exists a sequential pattern $SP = \langle S_1, S_2, \dots, S_n \rangle$, such that $S_1 = C_{j+1}, S_2 = C_{j+2}, \dots, S_n = C_{j+n}$, where $0 \leq j \leq (m-n)$ and $1 \leq n \leq m$, then the transaction time interval $TI = D_{j+n} - D_{j+1}$. On the contrary, if the sequential pattern does not exist, then the transaction time interval $TI = 0$.

Take Table I as an example, the whole transaction time interval of sequential pattern $\langle (30)(40) \rangle$ in the customer sequence $\langle (10)(30)(40,80)(20,30)(30,80)(40) \rangle$ of the customer no. ID0001 are 9 days (2001/11/15 - 2001/11/06) and 34 days (2001/12/24 - 2001/11/20); with the

same method, the whole transaction time interval of sequential pattern $\langle (30)(40) \rangle$ in the customer sequence $\langle (20) (30) (40,70,80) (30) (40,70)(30,70)(40) \rangle$ of the customer no. ID0002 are 10 days (2001/08/16 - 2001/08/06), 25 days (2001/12/20 - 2001/11/25), and 3 days (2001/12/28 - 2001/12/25); however, in customer no. ID0003, the sequential pattern $\langle (30)(40) \rangle$ is not included in the sequential pattern $\langle (30,50)(60) \rangle$, so the transaction time interval is 0 day; by using the same method, in customer no. ID0004, because the sequential pattern $\langle (30)(40) \rangle$ is not included in the sequential pattern $\langle (10,20)(90) \rangle$, the transaction time interval is 0 day; the sequential pattern $\langle (30)(40) \rangle$ in the customer sequence $\langle (30)(40,70) \rangle$ of the customer no. ID0005 is 22 days (2001/02/25 - 2001/02/03).

In addition, from definition 3.1 we can calculate transaction time interval of individual customers, the following are the associated definitions:

Definition 3.2: Transaction time interval of individual customers

- The shortest transaction time interval of individual customers: If sequential pattern $SP = \langle S_1, S_2, \dots, S_n \rangle$ appears in customer sequence $C = \langle C_1, C_2, \dots, C_m \rangle$ one or more than one time, then the shortest transaction time interval of the individual customer is the minimum of the transaction time interval (if it contains non-zero values, the minimum of the transaction time interval must be greater than zero) that appeared the sequential pattern.
- The longest transaction time interval of individual customers: If sequential pattern $SP = \langle S_1, S_2, \dots, S_n \rangle$ appears in customer sequence $C = \langle C_1, C_2, \dots, C_m \rangle$ one or more than one time, then the longest transaction time interval of the individual customer is the maximum of the transaction time interval that appeared the sequential pattern.
- The average transaction time interval of individual customers: If sequential pattern $SP = \langle S_1, S_2, \dots, S_n \rangle$ appears in customer sequence $C = \langle C_1, C_2, \dots, C_m \rangle$ one or more than one time, then the average transaction time interval of the individual customer is the average of the transaction time interval that appeared the sequential pattern.

Take Table I as an example, the overall transaction time interval of sequential pattern $\langle (30)(40) \rangle$ in customer sequential pattern of customer no. ID0001 is 9 days and 34 days; therefore, the shortest time interval $= \min(9, 34) = 9$ days, the longest time interval $= \max(9, 34) = 34$ days, and the average time interval $= (9 + 34) / 2 = 21.5$ days. the overall transaction time interval in customer sequential pattern of customer no. ID0002 is 10 days, 25 days and 3 days; therefore, the shortest time interval $= \min(10, 25, 3) = 3$ days, the longest time interval $= \max(10, 25, 3) = 25$ days, and the average time interval $= (10 + 25 + 3) / 3 = 12.67$ days; the overall transaction time interval in customer sequential pattern of customer no. ID0003 is 0 day; therefore, the shortest time interval $= \min(0) = 0$ day, the longest time interval $= \max(0) = 0$ days, and the average time interval $= 0$ day; the overall transaction time interval in customer sequential pattern of customer no. ID0004 is 0 day;

therefore, the shortest time interval= $\min(0)=0$ day, the longest time interval= $\max(0)=0$ days, and the average time interval= 0 day; the overall transaction time interval in customer sequential pattern of customer no. ID0005 is 22 days; therefore, the shortest time interval= $\min(22)=22$ days, the longest time interval= $\max(22)=22$ days, and the average time interval= $22/1=22$ days.

