การค้นหากฎความสัมพันธ์จากข้อมูลอนุกรมเวลา

กิตติพงศ์ วะระทรัพย์ [1] และ ชาคริดา นุกูลกิจ [2*]

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี บางมด ทุ่งครุ กรุงเทพฯ 10140

บทคัดย่อ

เทคนิคในการค้นหากฎความสัมพันธ์ที่ใช้ในงานวิจัยนี้ เป็นเทคนิคการทำเหมืองข้อมูลที่นำเสนอขึ้นเพื่อใช้ในการค้นหากฎความสัมพันธ์ของข้อมูลอนุกรมเวลาโดยเฉพาะ เนื่องจากข้อมูลอนุกรมเวลาเป็นข้อมูลที่ต้องคำนึงถึงลำดับก่อน-หลังของข้อมูลด้วย ซึ่งต่างจากข้อมูลทรานแซกชั่นโดยทั่วไปที่ไม่ต้องคำนึงถึงเรื่องดังกล่าว ก่อนหน้านี้การค้นหากฎความสัมพันธ์จากข้อมูลอนุกรมเวลามักจะสนใจเฉพาะข้อมูลชนิดไม่ต่อเนื่อง ซึ่งอัลกอริทึมที่นำเสนอในงานวิจัยนี้สามารถทำงานได้ทั้งข้อมูลชนิดต่อเนื่องและไม่ต่อเนื่อง โดยกฎความสัมพันธ์ที่หาได้จะอยู่ในรูปแบบ $X \xrightarrow{t} Y$ ซึ่งหมายถึง " ถ้าพบรูปแบบ X แล้วภายในระยะเวลา t จะพบรูปแบบ Y "

คำสำคัญ : ข้อมูลอนุกรมเวลา / กฎความสัมพันธ์ / ข้อมูลเรียงลำดับ / การทำเหมืองข้อมูล

** Corresponding author: E-mail: chakarida@it.kmutt.ac.th*

[1] นักศึกษาปริญญาโท คณะเทคโนโลยีสารสนเทศ

[2] อาจารย์ คณะเทคโนโลยีสารสนเทศ

Discovery Association Rules in Time Series Data

Kittipong Warasup[1] and Chakarida Nukoolkit [2*]
King Mongkut's University of Technology Thonburi, Bangmod, Toongkru, Bangkok 10140

Abstract

Rule discovery from time series data is a data mining technique that tries to find relationships of sequential data. Finding association rules from time series data is different from finding such rules in traditional data because time series data is orderly data with a sequence that must be preserved. Many researchers have proposed many methods of analyzing and mining time series data, but most of them did not focus on finding association rules, and the data used in their experimentations were discretized symbols. In fact, many situations collect data in continuous numeration time series. In this paper, we propose a novel technique to find association rules from time series data. Our technique can analyze either the numerical time series or the symbolic time series and show the resulting rules as $X \xrightarrow{t} Y$, which means that the group of pattern *Y* should occur within time *t* when the group of pattern *X* occurs.

Keywords : Time Series / Association Rules / Sequential Data / Data Mining

* *Corresponding author: E-mail: chakarida@it.kmutt.ac.th*

[1] *Graduate Student, School of Information Technology.*

[2] *Lecturer, School of Information Technology.*

1. Introduction

Finding association rules is a technique of data mining that discovers the co-occurrence pattern of the data. This technique is widely used to solve many problems in a variety of fields, such as predicting what items a customer might buy together, or predicting a stock price pattern. The first example is the market basket analysis, in which input data are items purchased by customers, but input data of the second example are time series data. Finding association rules from these two types of input data requires different approaches, because the second data is orderly data with a sequence that must be preserved, but the first data is a set of items purchased together in a point-of-sale transaction. Thus, the order of how each item appears in the same transaction does not affect the resulting association rules. Moreover, some situations collect time series data in continuous numeration, which is different from symbolic data. So we propose a new method of finding association rules in numerical time series. However, our method also works with symbolic time series by just skipping the data transformation step that we suggest for the numerical time series case.

