1000-9825/2001/12(07)0949-11 ©2001 Journal of Software 软件学报 Vol.12, No.7

# Efficient Algorithms for Mining Fuzzy Rules in Large Relational Databases*

CHEN Ning[1], CHEN An[2], ZHOU Long-xiang[1]

[1](*Institute of Mathematics, Academy of Mathematics and System Sciences, The Chinese Academy of Sciences, Beijing* 100080, *China*);

[2](*Institute of Policy and Management, The Chinese Academy of Sciences, Beijing* 100080, *China*)

E-mail: nchen@math08.math.ac.cn

http://www.amss.ac.cn

Received January 7, 2000; accepted December 19, 2000

**Abstract:** Mining association rules and sequential rules from large databases is an important task of data mining. Previous work is focused on definite and accurate concepts, which may not be concise and meaningful enough for human experts to easily obtain nontrivial knowledge from the rules discovered. The definition of fuzzy concepts is based on fuzzy set theory, which is especially useful when the discovered rules are presented to human experts for examination. In this paper, algorithms are presented for discovering fuzzy association rules and fuzzy sequential rules expressed by fuzzy concepts from large relational databases.

**Key words:** fuzzy concept; fuzzy association rule; fuzzy sequential rule

Data mining, sometimes referred to as knowledge discovery from databases, is a non-trivial process of identifying implicit, valid, novel, potentially useful, and ultimately understandable patterns in data[1]. Data mining or knowledge discovery from databases has been recognized by many researchers as a key research topic in database systems and machine learning and by many industrial companies as an important area with an opportunity of major revenues.

Mining association rules between items over basket data was introduced in Ref. [2]. Given a transaction database, it is desirable to discover the important association among items such that the presence of some items in a transaction will imply that of other items in the same transaction. Association rules can be used for catalog design, store layout, product placement and target marketing. Many algorithms have been proposed for discovering boolean association rules. Quantitative association rule discovery involves discretizing the domain of quantitative attributes into intervals. These intervals may not be concise and meaningful enough for human experts to easily obtain nontrivial knowledge from those rules discovered. The definition of fuzzy concepts is based on fuzzy set theory, which is especially useful when the discovered rules are presented to human experts for examination and is easy to be understood by users. In this paper, we propose the algorithms for discovering fuzzy association rules with

* Supported by the National Natural Science Foundation of China under Grant No.69983011 (国家自然科学基金)

**CHEN Ning** was born in 1974. She received her Ph.D. degree in computer software from Mathematics Institute, The Chinese Academy of Sciences in 2001. Her research interests are data mining and knowledge discovery. **CHEN An** was born in 1970. He received his Ph.D. degree from Economics and Management School, Beijing University of Aeronautics & Astronautic in 2001. His research interests are supply chain management, operational research and data mining. **ZHOU Long-xiang** was born in 1938. He is a professor and doctoral supervisor of Mathematics Institute, The Chinese Academy of Sciences. His current research interests are distributed database, multimedia database, data mining and data warehouse.

these fuzzy concepts from large databases. The fuzzy concept is better than the partition method because fuzzy sets provide a smooth transition between member and non-member of a set. The use of fuzzy techniques makes the algorithms resilient to noise and missing values in the databases. In contrast with other related work, fuzzy concepts are not confined to single attribute, instead, they can be defined on a set of attributes. We also introduce the definition of fuzzy sequential rules and extend the above algorithms for the discovery of fuzzy sequential rules.

The paper is organized as follows: We introduce the related work in Section 1. The definition of fuzzy association rules is given and two algorithms for mining fuzzy association rules are studied in Section 2. We give the definition of fuzzy sequential rules and extend the above algorithms to mining fuzzy sequential rules in Section 3. Performance results are described in Section 4. The paper is concluded in Section 5.

