Applied Soft Computing 12 (2012) 536–542

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Fuzzy data mining for time-series data

Chun-Hao Chen[a], Tzung-Pei Hong[b,c,*], Vincent S. Tseng[d]

[a] *Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan*
[b] *Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan*
[c] *Department of Computer Science and Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan*
[d] *Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan*

ARTICLE INFO

Article history:
Received 18 April 2008
Received in revised form 17 January 2011
Accepted 14 August 2011
Available online 22 August 2011

Keywords:
Association rule
Data mining
Fuzzy set
Sliding window
Time series

ABSTRACT

Time series analysis has always been an important and interesting research field due to its frequent appearance in different applications. In the past, many approaches based on regression, neural networks and other mathematical models were proposed to analyze the time series. In this paper, we attempt to use the data mining technique to analyze time series. Many previous studies on data mining have focused on handling binary-valued data. Time series data, however, are usually quantitative values. We thus extend our previous fuzzy mining approach for handling time-series data to find linguistic association rules. The proposed approach first uses a sliding window to generate continues subsequences from a given time series and then analyzes the fuzzy itemsets from these subsequences. Appropriate post-processing is then performed to remove redundant patterns. Experiments are also made to show the performance of the proposed mining algorithm. Since the final results are represented by linguistic rules, they will be friendlier to human than quantitative representation.

1. Introduction

Time series analysis has always been an important and interesting research field due to its frequent appearance in different applications. Some domains such as bioinformatics [2,13], medical treatment [27] and finance [7] especially emphasize it for making good prediction and decision. A time series is usually composed of lots of data points, each of which represents a value at a certain time. In the past, many approaches based on regression [22], neural networks [16,25,31] and other mathematical models [10] were proposed to analyze the time series. Many previous studies on data mining have focused on handling binary-valued data. Time series data, however, are usually quantitative values, so designing a sophisticated data-mining algorithm able to deal with this type of data presents a challenge to workers in this research field [12,31].

Recently, fuzzy set theory has been used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [20,33]. It was first proposed by Zadeh in 1965 [33]. It is primarily concerned with quantifying and reasoning using natural language in which words can have ambiguous meanings. This can be thought of as an extension of traditional crisp sets, in which each element must either be in or not in a set. The theory has been applied in fields such as manufacturing, engineering, diagnosis, economics, among others [15,20,24,29]. Several fuzzy learning algorithms for inducing rules from given sets of data have been designed and used to good effect with specific domains. As to fuzzy data mining, Hong et al. proposed several fuzzy mining algorithms to mine linguistic association rules from quantitative data [16,19,23]. They transformed each quantitative item into a fuzzy set and used fuzzy operations to find fuzzy rules. The mining results obtained could be smooth due to the fuzzy membership characteristics. Cai et al. proposed a weighted mining approach to reflect different importance to different items [8]. Each item was attached a numerical weight given by users. Weighted supports and weighted confidences were then defined to determine interesting association rules. Yue et al. then extended their concepts to fuzzy item vectors [32].

In this paper, we thus extend our previous approach [16] and propose a fuzzy mining algorithm for time series to find linguistic association rules. The proposed approach first uses a sliding window to generate continues subsequences from a given time series and then analyzes the fuzzy itemsets from these subsequences. Appropriate post-processing is also performed to remove redundant patterns.

The proposed approach thus has two advantages. Firstly, since the final results are represented by linguistic rules, they will be friendlier to human than quantitative representation. Secondly, one problem for association-rule mining approaches is that too many

* Corresponding author at: Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan.
E-mail addresses: chchen@mail.tku.edu.tw (C.-H. Chen), tphong@nuk.edu.tw (T.-P. Hong), tsengsm@mail.ncku.edu.tw (V.S. Tseng).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.08.006

rules may be generated. Through the post-processing procedure in the algorithm, lots of redundant rules can be filtered such that the mined rules can be compact. Users can thus utilize the rules more easily.

The remaining parts of this paper are organized as follows. Some related works and the proposed approach for mining fuzzy association rules on time series are given in Section 2. An example to illustrate the proposed algorithm is described in Section 3. Experiments to demonstrate the performance of the proposed algorithm are stated in Section 4. A discussion is given in Section 5. Conclusions and future works are given in Section 6.

2. Mining fuzzy association rules for time series

Data mining is most commonly used in attempts to induce association rules from transaction data [6]. The goal is to discover important associations among items such that the presence of some items in a transaction will imply the presence of some other items. To achieve this purpose, Agrawal and his co-workers proposed several mining algorithms based on the concept of large itemsets to find association rules in transaction data [3–6].

Das et al. proposed a mining algorithm for time-series analysis [12]. Their approach composed of two phases: time-series discretization and association-rule generation. In the discretization phase, the approach used a clustering method to find basic shapes from time series, and then transformed the time series into a discretized series based on the basic shapes found. In the second phase, an *Apriori*-like method was used to generate association rules. The rules found out in the above way were different from traditional association rules. The rule format was “If *A* occurs, then *B* occurs within time *T*”, which meant the occurrence of *B* was followed by *A* within *T* time units. In addition, Yuan et al. proposed a mining algorithm for discovering quantitative movement pattern from time series. A rule mined out might be like this: “If the rate of exchange of some stock rises 15%, then its closing price may rise 10%”.

In handling time-series data, Song and Chissom proposed a fuzzy stochastic fuzzy time series and its models by assuming its values are fuzzy sets [26]. Chen and Hwang proposed a two-factor time-variant fuzzy time series model to deal with forecasting problems [9]. Au and Chan proposed a fuzzy mining approach to find fuzzy rules for time series classification [1]. Watanabe exploited the Takagi-Sugeno model to build a time-series model [28].