Little work has been done on finding association rules from time series data [1], and some work has been done in mining sequential data which is symbolic [2, 3, 4] so the problem we introduce is finding association rules from time series data that are in numerical format. When numerical sequences are given as input, each sequence is assigned a transaction number and put in a database like a customer transaction in market basket analysis, but there is a difference in that the order of items in the sequence we consider must be preserved, unlike in market basket analysis in which a transaction that contains A, C is viewed the same as another transaction which contains C, A. In the time series case, we set up a new term called "item-series" to describe those items.

Once we get input data, numerical sequences are transformed to symbolic sequences by SAX [5]. Then by using our proposed technique we will discover the rules, which are in the form of $X \xrightarrow{t} Y$. The meaning of this format is that the group of pattern Y should occur within time t when the groups of pattern X occurs. In our experimentation we show the rules obtained by our technique and interpret their meaning. Furthermore, the experimentation indicates the effect of parameters used in the transformation step.

We organize the rest of this paper as follows. Section 2 refers to the previous works that relate to our work. Section 3 is a brief discussion of background on time series data transformation and association rules. Section 4 focuses on our new approach in finding association rules from time series data. Section 5 is about the experimentation of our technique on three datasets. Section 6 summarizes the idea and experimental results along with suggestions for future work.

2. Related Works

In [6], finding similarity between time series, which is a common task of time series analysis, is discussed. Many techniques such as Dynamics Time Warping (DTW) and Longest Common Subsequence (LCS) are mentioned. The work in [7] summarizes the techniques of data transformation such as Discrete Fourier Transform (DFT) and Discrete Wavelet Transform (DWT). Paper [8] reviews the symbolic analysis and shows an example of the successful application of symbolization. A data transformation method called symbolic aggregate approximation (SAX), which will be covered briefly in the next section, is shown in full detail in [5]. The works of [9, 10] focus on traditional association rules. [9] proposes mining association rules in a large database. [10] surveys and compares the techniques of association rule finding.

In [1], clustering methods are used for discretization, and the rules are found in $X \xrightarrow{t} Y$ format. However, both the left hand side and the right hand side of the rule, which are X and Y respectively, are single-item patterns which differ from our rules that allow patterns with multiple items. The works in [2, 3, 4] discuss mining sequential data. [2] proposes mining sequential patterns from a database of customer transactions which consists of ID, time, and item purchased. The work of [4] proposes the algorithm for discovery of frequent groups of patterns called episodes.

3. Background

3.1 Transformation

Time series data transformation is a part of data preprocessing before mining. This preprocessing step can help reduce analysis time due to the dimension reduction. This step also reduces the effect of noise in data sets. In our experimentation, we use a data transformation technique called symbolic aggregate approximation (SAX) [5] to transform our numerical time series to symbolic format. However, other algorithms, which can transform continuous data to symbolic format are also applicable. SAX can reduce the existing data size n down to size w (where $w < n$). SAX preprocesses the raw numerical time series by first applying Piecewise Aggregate Approximation (PAA) [11] to the raw data. This step divides time series data into w intervals and finds an average for each interval. Then SAX applies a discretization step to the result of the previous PAA step. SAX's discretization step is based on the Gaussian distribution assumption. An example of the final transformed time series resulting from SAX is shown in Fig. 1.

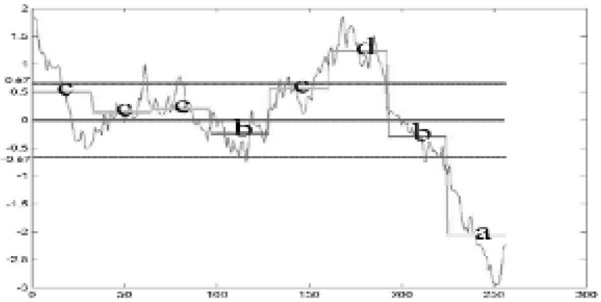


Fig. 1 A symbolic time series after SAX data transformation.