## 1 Related Work

The problem of generating association rules was first introduced and an algorithm called AIS was proposed in Ref. [2]. Another algorithm called SETM was proposed using relation operations in Ref. [3]. Apriori and Apriori-Tid were proposed to decrease the run time of mining association rules in Ref. [4]. A boolean association rule involves binary attributes, a generalized association rule involves attributes that are hierarchically related, a multiple-level association rule involves multiple-level concepts of hierarchies and a quantitative association rule involves attributes that can take quantitative or categorical values. Many algorithms have been proposed for mining boolean, quantitative, generalized and multiple-level association rules. Other techniques such as parallel and distributed algorithms were also discussed to increase the efficiency of mining association rules.

Quantitative association rules[5] were defined over quantitative and categorical attributes. The values of categorical attributes were mapped to a set of contiguous integers. While the domain of quantitative attributes was discretized into intervals by fine-partitioning the values of the attributes and combining the adjacent partitions as necessary, the intervals were then mapped to contiguous integers. As a result, each attribute had a form of ⟨attribute, value⟩ where value was the mapped integer of an interval for quantitative attributes or a single value for categorical attributes.

Then the algorithms for finding boolean association rules can be used on the transformed database to discover quantitative association rules.

R. Agrawal first introduced the problem of mining sequential patterns and presented three algorithms for solving this problem in Ref. [6]. The problem of mining sequential patterns from large database has been observed by many researchers recently.

F-APACS[7] was able to mine fuzzy association rules which utilized linguistic representation and it used an objective yet meaningful confidence measure to determine the interest of a rule making it very effective in the discovery of rules from a real-life transaction database. But F-APACS only found association rules of two fuzzy terms. To deal with quantitative attribute, Ref. [8] assigned each attribute with several fuzzy sets, finding the fuzzy association rules in the form of "If $X$ is $A$ the $Y$ is $B$" where $X$ and $Y$ were sets of attributes, $A$ and $B$ were fuzzy sets which described $X$ and $Y$ respectively. But the fuzzy sets were only defined on one attribute which confined the implication of fuzzy set concept. Fuzzy association rules were also introduced and an algorithm was proposed to discover the rules of interest in Ref. [9].

## 2 Mining Fuzzy Association Rules

In this section, we present the definition of fuzzy concept based on fuzzy set theory, and describe two algorithms for discovering fuzzy association rules, which make use of fuzzy concepts to represent the regularities and exceptions discovered from relational databases. The definition of fuzzy concept and fuzzy association rule is given in Section 2.1. The overview of mining fuzzy association rules is described in Section 2.2. An algorithm called F_Apriori for finding large fuzzy patterns is presented in Section 2.3. After that, F_AprioriSet algorithm is proposed in Section 2.4, adopting fuzzy set operations to decrease the amount of scanned data during each pass.

## 2.1 Definitions

In a given relational database $D=\{d_1,d_2,\ldots,d_n\}$, each record consists of a tid and a set of attributes: $I=\{A_1,A_2,\ldots,A_m\}$, which can be quantitative or categorical attributes. We do not consider boolean attributes since they can be seen as a special case of categorical attributes with only two values: 0 and 1. For any record $d\in D$, let $d[A]$ denote the value in d for attribute $A$. Let $\mathrm{dom}(A)=[l_A,u_A]\subseteq R$ denote the domain of a quantitative attribute $A$. For a categorical attribute $A$, $\mathrm{dom}(A)$ is a set of discrete values.

**Definition 1.** A fuzzy concept $x$ is a linguistic term defined over the domain of a set of attributes $\{A_1,A_2,\ldots,A_i\}\subseteq I$ based on fuzzy set theory. The definition of $x$ includes two parts: the domain which is the cartesian product of domain of $A_j$, $1\leqslant j\leqslant i$, and the membership function $\mu_x$ mapping each element of the domain to a real value between 0 and 1, that is, $\mu_x$: $\mathrm{dom}(A_1)\times\mathrm{dom}(A_2)\times\ldots\times\mathrm{dom}(A_i)\to[0,1]$.