Different with individual customer transactions, the overall customer transaction discusses the transaction behavior models of the whole customers, and these overall customer transaction models are generated and formed a common rule from transaction behavior models of individual customers, which have the same transaction behavior models as those of overall customers. In other words, the rule that generated by overall customers' transaction sequential pattern must larger than one of the minimal support, that is, it appears in the rule of overall customer transaction sequential pattern only if some certain amount of individual customers that have such transaction behavior model. The following are the associated definitions:

Definition 3.3: Transaction time interval of overall customers

- The shortest transaction time interval of overall customers: The shortest transaction time interval of overall customers of sequential pattern $SP=\langle S_1, S_2, \dots, S_n \rangle$ is the minimum of the shortest transaction time interval of all individual customers (if it contains non-zero values, the minimum of the shortest transaction time interval must be greater than zero).
- The longest transaction time interval of overall customers: The longest transaction time interval of overall customers of sequential pattern $SP=\langle S_1, S_2, \dots, S_n \rangle$ is the maximum of the longest transaction time interval of all individual customers.
- The average transaction time interval of overall customers: The average transaction time interval of overall customers of sequential pattern $SP=\langle S_1, S_2, \dots, S_n \rangle$ is the weighted mean of the average transaction time interval of all individual customers.

As in Table I, the shortest transaction time interval of overall customers in sequential pattern $\langle(30)(40)\rangle$ is equal to $\min(9,3,0,0,22)=3$ day, the longest transaction time interval of overall customers is equal to $\max(34,25,0,0,22)=34$ days, the average transaction time interval of overall customers is equal to $(21.5*2+12.67*3+0*0+0*0+22*1)/(2+3+0+0+1)=17.17$ days.

3.2 ISP Algorithm

As shown in section 2, the complicated work in the problems of mining sequential pattern is at the fourth step, the sequence phase, that is, find out all the frequent sequence. Therefore, our method emphasizes on how to calculate the shortest time interval, the longest time interval, the average time interval, number of appearance, and the support of each sequential pattern in each customer sequence in the phase of sequence. As for other work phases, we follow the steps in

section 2 without changing the steps very much.

The principle of processing the algorithm we proposed is constantly repeated the calculation for many times. Each round includes three steps and the final goal of each round is to calculate the frequent sequence which is longer 1 in length than the last time, and it ends until the ISP judge one ending value is true. The corresponding algorithm is showing in Figure 1.

```

FS1={Frequent 1-sequences};
For (k=2; FSk-1?∅; k++) do begin
    CSk = Apriori-All (FSk-1);
    ISP_individual_customers(CSk)
    ISP_overall_customers(CSk)
End

Apriori-All()
{
Insert into CSk
Select p.item1, p.item2, ..., p.itemk-1, q.itemk-1
From FSk-1 p, FSk-1 q
Where p.item1 = q.item1, ..., p.itemk-2 = q.itemk-2, p.itemk-1 < q.itemk-1;
Insert into CSk
Select p.item1, p.item2, ..., q.itemk-1, p.itemk-1
From FSk-1 p, FSk-1 q
Where p.item1 = q.item1, ..., p.itemk-2 = q.itemk-2, p.itemk-1 < q.itemk-1;
For itemsets i ∈ CSk do
    For all (k-1)-subsets s of i do
        If (s ∉ FSk-1) then
            Delete i from CSk
}

ISP_individual_customers()
{
min_t = 32767, max_t = 0, count = 0, total = 0, avg_t = 0

While there are k-candidate sequence s=<s1,s2,...,sk> generated do

begin

    While there are every customer sequence c=<c1,c2,...,cm> do

        begin

            Select Count(c.*) As n
            From c,s
            Where c.item = s1;

            Foreach n do

                begin

                    Select c.timej+1 As t(s1), c.timej+2 As t(s2),..., c.timej+k As t(sk)
                    From c,s
                    Where c.itemj+1 = s1, c.itemj+2 = s2, ..., c.itemj+k = sk;

                    If t(sk) - t(s1) < min_t then min_t = t(sk) - t(s1)

                    If t(sk) - t(s1) > max_t then max_t = t(sk) - t(s1)

                    total = total + t(sk) - t(s1)

                End

            End

        End

    End

}