3.2 Rules discovery

The main goal of extraction of association rules is finding the interesting rules which are in the format X⟶Y. The interestingness of the rules can be determined from two measures: support and confidence. The higher the support and the higher the confidence level of the rules, the better rules they are. Thus, in association rule finding, we are looking for rules with high supports and high confidences. Common practice usually involves setting up a minimum threshold value for an acceptable level of a support value and a confidence value. Then the association rule mining algorithm will try to find association rules which have supports and confidences higher than the specified minimum limit.

Let

D represents a database

T_j represents a transaction j in a database D where $T_j \in D$

i represents one item.

I represents a group of items called item-set $I = \{i_1, i_2, \dots, i_n\}$

Thus T_j contains item i in the transaction j where $T_j \subseteq I$ and if we have item-set X and Y where $X, Y \subseteq Tj \subseteq I$ and $X \cap Y = \varnothing$ we can represent an association rule as $X \longrightarrow Y$

The above rule means when a customer buys items in X then he or she will buy items in Y. This association rule's interesting level can be measured from two values: support and confidence, which can be calculated from equations (1) and (2) respectively.

If $X \subseteq T_j$, then we can conclude that the transaction $T_j \in D$ supports item-set $X \subseteq I$ where the ratio of the number of transactions T_j which supports item-set X compared to the number of transactions in the whole database D is called the support of X. This is written as the equation (1) below.

$$\sup(X) = \frac{\left|\left\{T_j \in D \middle| X \subseteq T_j\right\}\right|}{|D|} \quad \textbf{(1)}$$

The confidence of an association rule is defined as the equation (2) below.

$$conf(X \rightarrow Y) = \frac{\sup(X \rightarrow Y)}{\sup(X)} \qquad \textbf{(2)}$$

where $\sup(X \longrightarrow Y)$ is obtained from $\sup(X \longrightarrow Y) = \sup(X \cap Y)$.

4. Association Rules for Sequential Data

4.1 Formal problem

The database we consider is a collection of sequential data. Each transaction of the database contains one sequential data that consists of two parameters: value (s_j) and time (t_j). Suppose that there are m transactions in the database, T_1 to T_m. The database and its transactions can be written as follows.

$$D = \{T_1, T_2, \ldots, T_i, \ldots, T_m\}$$
$$T_i = ([s_1,t_1], [s_2,t_2], \ldots, [s_j,t_j], \ldots, [s_n,t_n])_i$$
$$\text{where } t_{j+1} > t_j$$

However, the data we refer to is in constant interval; that means $t_{j+1} - t_j = t_j - t_{j-1}$ for all j where $1 < j < n$, then the transaction T_i can be easily written as $T_i = (s_1, s_2, \ldots, s_j, \ldots, s_n)_i$

Table 1 The input database.

Transaction No.	Item-series
1	$(s_1\ s_2\ \ldots\ s_j\ \ldots\ s_n)_1$
2	$(s_1\ s_2\ \ldots\ s_j\ \ldots\ s_n)_2$
⋮	⋮
i	$(s_1\ s_2\ \ldots\ s_j\ \ldots\ s_n)_i$
⋮	⋮
m	$(s_1\ s_2\ \ldots\ s_j\ \ldots\ s_n)_m$

Each s_j shown in Table 1 is a symbol that refers to the original data. The domain of the symbol is a set $A=\{a_1, a_2, \ldots, a_k, \ldots, a_q\}$ where a_k is a single alphabet and q is a number of alphabet that relate the pattern of data.

4.2 Item-series

Items in transactions in a market basket analysis database are non-orderly data, but items in transactions of our time series database are orderly data. To illustrate that, consider the following transaction.

Table 2 Example of transaction in database.

Transaction No.	Item
1	*a b c b*

In market basket analysis, items in transactions are purchased products, so it is not different to say that the customer buys item *'a'* and *'b'* or buy item *'b'* and *'a'*. On the other hand, the transactions we consider refer to the events or patterns of the data involving time, so we have to know which item comes first.