The fuzzy concepts and corresponding membership functions are provided by domain experts. In general, a fuzzy concept is defined on single attribute. But it also can be defined on a set of attributes. According to Zadeh's notation, the fuzzy set formed by $x$ is then defined as

$$L_x=\sum_{t\in\mathrm{dom}(A_1)\times\ldots\times\mathrm{dom}(A_i)}\mu_x(t)/t \text{ or } L_x=\int_{t\in\mathrm{dom}(A_1)\times\ldots\times\mathrm{dom}(A_i)}\mu_x(t)/t$$

In the above form, the integral notation does not imply the normal meaning, instead, it only summarizes the relationship between each element and its membership to the fuzzy set. The set of fuzzy concepts defined on $D$ is denoted as $F=\{x_1,x_2,\ldots,x_p\}$, each concept $x_i$ representing a fuzzy set $L_{x_i}$. For example, if three fuzzy concepts are defined on attribute 'Salary': {low, medium, high}, we will have $F=\{\mathrm{Salary}_{\mathrm{low}},\mathrm{Salary}_{\mathrm{medium}},\mathrm{Salary}_{\mathrm{high}}\}$, with each fuzzy concept corresponding to a fuzzy set.

Let $d\in D$ be a record and $x$ be a fuzzy concept defined on a set of attributes: $A_x=\{A_1,A_2,\ldots,A_i\}$, $d[A_x]\in\mathrm{dom}(A_1)\times\mathrm{dom}(A_2)\times\ldots\times\mathrm{dom}(A_i)$ be the value of $d$ on $A_x$. In order to simplify the description, we denote $\mu_x(d[A_x])=\mu_x(d)$. The membership grade of $d$ with respect to $x$ is given by $\mu_{x_i}(d)$. If $\mu_x(d)=1$, $d$ is completely characterized by the concept $x$. If $\mu_x(d)=0$, $d$ is undoubtedly not characterized by the concept $x$. If $0<\mu_x(d)<1$, $d$ is partially characterized by the concept $x$.

**Definition 2.** A fuzzy pattern $X$ is a non-empty set of fuzzy concepts, denoted by $x_1\wedge x_2\wedge\ldots\wedge x_k$ where $x_j$ is a fuzzy concept. The length of a fuzzy pattern is the number of concepts in it. A fuzzy pattern of length $k$ is called a $k$-pattern. A fuzzy concept is also called 1-pattern. The membership grade of $d$ with respect to $X$ is defined as $\mu_X(d)=\min\{\mu_{x_i}(d)\mid x_i\in X\}$. Since $0\leqslant\mu_{x_i}(d)\leqslant1$, then $0\leqslant\mu_X(d)\leqslant1$.

**Definition 3.** The support for a fuzzy pattern $X$ is defined as the faction of the sum of membership grades for all records with respect to $X$ and the total number of records in $D$, that is $\sup(X)=\left(\sum_{d\in D}(\mu_X(d)\mid\mu_X(d)\geqslant\varepsilon,\ \varepsilon\in[0,1])\right)/|D|$. If the membership grade is less than a user specified threshold $\varepsilon$, it will not be considered. Since for each record $d\in D$, $0\leqslant\mu_X(d)\leqslant1$, then $0\leqslant\sup(X)\leqslant1$. A fuzzy pattern $X$ is large if its support is no less than min-sup, which is a given minimum support.

**Definition 4.** A fuzzy association rule is an implication of the form $A\Rightarrow B$, where $A$ and $B$ are fuzzy patterns. The support count of the rule is defined as $\sup(A\wedge B)$. The interest of the rule is defined as $\sup(A\wedge B)/\sup(A\cup B)=\sup(A\wedge B)/(\sup(A)+\sup(B)-\sup(A\wedge B))$, which considers the effects of both $\sup(A)$ and $\sup(B)$ and is more objective and meaningful than the definition of confidence. Let min-interest be the minimum interest implying the minimum strength of rules. A fuzzy association rule is strong if its support and interest are no less than min-sup and min-interest respectively.