The proposed fuzzy mining algorithm in this paper integrates the fuzzy sets, the *Apriori* mining algorithm, and the time-series concepts to find out appropriate linguistic association rules. It first generates continuous subsequences from the given time series by a sliding window size. The algorithm then uses membership functions to transform data points in each subsequence into fuzzy sets. After transformation, the scalar cardinalities of all linguistic terms are calculated as their fuzzy counts. The mining process then finds fuzzy large itemsets based on the fuzzy counts. A post-processing step is then performed to remover redundant fuzzy large itemsets. The fuzzy association rules are finally discovered from the fuzzy large itemsets. Details of the proposed mining algorithm are described below.

The proposed fuzzy time-series mining algorithm:

INPUT: A time series S with k data points, a set of h membership functions for data values, a predefined minimum support α, a predefined minimum confidence λ, and a sliding-window size w.

OUTPUT: A set of fuzzy association rules with confidence values from S.

STEP 1: Transform the time series S into a set of subsequences $W(S)$ according to the sliding-window size w. That is,

$$W(S) = \{s_p | s_p = (d_p, d_{p+1}, \ldots, d_{p+w-1}), p = 1 \text{ to } k - w + 1\},$$

where d_p is the value of the p-th data point in S.

STEP 2: Transform the j-th (j = 1 to w) quantitative value v_{pj} in each subsequence s_p (p = 1 to $k-w+1$) into a fuzzy set f_{pj} represented as $(f_{pj1}/R_{j1} + f_{pj2}/R_{j2} + \ldots + f_{pjh}/R_{jh})$ using the given membership functions, where R_{jl} is the l-th fuzzy region of the j-th data point in each subsequence, h is the number of fuzzy memberships, and f_{pjl} is v_{pj}'s fuzzy membership value in region R_{jl}. Each R_{jl} is called a fuzzy item.

STEP 3: Calculate the scalar cardinality of each fuzzy item R_{jl} as:

$$count_{jl} = \sum_{p=1}^{k-w+1} f_{pjl}.$$

STEP 4: Collect each fuzzy item to form the candidate 1-itemsets C_1.

STEP 5: Check whether the support value (= $count_{jl}/k - w + 1$) of each R_{jl} ($p \le j \le p + w - 1$ and $1 \le l \le h$) in C_1 is larger than or equal to the predefined minimum support value α. If R_{jl} satisfies the above condition, put it in the set of large 1-itemsets (L_1). That is:

$$L_1 = \{R_{jl} | count_{jl} \ge \alpha, 1 \le j \le p + w - 1 \text{ and } 1 \le l \le h\}.$$

STEP 6: IF L_1 is not null, then do the next step; otherwise, exit the algorithm.

STEP 7: Set r = 1, where r is used to represent the number of fuzzy items in the current itemsets to be processed.

STEP 8: Join the large r-itemsets L_r to generate the candidate $(r+1)$-itemsets C_{r+1} in a way similar to that in the apriori algorithm [6] except that two items generated from the same order of data points in subsequences cannot simultaneously exist in an itemset in C_{r+1}. Restated, the algorithm first joins L_r and L_r under the condition that $r-1$ items in the two itemsets are the same and the other one is different. It then keeps in C_{r+1} the itemsets which have all their sub-itemsets of r items existing in L_r and do not have any two items R_{jp} and R_{jq} of the same j.

STEP 9: Do the following substeps for each newly formed $(r+1)$-itemset I with fuzzy items $(I_1, I_2, \ldots, I_{r+1})$ in C_{r+1}:

(a) Calculate the fuzzy value of I in each subsequence s_p as $f_I^{s_p} = f_{I_1}^{s_p} \Lambda f_{I_2}^{s_p} \Lambda \ldots \Lambda f_{I_{r+1}}^{s_p}$, where $f_{I_j}^{s_p}$ is the membership value of fuzzy item I_j in s_p. If the minimum operator is used for the intersection, then:

$$f_I^{s_p} = \underset{j=1}{\overset{r+1}{Min}} f_{I_j}^{s_p}.$$

(b) Calculate the count of I in all the subsequences as:

$$count_I = \sum_{p=1}^{k-w+1} f_I^{s_p}.$$

(c) If the support (= $count_I/k - w + 1$) of I is larger than or equal to the predefined minimum support valueα, put it in L_{r+1}.

STEP 10: If L_{r+1} is null, then do the next step; otherwise, set $r = r + 1$ and repeat STEPs 8–10.

STEP 11: Shift each large itemsets $(I_1, I_2, \ldots, I_q)$, $q \ge 2$, into $(I'_1, I'_2, \ldots, I'_q)$, such that the fuzzy region R_{jl} in I_1 will become R_{11} in I'_1 and a fuzzy region R_{it} in the other items becomes $R_{(i-j+1)t}$, where R_{jl} is the l-th fuzzy region of the j-th data point in each subsequence.

Table 1
The time series used in this example.

Time series
1, 6, 9, 8, 4, 3, 7, 9, 6, 4, 2, 4, 7, 8, 3

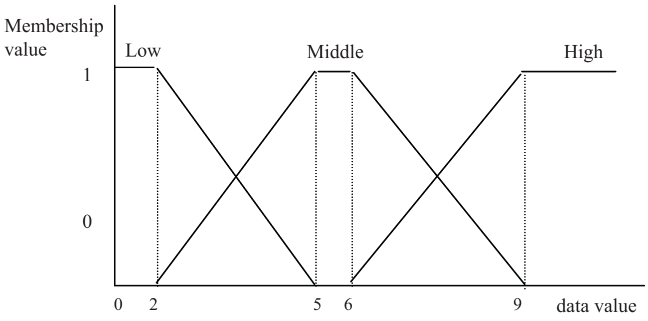

Fig. 1. The membership functions used in this example.