```

```

If total > 0 then
    count = count + 1
Else
    Exit
End
End
If count > 0 then avg_t = total / count
End
}

ISP_overall_customers()
{
t_min_t = 32767, t_max_t = 0, t_count = 0, t_total = 0, t_avg_t = 0
While there are k-candidate sequence s = <s1, s2, ..., sk> generated do
begin
    While there are every customer sequence c = <c1, c2, ..., cm> do
    begin
        Select c.shortesttime As min_t, c.longesttime As max_t, c.averagetime As ave_t, c.count As count
        From c, s
        Where c.itemj+1 = s1, c.itemj+2 = s2, ..., c.itemj+k = sk;
        If min_t < t_min_t then t_min_t = min_t
        If max_t > t_max_t then t_max_t = max_t
        t_total = t_total + ave_t * count
        temp = temp + count
        If count > 0 then t_count = t_count + 1
    End
    If temp > 0 then t_avg_t = t_total / temp
    If t_count >= Min_support then FSk = FSk ∪ s
End
}

```

Figure 1. ISP Algorithm

The processing of each step is explained in detail as following: Let FS_{*k*} be the set of *k*-frequent sequence, FS be the total set of the all frequent sequence, and CS_{*k*} be the set of *k*-candidate sequence.

- Step 1: Producing Candidate sequence CS_{*k*} (*k*=2,3,4...)

The first step of each round must generate the candidate sequence of this round based on (*k*-1)-frequent sequence set of the last round.

In the first time, the length 2 that is the candidate sequence set of CS_k ($k=2$) is generated by: For any two items s_1 and s_2 both belong to FS_1 , and $s_1 \neq s_2$, it generates an outcome that $\langle s_1, s_2 \rangle$ and $\langle s_2, s_1 \rangle$ belong to CS_2 .

The way that CS_k ($k>2$) is generated in other times is: For any two $(k-1)$ -frequent sequences $x_1 = \langle a_1, \dots, a_{k-2}, a_{k-1} \rangle$ and $y_1 = \langle b_1, \dots, b_{k-2}, b_{k-1} \rangle$, If $a_1 = b_1, \dots, a_{k-2} = b_{k-2}$, we can generate two k -frequent sequences $\langle a_1, \dots, a_{k-2}, a_{k-1}, b_{k-1} \rangle$ and $\langle a_1, \dots, a_{k-2}, b_{k-1}, a_{k-1} \rangle$.

The action follows step one is to check whether CS_k is an empty set. If CS_k is an empty set, ISP stops; this is because there is no more frequent sequence being generated in this time. It means frequent sequences of all lengths are all found. Otherwise, we delete those that are not possible to become candidate sequences of k -frequent sequence. In other words, if a k -sequence wants to become a frequent sequence, all the subsequences of the length $k-1$ must be frequent sequences.

As for every s in each candidate sequence CS_k , each s subsequence has to be checked individually to see if it appears in FS_{k-1} . If not, we shall delete s from CS_k .

After deleting, if CS_k becomes an empty set, ISP stops, or we will enter to the next step.

- Step 2: Generating the shortest, the longest, and the average transactional time interval of individual customers and its frequency.