Problem statement: Given a database $D$ with attributes $I$ and those fuzzy concepts $F$ associated with attributes in $I$, the problem of mining fuzzy association rules is to find the desired rules expressed by fuzzy concepts that have certain user specified minimum support and minimum interest.

**2.2 Overview: mining fuzzy association rules**

The process of mining fuzzy association rules can be divided into two steps:

(1) Generate large fuzzy patterns by scanning the database: Two algorithms are given respectively in Sections 2.3 and 2.4. We assume the fuzzy concepts are mapped to a set of contiguous integers and the records are stored in increasing order of tids. The notations used in the algorithms are listed in Table 1.

**Table 1** Notations of $F$_Apriori and $F$_AprioriSet

| Symbol | Description |
|---|---|
| $L_k$ | Set of large fuzzy $k$-patterns |
| $C_k$ | Set of candidate fuzzy $k$-patterns |
| $C[i]$ | The $i$th fuzzy concept of fuzzy pattern $C$ |
| $S_k$ | Set of tidlists for $L_k$, $S_k=\{s(X)\mid X\in L_k\}$ (Used in $F$_AprioriSet) |
| $L$ | Set of all large fuzzy patterns, $L=\bigcup_{k=1}^{n} L_k$, where $n$ is the maximum length of large fuzzy patterns |
| $S$ | Set of all tidlists of large fuzzy patterns, $S=\bigcup_{k=1}^{n} S_k$(Used in $F$_AprioriSet) |

(2) Get fuzzy association rules: Having gotten large fuzzy patterns, the desired fuzzy association rules are generated as follows: for each large fuzzy pattern $C$, we find all non-empty prefix patterns $A$ of $C$. For every such sub-pattern $A$, a rule of the form $A\rightarrow B$ ($A\wedge B=C$) is output if its interest is at least min-interest. We need to consider all prefix subsequences of $C$ in order to generate all fuzzy association rules having the minimum interest with corresponding consequences.

**2.3 F_Apriori algorithm**

We get large fuzzy 1-patterns by searching $D$ in the first pass. In the $k$th ($k>1$) pass, we first generate candidate $k$-patterns using large ($k-1$)-patterns, then calculate the membership grade of each record with respect to every candidate pattern, accumulate the support of candidate patterns and get all large fuzzy $k$-patterns at the end of the pass. The process terminates until an empty set of large patterns is generated at some pass. The algorithm is as follows:

**F_Apriori algorithm.**

```
$L_1$=set of large 1-patterns in $F$;
for ($k:=2$; $L_{k-1}\neq\varnothing$; $k++$) do {
$C_k=\{C=P[1]P[2]...P[k-1]Q[k-1]\mid P[1]=Q[1],...,P[k-2]=Q[k-2],P[k-1]<Q[k-1]$ and $P\in$
      $L_{k-1}$, $Q\in L_{k-1}$ and $\forall c\in L_{k-1}$, where $c$ is a $(k-1)$-subpattern of $C\}$;
for each $C\in C_k$ do sup$(C)=0$;
for each record $d\in D$ do
for each $C\in C_k$ do
{$\mu_C(d)=\min(\mu_{C[i]}(d))$;
 if $\mu_C(d)>\varepsilon$ then sup$(C)=$sup$(C)+\mu_C(d)$;
 }
$L_k=\{C\in C_k\mid$sup$(C)\geq$min-sup$\}$;
}
$L=\bigcup L_k$;
```

In contrast to the original Apriori algorithm, $F$_Apriori needs extra space to store the membership function of fuzzy concepts and the membership grade of records instead of support count. More over, it costs more time to calculate the membership grade of records in each pass.

## 2.4 F_AprioriSet algorithm

In F_Apriori algorithm, we scan the database and calculate the membership grade of each record with respect to candidate fuzzy patterns in every pass. In fact, since the membership grade of a record with respect to a $k$-pattern can be calculated from those with respect to two $(k-1)$-subpatterns of it, so it is not necessary to scan the database in every pass.