Table 2
The subsequences obtained from the given time series for $w = 5$.

s_p	Subsequence	s_p	Subsequence
s_1	(1, 6, 9, 8, 4)	s_7	(7, 9, 6, 4, 2)
s_2	(6, 9, 8, 4, 3)	s_8	(9, 6, 4, 2, 4)
s_3	(9, 8, 4, 3, 7)	s_9	(6, 4, 2, 4, 7)
s_4	(8, 4, 3, 7, 9)	s_{10}	(4, 2, 4, 7, 8)
s_5	(4, 3, 7, 9, 6)	s_{11}	(2, 4, 7, 8, 3)
s_6	(3, 7, 9, 6, 4)		

STEP 12: Remove redundant large itemsets from the results after STEP 11.

STEP 13: Construct the association rules for each large q-itemset I (after STEP 12) with items $(I_1, I_2, \ldots, I_q)$, $q \geq 2$, using the following substeps:

(a) Form each possible association rule as follows:
$I_1 \Lambda \ldots \Lambda I_{k-1} \Lambda I_{k+1} \Lambda \ldots \Lambda I_q \rightarrow I_k, \quad k = 1 \text{ to } q.$

(b) Calculate the confidence values of all association rules by the following formula:

$$\frac{\sum_{p=1}^{k-w+1} f_I^{s_p}}{\sum_{p=1}^{k-w+1} (f_{I_1}^{s_p} \Lambda \ldots \Lambda f_{I_{k-1}}^{s_p}, f_{I_{k+1}}^{s_p} \Lambda \ldots \Lambda f_{I_q}^{s_p})}.$$

STEP 14: Output the association rules with confidence values larger than or equal to the predefined confidence threshold λ.

Note that in STEP 2, the transformation from data values to fuzzy sets in each subsequence can be simplified by removing the first fuzzy set from the previous subsequence and add the fuzzy set derived from the last data point in the current one.

3. An example

In this section, a simple example is given to show how the proposed algorithm can generate fuzzy association rules from the given time series. Assume the data points in the time series are given as shown in Table 1.

The time series in Table 1 contains 15 data points. Each data point represents a value at a certain time. For example, the second data point in the time series means the value obtained at time 2 is 6.

Assume the fuzzy membership functions for the data values are defined as shown in Fig. 1. There are three fuzzy membership functions, *Low*, *Middle* and *High*, used in this example.

For the time series given in Table 1, the proposed mining algorithm proceeds as follows.

STEP 1: The given time series is first transformed into a set of subsequences according to the predefined window size. Assume the given window size is 5. There are totally $11 (= 15 - 5 + 1)$ subsequences obtained from the time series. The results are shown in Table 2.

STEP 2: The data values in each subsequence are then transformed into fuzzy sets according to the membership functions given in Fig. 1. Take the first value v_{11} (= 1) in the subsequence s_1 as an example. The value "1" is converted into the fuzzy set $((1/A_1 \cdot Low) + (0/A_1 \cdot Middle) + (0/A_1 \cdot High))$, where $A_i{\cdot}term$ is a fuzzy region of the i-th data in the subsequences and is called a fuzzy item. This step is repeated for the other data points and subsequences, with the results shown in Table 3.

STEP 3: The scalar cardinality of each fuzzy item is calculated as its *count* value. Take the fuzzy item $A_1{\cdot}Low$ as an example. Its scalar cardinality = (1 + 0 + 0 + 0 + 0.33 + 0.67 + 0 + 0 + 0 + 0.33 + 1) = 3.33. This step is repeated for the other fuzzy items, with the results shown in the bottom row of Table 3.

STEP 4: All the fuzzy items are collected as the candidate 1-itemsets.

STEP 5: For each fuzzy item, its count is checked against the predefined minimum support value α. Assume in this example, α is set at 30%. Since the support values of $A_1{\cdot}Low$, $A_1{\cdot}Middle$, $A_2{\cdot}Middle$, $A_3{\cdot}Middle$, $A_3{\cdot}High$, $A_4{\cdot}Middle$, $A_5{\cdot}Low$, and $A_5{\cdot}Middle$, are all larger than 30%, these items are thus put in L_1 (Table 4).

STEP 6: Since L_1 is not null, the next step is then done.

STEP 7: Set $r = 1$, where r is the number of fuzzy items in the current itemsets to be processed.

Table 3
The fuzzy sets transformed from the data in Table 2.