Notation:

TTIC is the abbreviation of “transactional time interval of individual customers”

min_t : the shortest *TTIC*

max_t : the longest *TTIC*

count : the frequency of transaction of individual customers

total : the sum of *TTIC*

avg_t : the average of *TTIC*

First step is to calculate the number n , where n represents the number of the first item s_1 of each k -candidate sequence $s = \langle s_1, s_2, \dots, s_k \rangle$ shown on every customer sequence $c = \langle c_1, c_2, \dots, c_m \rangle$. Thus by using a loop to search n , the position and transaction time $t(s_1)$ of item s_1 in customer sequence can be found.

In addition, using the position and the time to find forwardly, to check whether k -candidate sequence s_2, \dots, s_k appear in customer sequence c , if it does, then record the position and transaction time $t(s_k)$ of item s_k in the customer sequence, and calculate the transaction time interval $t(s_k) - t(s_1)$ of individual customers from s_1 to s_k .

To compare the *TTIC* with the shortest *TTIC* *min_t*, if $t(s_k) - t(s_1) < min_t$, then $min_t = t(s_k) - t(s_1)$.

To compare the *TTIC* with the longest *TTIC* *max_t*, if $t(s_k) - t(s_1) > max_t$, then $max_t = t(s_k) - t(s_1)$.

The number of transaction of individual customer $count=count+1$, and the sum of $TTIIC\ total=total+t(s_k)-t(s_1)$.

After finding n times in this loop, we calculate average $TTIIC\ avg_t=total / count$.

- Step 3: Generating the shortest time interval, the longest time interval, the average time interval, the support and frequent sequence FS_k ($k=2,3,4,\dots$) of the transaction of the whole customers.

Notation:

$TTIWC$ is the abbreviation of “transactional time interval of whole customers”

t_min_t : the shortest $TTIWC$

t_max_t : the longest $TTIWC$

t_count : the number of individual customer present in the k -candidate sequence

t_total : the sum of $TTIWC$

t_avg_t : the average of $TTIWC$

First, select the shortest $TTIIC\ min_t$, the longest $TTIIC\ max_t$, transaction counts and average $TTIIC\ ave_t$ of every k -candidate sequence s_i (i represent the i k -candidate sequence) in every customer sequence c_j (j represent the j customer sequence) form the transaction database.

To compare the shortest $TTIIC\ min_t$ with the shortest $TTIWC\ t_min_t$, if $min_t < t_min_t$, then $t_min_t=min_t$.

To compare the longest $TTIIC\ max_t$ with the longest $TTIWC\ t_max_t$, if $max_t > t_max_t$, then $t_max_t=max_t$.

After that, calculating the average of $TTIWC\ t_avg_t$ which equals to the weighted mean of the average of $TTIIC$.

After comparing every k -candidate sequence s_i with every individual customer sequence c_j , the shortest $TTIWC\ t_min_t$, the longest $TTIWC\ t_max_t$, and average $TTIWC\ t_avg_t$ can be generated.

One can tell how many customer sequences in which count is larger than 1 by the number of count that appears in each individual customer sequence c_j .

By calculating the support of k -candidate sequence, FS_k can finally be generated by comparing the minimum support.

As mentioned before, the complicated task of ISP algorithm is in the fourth step, the sequence phase. Therefore, the efficiency of sequence phase affects the executing time of examining the whole interval sequential pattern. In calculating the complexity of time interval, supposing the biggest large itemset we found includes k items, we need to scan the database $(k+1)$ time, and have to calculate time interval of every customer sequence every time we scan the database. The time we spent here at this part is $O(CS_k) \times O(n)$, where CS_k is number of every

k-candidate sequence, n is the number of time that the first item of the candidate sequence appeared in a customer sequence. Therefore, the maximum of the whole time complexity is $O(k \times t \times r \times n)$, where k is the length of biggest large itemset, t is number of transaction in the database, r is the total of rules being generated, n is number of a item that only shows on a customer sequence. The associated efficiency analysis will be discussed in detail in the next section.

4. THE EXPERIMENT RESULTS

To examine the executing efficiency of the Interval sequential pattern algorithm, we separately design some experiments to examine our method, and compare it with AprioriAll algorithm. We will introduce experimental environments, including the experiment desktop, the practicing program language and information resources. In section 4.1, we will introduce the experiment results. Moreover, in section 4.2, we will compare the executing efficiency of the Interval Sequential Pattern Algorithm.