Given a fuzzy pattern $X$, let $s(X)$ be the tidlist of $X$, each element of which is the tid of a record and the membership grade with respect to $X$. The elements in $s(X)$ are arranged in an increasing order of tids, i.e., $s(X)=\{(d.\text{tid}, \mu_X(d)) \mid \mu_X(d)>\varepsilon, d\in D\}$. $S_k=\{s(X)\mid X\in L_k\}$, where $L_k$ is the set of all large $k$-patterns. Since a record with membership less than $\varepsilon$ does not contribute to the support, it will not be saved in tidlists. Although there is only a small savings of one element, the reduction in elements can be significant for fuzzy sets with many elements of small membership.

**Lemma 1.** $\forall X=x_1\wedge x_2\wedge\ldots\wedge x_k\in C_k$, $s(X)$ can be generated by $s(Y)$ and $s(Z)$ where $Y$ and $Z(Y\neq Z)$ are two elements of $L_{k-1}$.

*Proof.* Let $Y=x_1\wedge x_2\wedge\ldots\wedge x_{k-2}\wedge x_{k-1}$, $Z=x_1\wedge x_2\wedge\ldots\wedge x_{k-2}\wedge x_k$ be two $(k-1)$-subpatterns of $X$. For $\forall d\in D$, $\mu_Y(d)=\min\{\mu_{x_i}(d)\mid i=1,\ldots,k-1\}$, $\mu_Z(d)=\min\{\mu_{x_i}(d)\mid i=1,\ldots,k-2,k\}$. Then $\mu_X(d)=\min\{\mu_{x_i}(d)\mid x_i\in X, i=1,\ldots,k\}=\min(\mu_Y(d),\mu_Z(d))$. So $s(X)=\{(d.\text{tid}, \min(\mu_Y(d),\mu_Z(d)))\mid (d.\text{tid},\mu_Y(d))\in s(Y)$ and $(d.\text{tid},\mu_Z(d))\in s(Z)\}$. $X\in C_k$, then $Y,Z\in L_{k-1}$ according to the generation procedure of $C_k$.

From Lemma 1, we know that the algorithm is independent of the database $D$ after $S_1$ has been built at the end of the first pass. $S_k$ can be calculated by fuzzy set intersection operations of two elements in $S_{k-1}$ for $k\geq 2$. We can obtain $L_k=\{X\mid X\in C_k$ and $\sup(X)\geq$min-sup$\}$ at the end of $k$th pass. In general, the number of tidlists and the size of each tidlist in $S_k$ decrease as $k$ increases so that the amount of scanned data in every pass decreases gradually.

**F_AprioriSet algorithm.**

```
(1)  $C_1=F$; // $C_1$ is the set of fuzzy concepts
(2)  for each $x\in C_1$ do {$s(x)=\varnothing$; $\sup(x)=0$;}
(3)  for each $d\in D$ do
(4)      for each $x\in C_1$ do
(5)          if $\mu_x(d)>\varepsilon$ then
(6)          {add($d$.tid, $\mu_x(d)$) to $s(x)$; $\sup(x)=\sup(x)+\mu_x(d)$;}
(7)  $L_1=\{x\mid x\in C_1$ and $\sup(x)\leqslant$min-sup$\}$;
(8)  $S_1=\{s(x)\mid x\in L_1\}$;
(9)  for ($k$:=2; $L_{k-1}\neq\varnothing$; $k$++) do {
(10)     $L_k=\varnothing$; $S_k=\varnothing$;
(11)     $C_k=\{C=P[1]P[2]\ldots P[k-1]Q[k-1]\mid P[1]=Q[1],\ldots,P[k-2]=Q[k-2],P[k-1]<Q[k-1]$
         and $P\in L_{k-1}$, $Q\in L_{k-1}$ and $\forall c\in L_{k-1}$, where $c$ is a $(k-1)$-subpattern of $C\}$;
(12)     for all $C\in C_k$ do{
(13)         $\sup(C)=0$;
(14)         $A=C[1]C[2]\ldots C[k-2]C[k-1]$;
(15)         $B=C[1]C[2]\ldots C[k-2]C[k]$; // select two $(k-1)$-subpatterns of $C$
(16)         for each element pair $(d_1.\text{tid}, \mu_A(d_1))\in s(A)$ and $(d_2.\text{tid}, \mu_B(d_2))\in s(B)$ and $d_1.\text{tid}=d_2.\text{tid}$;
(17)         {add $(d_1.\text{tid}, \min(\mu_A(d_1),\mu_B(d_2)))$ to $s(C)$; $\sup(C)=\sup(C)+\min(\mu_A(d_1),\mu_B(d_2))$; } }
(18)     $L_k=\{C\in C_k\mid \sup(C)\geqslant$min-sup$\}$;
(19)     $S_k=\{s(X)\mid X\in L_k\}$; }
(20) $L=\bigcup L_k$
```