s_p	A_1			A_2			A_3			A_4			A_5		
	L	M	H	L	M	H	L	M	H	L	M	H	L	M	H
s_1	1	0	0	0	1	0	0	0	1	0	0.33	0.67	0.33	0.67	0
s_2	0	1	0	0	0	1	0	0.33	0.67	0.33	0.67	0	0.67	0.33	0
s_3	0	0	1	0	0.33	0.67	0.33	0.67	0	0.67	0.33	0	0	0.67	0.33
s_4	0	0.33	0.67	0.33	0.67	0	0.67	0.33	0	0	0.67	0.33	0	0	1
s_5	0.33	0.67	0	0.67	0.33	0	0	0.67	0.33	0	0	1	0	1	0
s_6	0.67	0.33	0	0	0.67	0.33	0	0	1	0	1	0	0.33	0.67	0
s_7	0	0.67	0.33	0	0	1	0	1	0	0.33	0.67	0	1	0	0
s_8	0	0	1	0	1	0	0.33	0.67	0	1	0	0	0.33	0.67	0
s_9	0	1	0	0.33	0.67	0	1	0	0	0.33	0.67	0	0	0.67	0.33
s_{10}	0.33	0.67	0	1	0	0	0.33	0.67	0	0	0.67	0.33	0	0.33	0.67
s_{11}	1	0	0	0.33	0.67	0	0	0.67	0.33	0	0.33	0.67	0.67	0.33	0
Count	3.33	4.67	3	2.66	5.34	3	2.66	5.01	3.33	2.66	5.34	3	3.33	5.34	2.33

Table 4
The set of large 1-itemsets L_1 for this example.

Itemset	Support	Itemset	Support
$A_1 \cdot \mathit{Low}$	0.303	$A_3 \cdot \mathit{High}$	0.303
$A_1 \cdot \mathit{Middle}$	0.425	$A_4 \cdot \mathit{Middle}$	0.485
$A_2 \cdot \mathit{Middle}$	0.485	$A_5 \cdot \mathit{Low}$	0.303
$A_3 \cdot \mathit{Middle}$	0.455	$A_5 \cdot \mathit{Middle}$	0.485

STEP 8: In this step, the candidate set C_{r+1} is generated from L_r. C_2 is then first generated from L_1 as follows: ($A_1 \cdot \mathit{Low}$, $A_2 \cdot \mathit{Middle}$), ($A_1 \cdot \mathit{Low}$, $A_3 \cdot \mathit{Middle}$), ($A_1 \cdot \mathit{Low}$, $A_3 \cdot \mathit{High}$), ($A_1 \cdot \mathit{Low}$, $A_4 \cdot \mathit{Middle}$), ($A_1 \cdot \mathit{Low}$, $A_5 \cdot \mathit{Low}$), ($A_1 \cdot \mathit{Low}$, $A_5 \cdot \mathit{Middle}$), ($A_1 \cdot \mathit{Middle}$, $A_2 \cdot \mathit{Middle}$), ($A_1 \cdot \mathit{Middle}$, $A_3 \cdot \mathit{Middle}$), ($A_1 \cdot \mathit{Middle}$, $A_3 \cdot \mathit{High}$), ($A_1 \cdot \mathit{Middle}$, $A_4 \cdot \mathit{Middle}$), ($A_1 \cdot \mathit{Middle}$, $A_5 \cdot \mathit{Low}$), ($A_1 \cdot \mathit{Middle}$, $A_5 \cdot \mathit{Middle}$), ($A_2 \cdot \mathit{Middle}$, $A_3 \cdot \mathit{Middle}$), ($A_2 \cdot \mathit{Middle}$, $A_3 \cdot \mathit{High}$), ($A_2 \cdot \mathit{Middle}$, $A_4 \cdot \mathit{Middle}$), ($A_2 \cdot \mathit{Middle}$, $A_5 \cdot \mathit{Low}$), ($A_2 \cdot \mathit{Middle}$, $A_5 \cdot \mathit{Middle}$), ($A_3 \cdot \mathit{Middle}$, $A_4 \cdot \mathit{Middle}$), ($A_3 \cdot \mathit{Middle}$, $A_5 \cdot \mathit{Low}$), ($A_3 \cdot \mathit{Middle}$, $A_5 \cdot \mathit{Middle}$), ($A_3 \cdot \mathit{High}$, $A_4 \cdot \mathit{Middle}$), ($A_3 \cdot \mathit{High}$, $A_5 \cdot \mathit{Low}$), ($A_3 \cdot \mathit{High}$, $A_5 \cdot \mathit{Middle}$), ($A_4 \cdot \mathit{Middle}$, $A_5 \cdot \mathit{Low}$), and ($A_4 \cdot \mathit{Middle}$, $A_5 \cdot \mathit{Middle}$). Note that no two fuzzy items with the same A_i are put in a candidate 2-itemset.

STEP 9: The following substeps are done for each newly formed candidate itemset.

(a) The fuzzy membership value of each candidate itemset in each subsequence is calculated. Here, assume the minimum operator is used for the intersection. Take ($A_1 \cdot \mathit{Low}$, $A_2 \cdot \mathit{Middle}$) as an example. The derived membership value for this candidate 2-itemset in s_p is calculated as: $\mathit{min}(1.0, 0.67) = 0.67$. The results for the other subsequences are shown in Table 5.

The results for the other 2-itemsets can be derived in a similar way.

(b) The scalar cardinality (count) of each candidate 2-itemset in the time series is then calculated. Results for this example are shown in Table 6.

(c) The supports of the above candidate itemsets are then calculated and compared with the predefined minimum support (30%). In this example, only the two itemsets, ($A_1 \cdot \mathit{Middle} \cap A_4 \cdot \mathit{Middle}$) and ($A_2 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Middle}$), satisfy this condition. They are thus kept in L_2 (Table 7).

STEP 10: Since L_2 is not null in the example, $r = r + 1 = 2$. STEPs 7–9 are then repeated to find L_3. C_3 is first generated from L_2. In this example, no candidate 3-itemsets can be formed. L_3 is thus an empty set. STEP 11 then begins.

STEP 11: The large 2-itemset ($A_2 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Middle}$) is then shifted into ($A_1 \cdot \mathit{Middle} \cap A_4 \cdot \mathit{Middle}$) since its first region occurs in A_2.