The following are the associated experimental environments:

1. Experimental Platform
 - CPU : Pentium IV 2.4GMHz
 - Memory : 1GB RAM , 80 G bytes 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 retail business in Taiwan, total more than 1.5 millions transaction data, and more than 18 thousand customer' s data from 2000 to 2001 years.

4.1 Interval Sequential Pattern Algorithm versus AprioriAll Algorithm

The executing frame of AprioriAll Algorithm is shown in Figure 2, this method only provides the purchase priority of the all customers and number of people who have done such transaction behavior (support), but on the executing frame of ISP Algorithm as shown in Figure 3, the algorithm we proposed obviously can provide more information. It knows not only the support of every sequential pattern, but also the shortest, the longest, the average time interval of the sequence in the whole customer transaction sequence. Besides, if we want to know what kinds

of customers have the same purchase order, we can see the further detailed information as in Figure 4. In this figure, we can get the shortest time interval, the longest time interval, the average time interval, and number of transaction of the sequential pattern in the transaction sequence of the individual customers. According to the date, marketing people know in what opportunity they can promote intended merchandise to the intended customers.

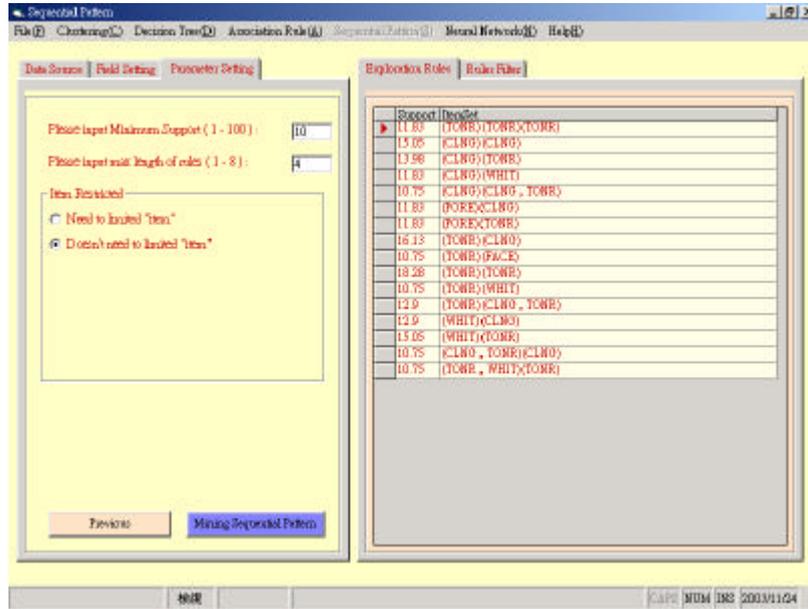


Figure 2. The executing frame of AprioriAll algorithm

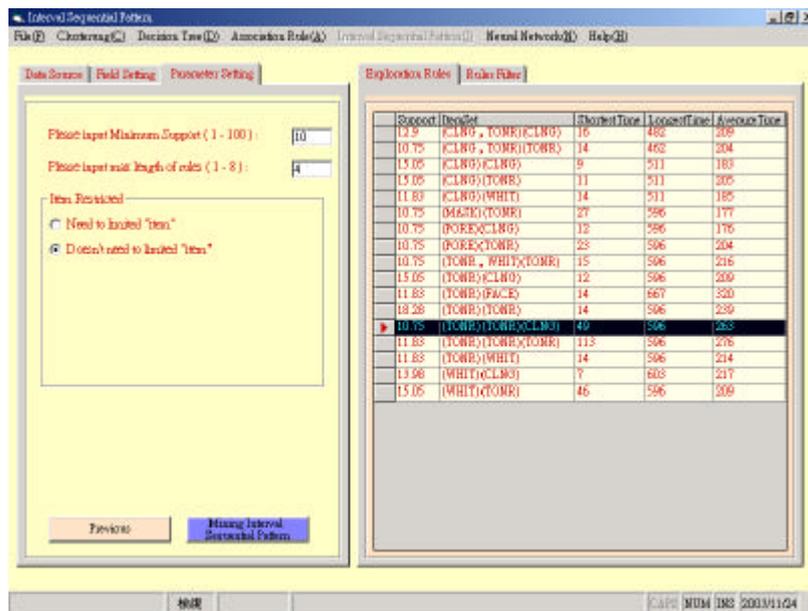


Figure 3. The executing frame of ISP algorithm

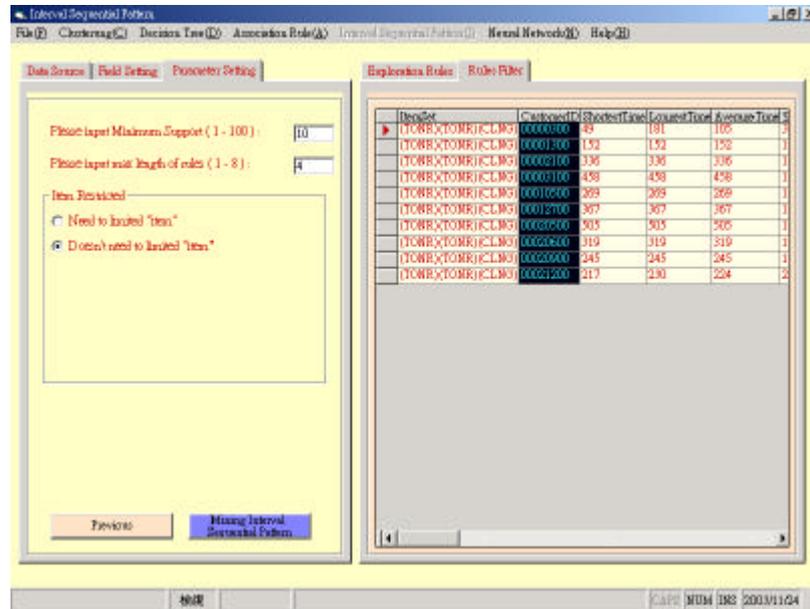


Figure 4. The executing frame of individual customers detail information