$S_1$ and $L_1$ are obtained by scanning $D$ in Steps (1)~(8), and $L_k$ and $S_k$ are determined in Steps (10)~(19) for $k\geqslant 2$: $C_k$ is first generated by $L_{k-1}$. For each candidate pattern $C\in C_k$, $s(C)$ can be derived by fuzzy set operations from two elements of $S_{k-1}$. If $|s(C)|\geqslant$min-sup, $s(C)$ and $C$ are inserted into $S_k$ and $L_k$ respectively, otherwise $s(C)$ is deleted from $S_k$.

Since the records in $D$ are stored in an increasing order of record tids, the elements in $s(X)$ are stored in the same order after the sequential scan of $D$ in the first pass. Furthermore, the result $s(C)$ is also in the same sorted order after set operations of $s(A)$ and $s(B)$. The process of getting $s(C)$ in this case involves only the cost of traversing the two sets once. The efficiency of this technique is higher than the method of getting the support by scanning the database adopted in F_Apriori algorithm. On the other hand, more space is necessary to store the intermediate tidlists.

Let $a$ be the space to store a fuzzy pattern $C\in L_k$, $b$ be the space to store the key of a record and the membership grade with respect to $C$, $n_C$ be the number of records having membership grade no less than ε. Then, $|S_k|=\sum_{C\in L_k}(a+n_C\times b)=|L_k|\times a+b\times\sum_{C\in L_k}n_C$. If $L_k$ has been given, the value of $|S_k|$ can be estimated using above heuristic. As $k$ increases, $|L_k|$ and number of record with membership grade no less than ε of each large fuzzy pattern decrease so that the space of $S_k$ decreases.

## 3 Mining Fuzzy Sequential Rules

Sequential data can be often seen in real databases. Sometimes, each customer corresponds to a set of records in a database and the records are stored with cus_tid as the major key and rec_tid as the minor key. In this section, we give the definition of fuzzy sequential rule first. Then we will describe two algorithms called F_Seq and F_SetSeq based on F_Apriori and F_AprioriSet respectively, for mining fuzzy sequential rules.

### 3.1 Definitions

In a given database $D_s$ of customer records, each customer corresponds to a set of records ordered by the increasing value of rec_tids. No customer has more than one record with the same rec_tid. A fuzzy sequence $S$ is a non-empty and ordered list of fuzzy patterns, denoted by $\langle X_1,X_2,\ldots,X_n\rangle$, where $X_j$ is a fuzzy pattern. The length of a sequence is the number of patterns in it. A sequence of length $k$ is called a $k$-sequence. Specially, a single pattern forms one 1-sequence. All records of a customer can together be viewed as a sequence, called a record-sequence.