Table 5
The membership values for $A_1 \cdot \mathit{Low} \cap A_2 \cdot \mathit{Middle}$.

s_p	$A_1 \cdot \mathit{Low}$	$A_2 \cdot \mathit{Middle}$	$A_1 \cdot \mathit{Low} \cap A_2 \cdot \mathit{Middle}$
1	1	1	1.0
2	0	0	0.0
3	0	0.33	0.0
4	0	0.67	0.0
5	0.33	0.33	0.33
6	0.67	0.67	0.67
7	0	0	0.0
8	0	1	0.0
9	0	0.67	0.0
10	0.33	0	0.0
11	1	0.67	0.67

Table 6
The fuzzy counts of the itemsets in C_2.

Itemset	Count	Itemset	Count
$A_1 \cdot \mathit{Low} \cap A_2 \cdot \mathit{Middle}$	2.67	$A_2 \cdot \mathit{Middle} \cap A_3 \cdot \mathit{High}$	2.33
$A_1 \cdot \mathit{Low} \cap A_3 \cdot \mathit{Middle}$	1.33	$A_2 \cdot \mathit{Middle} \cap A_4 \cdot \mathit{Middle}$	3
$A_1 \cdot \mathit{Low} \cap A_3 \cdot \mathit{High}$	2.33	$A_2 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Low}$	1.66
$A_1 \cdot \mathit{Low} \cap A_4 \cdot \mathit{Middle}$	1.66	$A_2 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Middle}$	3.67
$A_1 \cdot \mathit{Low} \cap A_5 \cdot \mathit{Low}$	1.33	$A_3 \cdot \mathit{Middle} \cap A_4 \cdot \mathit{Middle}$	2.66
$A_1 \cdot \mathit{Low} \cap A_5 \cdot \mathit{Middle}$	2.33	$A_3 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Low}$	2.33
$A_1 \cdot \mathit{Middle} \cap A_2 \cdot \mathit{Middle}$	1.66	$A_3 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Middle}$	3
$A_1 \cdot \mathit{Middle} \cap A_3 \cdot \mathit{Middle}$	2.67	$A_3 \cdot \mathit{High} \cap A_4 \cdot \mathit{Middle}$	2.33
$A_1 \cdot \mathit{Middle} \cap A_3 \cdot \mathit{High}$	1.33	$A_3 \cdot \mathit{High} \cap A_5 \cdot \mathit{Low}$	1.66
$A_1 \cdot \mathit{Middle} \cap A_4 \cdot \mathit{Middle}$	3.34	$A_3 \cdot \mathit{High} \cap A_5 \cdot \mathit{Middle}$	2.33
$A_1 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Low}$	1.67	$A_4 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Low}$	2.33
$A_1 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Middle}$	2.33	$A_4 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Middle}$	2.99
$A_2 \cdot \mathit{Middle} \cap A_3 \cdot \mathit{Middle}$	2.33		

STEP 12: The two large itemsets ($A_1 \cdot \mathit{Middle} \cap A_4 \cdot \mathit{Middle}$) and ($A_1 \cdot \mathit{Middle} \cap A_4 \cdot \mathit{Middle}$) are the same and only one of them is kept.

STEP 13: The association rules for each large itemset are then constructed by the following substeps.

(a) All the possible association rules are first formed from the large itemsets. In this example, only the large 2-itemset ($A_1 \cdot \mathit{Middle} \cap A_4 \cdot \mathit{Middle}$) exists. The following two possible association rules are then formed:

1. If A_1 = Middle, then A_4 = Middle;
2. If A_4 = Middle, then A_1 = Middle;

(b) The confidence values of the above association rules are then calculated. Take the first association rule as an example. The fuzzy counts of $A_1 \cdot \mathit{Middle}$, and $A_4 \cdot \mathit{Middle}$ are calculated as 4.67 and 3.34. The confidence of the association rule "If $A_1 = \mathit{Middle}$, then $A_4 = \mathit{Middle}$" is then calculated as:

$$\frac{\sum_{p=1}^{11}(A_1.\mathit{Middle} \cap A_4.\mathit{Middle})}{\sum_{p=1}^{11}(A_1.\mathit{Middle})} = \frac{3.34}{4.67} = 0.715.$$

Results for the other rule are shown below.

"If $A_4 = \mathit{Middle}$, then $A_1 = \mathit{Middle}$" has a confidence of 0.626;

STEP 14: The confidence values of the above association rules are then compared with the predefined confidence threshold λ. Assume λ is set at 0.65. The following rule is thus output to users:

1. If $A_1 = \mathit{Middle}$, then $A_4 = \mathit{Middle}$, with a confidence factor of 0.72;

This rule means that if the value of a data point is middle, then the value of a data point after three time units will also be middle with a high probability. The above rule can thus be used as the meta-knowledge concerning the given time series.

4. Experimental results

In this section, the experiments made to show the performances of the proposed method are described. They were implemented in Visual C++ 6.0 at a personal computer with Intel Pentium IV 3.00 GHz and 512MB RAM. The dataset consisted of the first 100 stock prices in Japan Nikkei 225 market from June, 30, 1989 [11], which is shown in Fig. 2. The sliding-window size was set at 5.

Table 7
The itemsets and their fuzzy supports in L_2.