Moreover, in Table 2 there are two items of *'b'*. Market basket analysis considers those as one item and the transaction can be re-written to *'a b c'*. For sequential transaction the item *'b'* must be left in correct order so the transaction cannot be reduced like the previous case.

The term *"item-series"* is setup to distinguish those cases. An item-series having one item such as (s_j) is called a 1-item-series, then the (s_j, s_{j+1}) is called a 2-item-series and so on. We can shortly write *n*-item-series as $\langle s_j, s_{j+n-1}\rangle$ and we can also write subscript *i* as $\langle s_j, s_{j+n-1}\rangle_i$ to identify the i^{th} transaction.

Now, consider subsequent transactions. Subsequence $X = Sub(T_i,j,t)$ is an item-series of the i^{th} transaction that follow by equation (3).

$$Sub(T_i,j,t) = \langle s_j, s_x\rangle_i \ ; \ x = min(n,j+t-1) \qquad \textbf{(3)}$$

where *n* is the length of the sequence.

For example, as illustrated in Table 3, $Sub(T_1,2,4) = \langle s_2, s_5\rangle_1$ and $Sub(T_3,3,4) = \langle s_3, s_5\rangle_3$.

Table 3 Database example.

Transaction No.	Item-series
1	s_1 s_2 s_3 s_4 s_5
2	s_1 s_2 s_3 s_4 s_5
3	s_1 s_2 s_3 s_4 s_5
4	s_1 s_2 s_3 s_4 s_5

4.3 Discovering large item-series

Like traditional association rules, our step of finding association rules can be decomposed into two sub-steps. First is the step of discovering all large item-series. Second is generating rules from those large item-series. In this section, we focus on the first one.

The problem for now is how to discover the large item-series. We first introduce the technique of finding frequency of item-series then we show how to calculate support of item-series. To complete this step, the algorithm broken down generating item-series that have their support higher than minimum support is discussed.

We define a new function called *within*(X) to find the set of subsequence X in the database. Suppose item-series X is a subsequence $(a_{p1}, a_{p2}, \ldots, a_{pr}, \ldots, a_{pt-1})$ then the function can be defined as

$$within(X) = \left\{ \langle s_j, s_{j+t-1} \rangle \mid \forall u \left(j \le u \le j+t-1 \wedge s_u = a_{p_{u-j+1}} \right) \right\} \quad \textbf{(4)}$$

where $p_r \in \{1, 2, \ldots, q\}$ and a_{pr} is the $(p_r)^{th}$ element of A.

For example, let database D, shown in Table 4, have four transactions that consist of five symbols each. There are three alphabets forming a sequence in transaction.

Table 4 Database.

Transaction No.	Item-series
1	*b b a c a*
2	*b a c a c*
3	*b b b a c*
4	*b b a c a*

Let X be an item-series (b b a). Finding *within(X)* using equation (4) get the output as $\{\langle s_1,s_3\rangle_1$, $\langle s_2,s_4\rangle_3$, $\langle s_1,s_3\rangle_4\}$ and also the frequency of item-series X is $|within(X)| = 3$

Next, we find all possible subsequence length t which is allSub(t) = {Sub(Ti,j,t) | $\forall$i,j(i$\in$ {1, …, m} $\wedge$ j$\in${1, …,n}) } where m is number of transaction in database and n is length of each transaction. From above example all subsequence that have length 3 is

allSub(3) ={Sub(T1,1,3), Sub(T1,2,3),…, Sub(T1,5,3),
Sub(T2,1,3), Sub(T2,2,3),…, Sub(T2,5,3),
Sub(T3,1,3), Sub(T3,2,3),…, Sub(T3,5,3),
Sub(T4,1,3), Sub(T4,2,3),…, Sub(T4,5,3)}

Thus, the number of all possible sequences that have length 3 is |*allSub*(3)| = 20.

Now, we can find the support of item-series by

$$\sup(X) = \frac{|within(X)|}{|allSub(t)|} \tag{5}$$

where t is the length of subsequence X

So support of item-series (b b a) is $\frac{3}{20} = 15\%$

The algorithm finding large item-series is shown as follows.