**Definition 5.** Given a record-sequence $C=\langle d_1,d_2,\ldots,d_n\rangle\in D_s$ and a fuzzy sequence $S=\langle X_1,X_2,\ldots,X_m\rangle$, the membership of $C$ with respect to $S$ is defined as $\mu_S(C)=\max_{1\leqslant i_1<i_2<\ldots<i_m\leqslant n}\min_{j=1,\ldots,m}\mu_{X_j}(d_{i_j})$. Since $0\leqslant\mu_{X_j}(d)\leqslant 1$, then $0\leqslant\mu_S(C)\leqslant 1$. Specially, for a fuzzy pattern $X$, the membership of $C$ with respect to $X$ is defined as $\mu_X(C)=\max_{1\leqslant i\leqslant n}\mu_X(d_i)$.

**Definition 6.** The support for a fuzzy sequence $S$ is defined as the faction of the sum of membership grades for all record-sequences with respect to $S$ and the total number of customers in $D_s$, such as $\sup(S)=\sum_{C\in D_s}(\mu_S(C)|\mu_C(d)\geqslant\varepsilon)/|\text{number of customers in } D_s|$. Since for each record-sequence $C\in D_s$, $0\leqslant\mu_S(C)\leqslant 1$, then $0\leqslant\sup(S)\leqslant 1$. A fuzzy sequence $S$ is large if its support is no less than min-sup, which is a given threshold.

**Definition 7.** A fuzzy sequential rule is an implication of the form $A\Rightarrow B$, where $A$ and $B$ are fuzzy sequences. The support of the rule is defined as $\sup(\langle A,B\rangle)$. The interest of the rule is defined as $\sup(\langle A,B\rangle)/\sup(A)$. A fuzzy sequential rule is strong if its support and interest are no less than min-sup and min-interest respectively.

**Problem statement:** Given a database $D_s$ of record-sequences with attributes $I$ and those fuzzy concepts $F$ associated with attributes in $I$, the problem of mining fuzzy sequential rules is to find the desired rules expressed by fuzzy concepts that have certain user specified minimum support and minimum interest.

## 3.2 Mining fuzzy sequential patterns

The process of mining fuzzy sequential rules can be divided into three phases:

Phase 1 (Large Fuzzy Patterns Phase). In this phrase, we get all large fuzzy patterns $L$ from $D_s$ using algorithms proposed in Section 2. We also simultaneously get the set of large 1-sequences: $SL_1 = \{\langle l\rangle \mid l\in L\}$. Lastly, the set of large fuzzy patterns is mapped to a set of contiguous integers. The reason of this mapping is that by treating large fuzzy patterns as single entities, we can compare two large patterns for equality in constant time, and reduce the time required for candidate sequences generation. Notice the definition of support of fuzzy pattern is different from that in Section 2.

Phase 2 (Large Fuzzy Sequences Phase). We find the set of large fuzzy sequences $SL$ in this phase.

Phase 3 (Fuzzy Sequential Rules Phase). Having found all large fuzzy sequences $SL$ in phase 2, we can generate the desired rules. For each large sequence $S$, we find all non-empty prefix subsequences $A$ of $S$. For every such subsequence $A$, a rule of the form $A\Rightarrow B(\langle A,B\rangle = S)$ is output if sup($S$)/sup($A$) is at least min-interest.

Since the first phase has been introduced in Section 2 and the third is simple, we focus on the second phase and propose two algorithms. The notations used in the algorithm are listed in Table 2.

**Table 2** Notations of F_Seq and F_SetSeq

| Symbol | Description |
|---|---|
| $SL_k$ | Set of large fuzzy $k$-sequences |
| $SC_k$ | Set of candidate fuzzy $k$-sequences |
| $SL$ | Set of all large fuzzy sequences, $SL=\bigcup_{k=1}^{n} SL_k$, where $n$ is the maximum length of large fuzzy sequences |
| $S[i]$ | The $i$th pattern of fuzzy sequence $S$ |
| $SS$ | Set of tidlists for $L$, $SS=\{ss(X)\mid X\in L\}$ (Used in $F\_SeqSet$) |