Itemset	Support
$A_1 \cdot \mathit{Middle} \cap A_4 \cdot \mathit{Middle}$	0.304
$A_2 \cdot \mathit{Middle} \cap A_5 \cdot \mathit{Middle}$	0.334

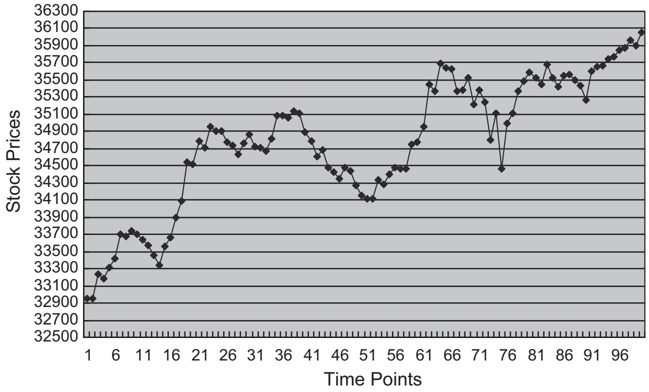

Fig. 2. The first 100 stock prices in Japan Nikkei 225 market from June, 30, 1989.

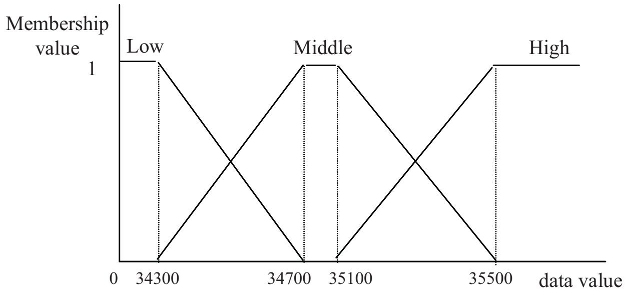

Fig. 3. The membership functions used in the experiments.

Assume the fuzzy membership functions for the data values are defined as shown in Fig. 3. There are three fuzzy membership functions, *Low*, *Middle* and *High*, used in the experiments.

Experiments were first made to show the relationships between numbers of association rules and minimum support values along with different minimum confidence values. The results are shown in Fig. 4.

From Fig. 4, it is easily seen that the numbers of association rules decreased along with the increase in minimum support values. Also, the curve of numbers of association rules with larger minimum confidence values was smoother than that of those with smaller minimum confidence values, meaning that the minimum support value had a large effect on the number of association rules derived from small minimum confidence values.

The relationship between numbers of association rules and minimum confidence values along with various minimum support values is shown in Fig. 5.

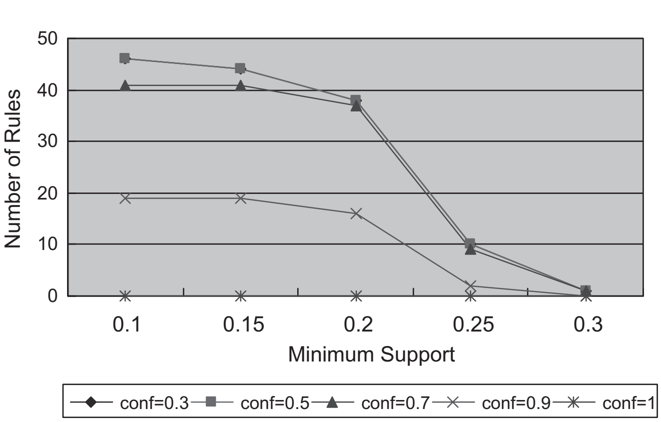

Fig. 4. The relationship between numbers of rules and minimum supports.

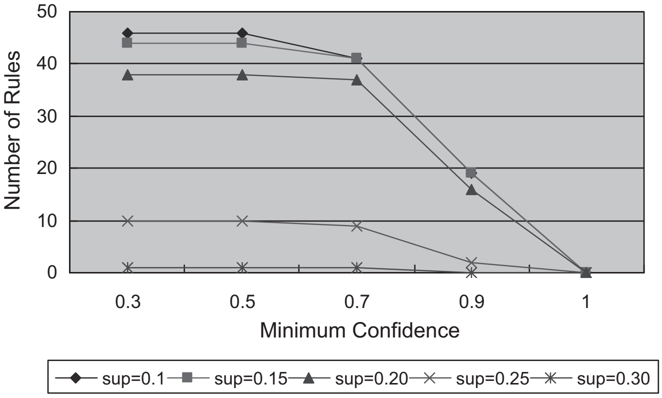

Fig. 5. The relationship between numbers of rules and minimum confidences.

From Fig. 5, it is easily seen that the numbers of association rules decreased along with an increase in minimum confidence values. The curve of numbers of association rules with larger minimum support values was smoother than that for smaller minimum support values, meaning that the minimum confidence value had a larger effect on the number of association rules when smaller minimum support values were used. All of the various curves approached to 0 as the minimum confidence value approached 1.

Fig. 6 shows the relationship between numbers of association rules and minimum confidence values along with different sliding-window sizes, when the minimum support was set at 20%. As expected, when the size of sliding-windows increased, the number of rules also increased.

Then, experiments were made to compare the numbers of association rules generated with and without STEP 12 of removing redundant large itemsets. The results are shown in Fig. 7.

From Fig. 7, it can be easily observed that removing redundant large itemsets during the mining process has its efficacy. Without this step, too many redundant rules may be generated and computational time may be wasted.

At last, we show some derived fuzzy rules to evaluate the proposed approach. When the sliding window size and the minimum support were set at 9 and 0.2, three selected fuzzy rules were shown as follows:

1. *Rule*$_1$: If $A_1 = High$, Then $A_2 = High$ (confidence = 0.86);
2. *Rule*$_2$: If $A_1 = Low$, and $A_5 = Low$, Then $A_7 = Low$ (confidence = 0.83);
3. *Rule*$_3$: If $A_1 = Middle$, and $A_5 = Middle \rightarrow A_9 = Middle$ (confidence = 0.70).