Algorithm Discovery__large__item__series

Input : Database D, minSupport

Output : Large Item-series L

L_1 = {large 1-Item-series}

for k=2 , $L_{k-1} \neq \varnothing$, k++

C_k = genCandidate(L_{k-1})

C_k.count = within(C_k)

$L_k = \{c \in C_k \mid c.\text{count} \geq \text{minSupport}\}$

end

$L = U_k L_k$

Using table 4 as database (D) and setting minimum support at 20%, we get a set of large 1-item-series which is { {a}, {b}, {c} } (See support of 1-item-series in table 5). Then we can find next large item-series, L_2 , L_3 and so on by following above algorithm. Finally, unite all large item-series.

Algorithm genCandidate

Input : L_{k-1}

Output : C_k

for all $l \in L_{k-1}$

if $l_i(2)=l_j(1)$, $l_i(3)=l_j(2)$,…,$l_i(k-1)=l_j(k-2)$

insert l_i+$l_j(k-2)$ into C_k

end

end

From table 6 showing support of all 2-item-series, we get large 2-item-series,L_2 , { {ac}, {ba}, {bb}}. Using genCandidate algorithm, for example, if we consider on ‘ac’ and ‘ba’, we see that last letter ‘c’ from ‘ac’ doesn’ t math first latter ‘b’ from ‘ba’ but last latter ‘a’ from ‘ba’ math first latter ‘a’ from ‘ac’. So we get ‘bac’. Continuing the same process, we get candidate item-series, C_3.

Algorithm within

Input : C_k

Output : C_k.count

for all transaction in D

Generate list of member in T (CT)

for all $c \in C_k$

c.count = c.count + number of c in C_T

end

end

We can find support of each item-series in candidate item-series, C_k, by following 'within' algorithm to pick item-series having support more than minimum support and generate next large item-series.

Table 7 shows the support of 3-item-series. It illustrates that only one item-series, which is *'bac'* has support higher than minimum support, so this is included in large item-series.

Table 5 Support of 1-item-series.

Item-series	Support
a	35%
b	40%
c	25%

Table 6 Support of 2-item-series.

Item-series	Support
aa	0%
ab	0%
ac	25%
ba	20%
bb	20%
bc	0%
ca	15%
cb	0%
cc	0%

Table 7 Support of 3-item-series.

Item-series	Support
bac	20%
bba	15%
bbb	5%

Thus, the set of large item-series (L) is {{*a*}, {*b*}, {*c*}, {*ac*}, {*ba*}, {*bb*}, {*bac*}}

Notice that the algorithm does not generate *'aca'* as 3-item-series despite the support of *'ac'* is 25% (from Table 6). This is because *'ca'*, a part of *'aca'*, has support less than minimum support.

4.4 Generating rules

Association rules generated by our algorithm are the pattern $X \xrightarrow{t} Y$ where X and Y are large item-series such as those found in section 0 and t is time interval. The rules which have a confidence higher than minimum confidence are acceptable. Our rules are generated from the set of large item-series. Let $X, Y \in L$, we define set A as a set of subsequence that follows the item-series X.

$$A = \left\{ \langle s_j, s_{j+t-1} \rangle \mid \exists u \exists v \left(v = j-1 \wedge u < v \rightarrow \langle s_u, s_v \rangle \in within(X) \right) \right\}$$

Also, set B is defined as a set of subsequence that contains item-series Y.

$$B = \left\{ \langle s_j, s_{j+t-1} \rangle \mid \exists u \exists v \left(j \le u \le v \le j+t-1 \rightarrow \langle s_u, s_v \rangle \in within(Y) \right) \right\}$$

The confidence of the rule $X \xrightarrow{t} Y$ is number of subsequence length t that contains item-series Y and follows the item-series X.

$$conf(X \xrightarrow{t} Y) = \left| \left\{ \langle s_j, s_{j+t-1} \rangle \mid \langle s_j, s_{j+t-1} \rangle \in A \cap B \right\} \right|$$

Consider parameter t. Suppose $t_1 < t_2$ then the number of set B with t_2 is higher than the number of set B with t_1 because the longer t the more possible it is to find the subsequence. Thus, if the rule $X \xrightarrow{t_2} Y$ is not accepted, the rule $X \xrightarrow{t_1} Y$ can not be accepted either.