Take Figure 3 as an example, the marked place represents (TONR) (TONR) (CLNG) rule, which means after buying (TONR), the customers in that group will buy (TONR) again, and then buying (CLNG). 10.75% of the customers have such purchase order and among these customers, some bought (TONR) again and (CLNG) within 49 days, and it will not longer than 596 days. The average purchase time interval of the general customers who fit this purchase order is 263 days. If we want to inquire how many customers have this (TONR) (TONR) (CLNG) priority order in advance, we can click the rule and browse the detailed information. As shown in Figure 4, there are ten customers 0000300, 00001300, 00002100, 00003100, 00010500, 00012700, 00020500, 00020600, 00020900, and 00021200 fit the (TONR) (TONR) (CLNG) priority order. Therefore, we use the last purchase dates, the last time that each of the ten customers bought (CLNG), to calculate how long it has passed, and then comparing the shortest, the longest, the average transaction time interval of each customer, and finally, come out the recommended product purchase list of (TONR) and (CLNG) and so on.

4.2 The Efficiency Analysis of Interval Sequential Pattern Algorithm

In order to compare the executing efficiency of these two algorithms, we separately utilize four different supports 0.05, 0.10, 0.15, and 0.20, and examining under the condition of allowing all the maximum length of rule is 4. The result as shown in Figure 5, ISP Algorithm gives more information but has only slight differences when comparing the executing efficiency with AprioriAll algorithm. It is because although the ISP algorithm takes one more circuit time to calculate the time interval of every customer sequence (ISP algorithm, Step 2), but generally, it

only takes $O(n)$ than the AprioriAll algorithm. In the sequence, n is the number of the first item of the sequential pattern that appeared in the customer sequence.