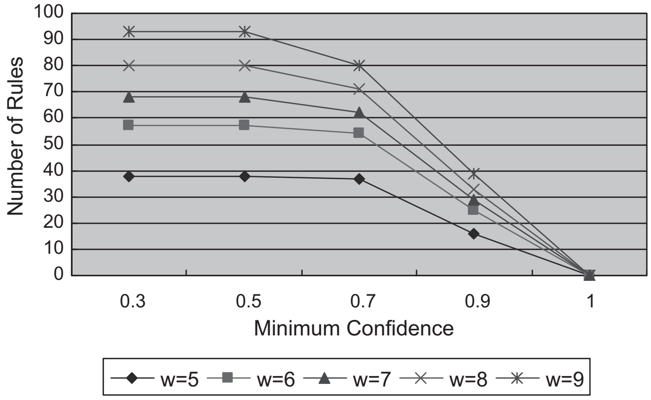

Fig. 6. The relationship between numbers of rules and minimum confidences among different sliding-windows.

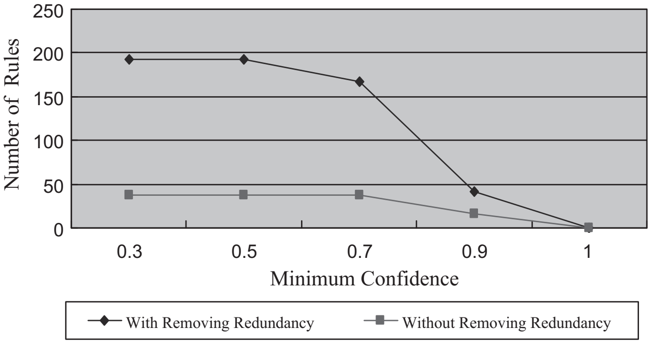

Fig. 7. The numbers of rules with and without removing redundancy.

Rule$_1$ meant that if the value of a data point was high, then the value of the data point in the next time unit would also be high with a probability of 0.86. This rule was reasonable and could be considered for a short-term reference. Next, when the time units increased to 7, it could be known from *Rule*$_2$ that if the stock prices were "*Low*" at the first and the fifth time units, then the stock price was also "*Low*" at the seventh time unit with a probability of 0.83. At last, when the time units increased to nine, it was shown from *Rule*$_3$ that if the stock prices were "*Middle*" at the first and the fifth time units, then the stock price was also "*Middle*" at the ninth time unit with a probability of 0.7. Since its confidence value was 0.7, it could be realized that the stock prices were a little unstable. The proposed approach is thus capable of providing some linguistic statements for describing the phenomenon of the applied financial data.

5. Discussion

As mentioned in the previous section, the proposed approach has two advantages. The first one is that the final results represented by linguistic rules are friendlier to human than quantitative representation. The second one is that the proposed approach can remove lots of redundant rules through appropriate post-processing, such that users can utilize the rules more easily. Some existing approaches like neural-network approaches [25,31] or T-S fuzzy models [28] can be used to deal with time series data as well. In neural-network approaches, the models are trained from the raw data through multiple middle layers (also called hidden layers). However, the hidden layers in neural networks are only a black box to users. When compared to our approach, the results from neural networks are not easily understood by users.

As to the T-S fuzzy model, it is usually used in the control field to find proper values of controllable variables. It can also be used in time series to predict the next value from the previous values. However, our approach is designed from the viewpoint of data mining for finding the regularity and the varying trends in time series. Thus by our approach, users can observe the tendency from the derived linguistic rules, which can not provided by neural-network approaches and the T-S fuzzy model.

6. Conclusions and future works

In this paper, we have attempted to use the data mining technique to analyze time series. We proposed a fuzzy time-series mining algorithm that integrates the fuzzy sets, the *Apriori* mining algorithm, and the time-series concepts to find out appropriate linguistic association rules. The proposed approach first uses a sliding window to generate continues subsequences from a given time series and then analyzes the fuzzy itemsets from these subsequences. Appropriate post-processing is then performed to remove redundant patterns. Experiments are also made to show the relationships between numbers of association rules, minimum supports and minimum confidences. Since the final results are represented by linguistic rules, they will be more friendly to human than quantitative representation. The rules thus mined exhibit quantitative regularity in time series and can be used to provide some suggestions to appropriate supervisors. They can be used in two ways, prediction and post-analysis. For instance, if a rule like "If A_1 is *Low* and A_3 is *Middle*, then A_5 is *High*" is mined, it can be used to predict the behavior of A_5 from A_1 and A_3. On the contrary, if a rule like "If A_1 is *Low* and A_5 is *High*, then A_3 is *Middle*" is mined, it can be used for post-analysis. The proposed approach thus provides another alternative to analyze time series.

Although the proposed method works for time series, it is just a beginning. There is still much work to be done in this field. Our method assumes that the membership functions are known in advance. In [17,18], we proposed some fuzzy learning methods to automatically derive the membership functions. In the future, we will attempt to dynamically adjust the membership functions in the proposed mining algorithm to avoid the bottleneck of membership-function acquisition. More validation and applications may also be explored in the future. We will also continuously attempt to enhance the mining algorithm for more complex problems.