Algorithm genRules

Input : L, minConfidence

Output : Rules

for all $l \in L$

    for t=n-1, 1, t---

```
                rule.count = calculateRuleConf
                if rule.count ≥ minConfidence
                        generate rule l1 --t--> l2
                else
                        break to next l
        end
end
```

Table 8 Example of the rules.

LHS	RHS	t	conf
b	c	3	88%
b	ac	3	88%
bac	a	2	75%
b	c	2	50%
b	bac	4	50%

5. Experimental Results

We test the proposed algorithm with three actual data sets in a variety of domains. Closed price of stocks from Stock Exchange of Thailand contains more than 2,000 points of data [12]. The price of 20 companies is collected from August 1996 to July 2004. The data from Thai meteorological department consists of average daytime temperature and relative humidity [13]. Last data set is the positions of a moon orbiting around Neptune [14]. This data set consists of latitude and longitude of the moon collected every two minutes.

Our experiment focuses on finding association rules from numerical time series data. We start by transforming the raw numerical inputs into symbolic representation. During this step we experiment with adjusting of system parameters such as the number of segments of data (*k*) and the number of alphabets used to represent data (α), the minimum support level, and the minimum confidence level. The results are shown in Fig. 2-5.

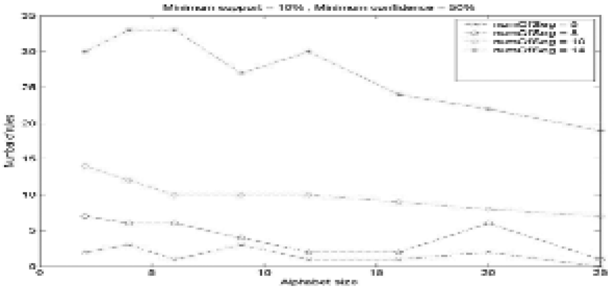


Fig. 2 Graph between α and the number of discovered association rules with 10% support and 50% confidence.

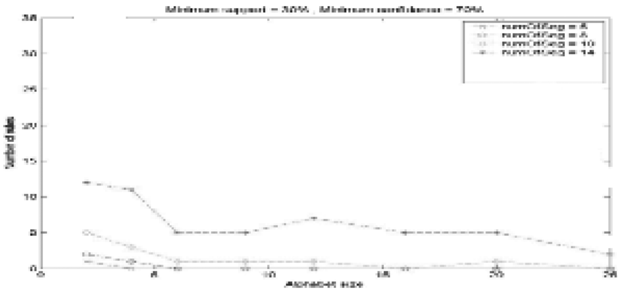


Fig. 3 Graph between α and the number of discovered association rules with 30% support and 70% confidence.

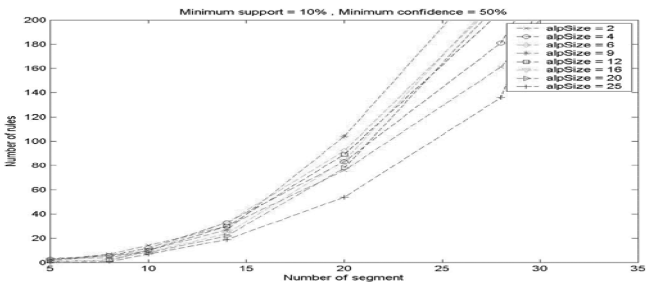


Fig. 4 Graph between the number of segments *k* and the number of discovered association rules with 10% support and 50% confidence.