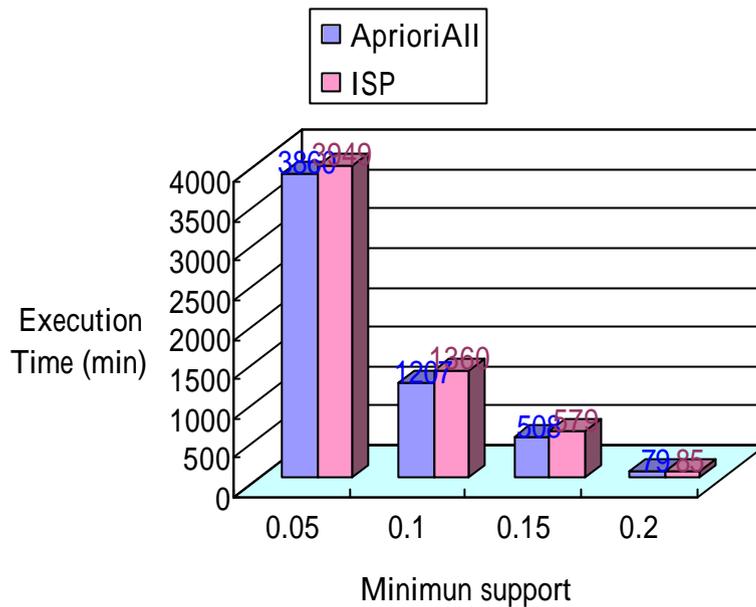


Figure 5. Execution Time (same data set and different minimum support)

5. CONCLUSION

In this paper, we propose a new algorithm called ISP Algorithm. The greatest difference between the ISP Algorithm and AprioriAll Algorithm is ISP Algorithm not only finds out the order of items that customers purchased and the number of people who have the same transactional behavior, but also provides the time interval customers spent from the first item to the last when purchasing. According to the result, we can compare customers' transactional records and predict the items a customer is going to buy (What: we can tell by the items in the sequential pattern), the time he or she will purchase (When: We can tell by the overall transactional intervals of customers), and the person who will follow this purchase order (Who: We can tell by the transactional details of each individual). With this method, the marketing people then can decide (What) commodities should be promoted to which customers (Who) at which timing (When), and it will gradually increase the response rate of the commercial merchandise and reach the marketing goal. Moreover, as to the overall efficiency, through this algorithm, one can obtain more information without spending much more extra time.

References

1. Agrawal R, Srikant R. Fast algorithms for mining association rules. Proceedings of the 20th International Conference on very large Data Bases 1994;478-499.

2. Agrawal R, Srikant R. Mining Sequential Patterns. Proc. 11th Int'l Conf. Data Eng 1995;3-14.
3. Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large database. Proceedings of the ACM SIGMOD International Conference on Management of Data 1993;207-216.
4. Fayyad UM, Shapiro GP, Smyth P. The KDD Process for extracting useful knowledge from volumes of data. Communication of the ACM 1996;39(11):27-34.
5. Han J, Pei J, Yin Y. Mining Frequent Patterns without Candidate Generation. Proceeding of the ACM SIGMOD Conference on management of Data 2000;241-250.
6. Kitts B, Freed D, Vrieze M. Cross-sell: a fast promotion-tunable customer- item recommendation method based on conditionally independent probabilities. Proceedings of the ACM 6th Int'l Conf. on Knowledge Discovery and Data Mining, 2000;437-446.
7. Lawrence RD, Almasi GS, Kotlyar V, Viveros MS, Duri SS. Personalization of supermarket product recommendations. Data Mining and Knowledge Discovery 2001;5(1/2):11-32.
8. Ozden B, Ramaswamy S, Silberschatz A. Cyclic association rules. In Proceedings of the 14th International Conference on Data Engineering 1998;412-421.
9. Tseng FC, Hsu CC. Generating frequent patterns with the frequent pattern list. Proceeding of the Asia Pacific Conference of Data Mining and Knowledge Discovery 2001;376-386.
10. Tyson KWM. Competitive knowledge development: reengineering competitive intelligence for maximum success. Competitive Intelligence Review 1995;6(4):14-21.