References

[1] W.H. Au, K.C.C. Chan, Mining fuzzy rules for time series classification, in: The 2004 IEEE International Conference on Fuzzy Systems, vol. 1, 2004, pp. 239–244.
[2] J. Aach, G. Church, Aligning gene expression time series with time warping algorithms, Bioinformatics 17 (2001) 495–508.
[3] R. Agrawal, T. Imielinksi, A. Swami, Mining association rules between sets of items in large database, in: The 1993 ACM SIGMOD Conference, Washington DC, USA, 1993, pp. 207–216.
[4] R. Agrawal, T. Imielinksi, A. Swami, Database mining: a performance perspective, IEEE Transactions on Knowledge and Data Engineering 5 (6) (1993) 914–925.
[5] R. Agrawal, R. Srikant, Q. Vu, Mining association rules with item constraints, in: The Third International Conference on Knowledge Discovery in Databases and Data Mining, Newport Beach, California, 1997, pp. 67–73.
[6] R. Agrawal, R. Srikant, Fast algorithm for mining association rules, in: The International Conference on Very Large Databases, 1994, pp. 487–499.
[7] F.L. Chung, T.C. Fu, R. Luk, V. Ng, Evolutionary time series segmentation for stock data mining, in: 2002 IEEE International Conference on Data Mining, 9–12 December, 2002, pp. 83–90.
[8] C.H. Cai, W.C. Fu, C.H. Cheng, W.W. Kwong, Mining association rules with weighted items, in: The International Database Engineering and Applications Symposium, 1998, pp. 68–77.
[9] S.M. Chen, J.R. Hwang, Temperature prediction using fuzzy time series, IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics 30 (2) (2000) 263–275.
[10] B. Chiu, E. Keogh, S. Lonardi, Research track: Probabilistic discovery of time series motifs, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.
[11] S.H. Chen, C.H. Yeh, Using genetic programming to model volatility in financial time series: The case of Nikkei 225 and S&P 500, in: The Second Annual Conference on Genetic Programming, 1997, pp. 58–63.
[12] G. Das, K. Lin, H. Mannila, G. Renganathan, P. Smyth, Rule discovery from time series, in: Proceedings of the 4 the International Conference on Knowledge Discovery and Data Mining, New York, NY, 1998, pp. 16–22.
[13] S. Erdal, O. Ozturk, D. Armbruster, H. Ferhatosmanoglu, W.C. Ray, A time series analysis of microarray data, Fourth IEEE Symposium on Bioinformatics and Bioengineering (2004) 366–375.
[15] I. Graham, P.L. Jones, Expert Systems – Knowledge, Uncertainty and Decision, Chapman and Computing, Boston, 1988, pp.117–158.
[16] T.P. Hong, C.S. Kuo, S.C. Chi, Mining association rules from quantitative data, Intelligent Data Analysis 3 (5) (1999) 363–376.
[17] T.P. Hong, C.H. Chen, Y.L. Wu, Y.C. Lee, Using divide-and-conquer GA strategy in fuzzy data mining, in: The Ninth IEEE Symposium on Computers and Communications, 2004.
[18] T.P. Hong, C.H. Chen, Y.L. Wu, Y.C. Lee, Fining active membership functions in fuzzy data mining, in: The Fourth Workshop on the Foundation of Data Mining and Discovery in The 2004 IEEE International Conference on Data Mining, 2004, pp. 65–71.
[19] T.P. Hong, C.S. Kuo, S.C. Chi, Trade-off between time complexity and number of rules for fuzzy mining from quantitative data, International Journal of Uncertainty, Fuzziness and Knowledge-based Systems 9 (5) (2001) 587–604.
[20] A. Kandel, Fuzzy Expert Systems, CRC Press, Boca Raton, 1992, pp. 8–19.

[22] H. Lei, V. Govindaraju, Regression time warping for similarity measure of sequence, in: The Fourth International Conference on Computer and Information Technology, September, 2004, pp. 826–830.

[23] Y.C. Lee, T.P. Hong, W.Y. Lin, Mining fuzzy association rules with multiple minimum supports using maximum constraints, Lecture Notes in Computer Science 3214 (2004) 1283–1290.

[24] E.H. Mamdani, Applications of fuzzy algorithms for control of simple dynamic plants, in: IEEE Proceedings, 1974, pp. 1585–1588.

[25] Y.Q. Peng, Y. Zhang, H.S. Tian, Research of time series pattern finding based on artificial neural network, in: 2003 International Conference on Machine Learning and Cybernetics, vol. 3, 2003, pp. 1385–1388.

[26] Q. Song, B.S. Chissom, Fuzzy time series and its models, Fuzzy Sets System 54 (3) (1993) 269–277.

[27] J.P.C. Valente, I.L. Chavarrias, Discovering similar patterns in time series, in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2000, pp. 497–505.

[28] N. Watanabe, A fuzzy rule based time series model, IEEE Annual Meeting of the Fuzzy Information 2 (2004) 936–940.

[29] R. Weber, Fuzzy-ID3: a class of methods for automatic knowledge acquisition, in: The Second International Conference on Fuzzy Logic and Neural Networks, Iizuka, Japan, 1992, pp. 265–268.

[31] Y. Yang, G. Liu, Multivariate time series prediction based on neural networks applied to stock market, in: 2001 IEEE International Conference on Systems, Man, and Cybernetics, vol. 4, 2001, p. 2680.

[32] S. Yue, E. Tsang, D. Yeung, D. Shi, Mining fuzzy association rules with weighted items, in: The IEEE International Conference on Systems, Man and Cybernetics, 2000, pp. 1906–1911.

[33] L.A. Zadeh, Fuzzy sets, Information and Control 8 (3) (1965) 338–353.