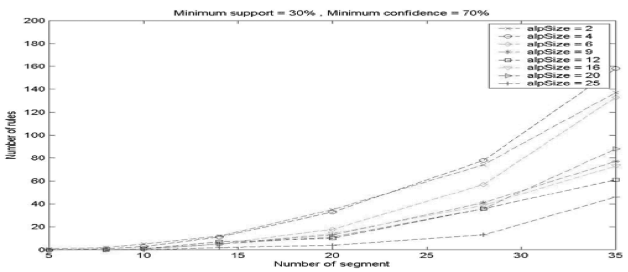
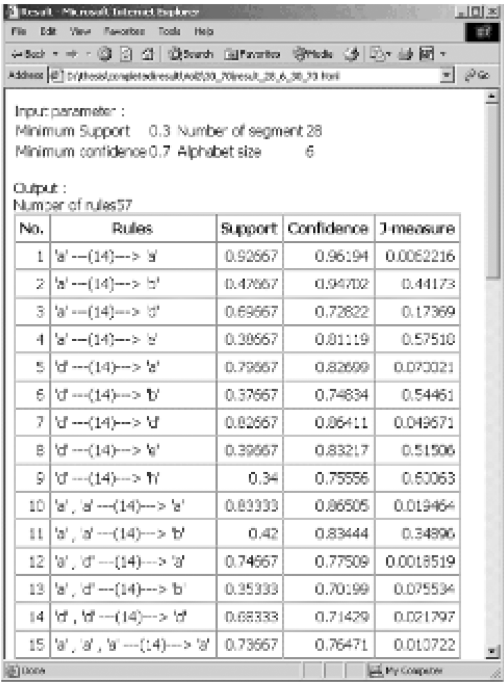


Fig. 5 Graph between the number of segments, *k* and the number of discovered association rules with 30% support and 70% confidence.

In addition, the resulting association rules from the proposed method are displayed in an html file in Fig. 6.



No.	Rules	Support	Confidence	J-measure
1	'a' ---(14)---> 'a'	0.92667	0.96194	0.0062216
2	'a' ---(14)---> 'b'	0.47667	0.94702	0.44173
3	'a' ---(14)---> 'd'	0.69667	0.72822	0.17369
4	'a' ---(14)---> 'e'	0.39667	0.81119	0.57510
5	'd' ---(14)---> 'a'	0.79667	0.82699	0.070021
6	'd' ---(14)---> 'b'	0.37667	0.74834	0.54461
7	'd' ---(14)---> 'd'	0.82667	0.86411	0.049671
8	'd' ---(14)---> 'e'	0.39667	0.83217	0.51506
9	'd' ---(14)---> 'f'	0.34	0.75556	0.60063
10	'a' , 'a' ---(14)---> 'a'	0.83333	0.86505	0.019464
11	'a' , 'a' ---(14)---> 'b'	0.42	0.83444	0.34896
12	'a' , 'd' ---(14)---> 'a'	0.74667	0.77509	0.0018519
13	'a' , 'd' ---(14)---> 'b'	0.35333	0.70199	0.075534
14	'd' , 'd' ---(14)---> 'd'	0.68333	0.71429	0.021797
15	'a' , 'a' , 'a' ---(14)---> 'a'	0.73667	0.76471	0.010722

Fig. 6 The extracted association rules.

The resulting association rules are in the format $X \xrightarrow{t} Y$ where *X* and *Y* are a pattern which should occur in the time interval *t*. An example of the rule interpretation can be explained as follows:

Let a time series contain continuous daily data points for 2,000 days. If we divide this data into 20 segments (*k* = 20), each data segment will contain $\frac{2000}{20} = 100$ data points. If the resulting association rule is $Z, W \xrightarrow{5} X$, we will conclude that if the pattern *Z* followed immediately by *W* occurs, then within 5 time units or 500 days (5 X 100) pattern *X* should occur.

Let's assume the same time series with 200 segments. Each segment contains $\frac{2000}{200} = 10$ data points. If the resulting association rule is $X \xrightarrow{2} Z$, this means that if the pattern *X* occurs, within 2 time units or 20 days (2 X 10) pattern *Z* should occur.

From the illustrated examples, if the number of segments (*k*) is large, the resulting association rules are suitable for long-term prediction. However, if the number of segments (*k*) is small, the resulting association rules are suitable for short-term prediction. However, from our experiment, the smaller the *k* value, the more processing time we need in association rule extraction. Thus, the user must take the requirement of time series analysis into account whether short-term or long-term prediction is of his/her interest. If the *k* parameter reflects too short an interval, the computation may be unnecessarily time consuming.

6. Conclusion

This work proposes a new method in finding association rules in sequential data such as numerical time series. We suggest using a symbolic data transformation called SAX in transforming the raw numerical time series into symbolic representation, and then mining the transformed data for association rules which meet the minimum support and minimum confidence with our new approach. Our algorithm defines the steps of extraction of large item-series and the method of generating association rules from them. This results in finding many interesting association rules in the form of

$X \xrightarrow{t} Y$ where *X* and *Y* are the pattern of Item-series which occur and *t* is the time interval in which the pattern *Y* should occur after the pattern *X* has occurred.

If the user has the raw time series data in symbolic format, our proposed method of association rule finding still applies. The users can skip the data transformation step from numerical to symbolic time series data, which we suggest.

In the future, we see the opportunity to improve the proposed algorithm into an online-type algorithm, which will be more useful to all real-time streaming-type applications.

7. References

1. Das, G., Lin, K., and Mannila, H., 1998, "Rule Discovery from Time Series", *Proceeding of the 4th International Conference of Knowledge Discovery and Data Mining*, pp. 16-22.

2. Agrawal, R. and Srikant, R., 1995, "Mining Sequential Patterns", *Proceeding of the 11th International Conference on Data Engineering,* Taipei, Taiwan, pp. 3-14.

3. Hoppner, F. and Klawonn, F., 2002, "Finding Informative Rules in Interval Sequences", *Intelligent Data Analysis - International Journal,* Vol. 6(3), pp. 237-256.

4. Mannila, H., Toivonen, H., and Verkamo, A. I., 1997, "Discovery of Frequent Episodes in Event Sequences", *Proceedings of the 1st Data Mining and Knowledge Discovery,* pp. 259-289.

5. Lin, J., Keogh, E., Lonardi, S., and Chiu, B., 2003, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms", *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery,* San Diego, California, pp. 2-11.

6. Das, G., Gunopulos, D., and Mannila, H., 1996, "Finding Similar Time Series", *Proceeding of KDD97*, pp. 88-100.

7. Keogh, E., 2001, “A Tutorial on Indexing and Mining Time Series Data”, *The 2001 IEEE International Conference on Data Mining*, November 29 - December 2, 2001, San Jose, California, USA.

8. Daw, C. S., Finney, C.E.A., and Tracy, E. R., 2002, “A Review of Symbolic Analysis of Experimental Data”, July 22, 2002.

9. Agrawal, R., Imielinski, T., and Swami, A., 1993, “Mining Association Rules between Sets of Items in Large Database”, *Proceedings of the 1993 ACM SIGMOD Conference,* May 1993, Washington D.C., USA, pp. 207-216.

10. Hipp, J., Guntzer, U., and Nakhaeizadeh, G., June 2000, “Algorithms for Association Rule Mining - A General Survey and Comparison”, *ACM SIGKDD Explorations Newsletter*, Vol. 2, pp. 58-64.

11. Lin, J., Keogh, E., Patel, P., and Lonardi, S., 2002. “Finding Motifs in Time Series”, *2nd Workshop on Temporal Data Mining and the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* July 23-26, Edmonton, Alberta, Canada, pp. 53-68.

12. KIM ENG Securities (Thailand) Public Company Limited, [Online], Available: http://www.kimeng.co.th., [2004, September 24].

13. The Meteorological Department of Thailand, [2004, April 1]

14. Institute of Geophysics and Planetary Physics, [Online], Available: http://www.igpp.ucla.edu, [2004, June 1].