## 3.3 F_Seq algorithm

We first get all large fuzzy patterns $L$ from $D_s$ using F_Apriori algorithm proposed in Section 2.3. In each pass, we generate the candidate fuzzy sequences using the large fuzzy sequences obtained in previous pass and scan the database to calculate the membership grade of each record sequence with respect to candidate fuzzy sequences. At the end of the pass we calculate the support of candidate sequences and get the large fuzzy sequences. The process is as follows:

```
(1)  $SL_1$ = large 1-sequnence; // $SL_1$ has been calculated at the end of phase 1
(2)  for ($k := 2$; $SL_{k-1} \neq \varnothing$; $k$++) do {
(3)  $CL_k = \{S = P[1]P[2]\ldots P[k-1]Q[k-1] \mid P[1]=Q[1],\ldots,P[k-2]=Q[k-2]$ and $P\in SL_{k-1}, Q\in$
         $SL_{k-1}$ and $\forall s\in SL_{k-1}$, where $s$ is a $(k-1)$-subsequence of $S\}$;
(4)  for each $S\in SC_k$ do sup($S$)=0;
(5)  for each record sequences $C$ do
(6)  for all $S\in SC_k$ do {
(7)  $\mu_S(C) = \max_{1\le i_1<i_2<\ldots<i_m\le |C|} \max_{j=1,\ldots,k} \mu_{S[j]}(d_{ij})$;
(8)  if $\mu_S(C) > \varepsilon$ then sup($S$) = sup($S$) + $\mu_S(C)$; }
(9)  $SL_k = \{S\in SC_k \mid \sup(S) \geq$ min-sup$\}$; }
(10) $S_L = \bigcup SL_k$
```

## 3.4 F_SetSeq algorithm

After getting all large fuzzy patterns $L$ from $D_s$ using F_AprioriSet algorithm proposed in Section 2.4, we transform $s(X)$ for each $X\in L$ to another representation by extracting cus_tid and merging all rec_tids with the same cus_tid into an element of the following data structure:

```
struct {
    int cus_tid;
    int rec_num; // number of records with the same cus_tid
    record[rec_num]; // the set of rec_tids and membership grades of records
      {int rec_tid;
      real mem_grade;
      }
    }ss[ ];
```

$SS=\{ss(X)\mid X\in L\}$ is defined as the set of tidlists of all large fuzzy patterns.

For each $X\in L$, the element of $ss(X)$ consists of cus_tid and a set of rec_tids and the corresponding membership grades of the records with respect to $X$ for the customer if the membership value is more than $\varepsilon$.

Next, we find the set of all large fuzzy sequences $SL$ by scanning the tidlists of $SL_1$ in this phase. $\forall\ S=X_1\wedge X_2\wedge\ldots\wedge X_k\in CL_k$, $\sup(S)$ can be calculated by $ss(X_j)$, $j=1,\ldots,k$, where $X_j\in L$ and $ss(X_j)$ have been generated. We scan every element with the same cus_tid for $ss(X_j)$, $j=1,\ldots,k$. Let $C$ be a record-sequence. If $\exists\ p_j$ for $X_j(j=1,\ldots,k)$, $ss(X_j)[p_j]$.cus_tid$=C$.cus_tid, we denote $r(X_j)=ss(X_j)[p_j]$.record and rec_num$(X_j)=ss(X_j)[p_j]$.rec_num.

$\mu_S(C)=\max\limits_{1\leq i_j\leq rec\_num(X_j)}\min\limits_{j=1,\ldots,k}(r(X_j)[i_j]$.mem_grade), where $r(X_1)[i_1]$.rec_tid$<r(X_2)[i_2]$.rec_tid$<\ldots<r(X_k)[i_k]$.rec_tid. Then, $\sup(S)=\sum\limits_{C\in D_S}(\mu_S(C)\mid\mu_C(d)\geq\varepsilon)/\mid$number of customers in $D_s\mid$. The process is as follows:

```
(1) $SL_1$=large 1-sequnence; // $SL_1$ has been calculated at the end of phase 1
(2) for ($k:=2$; $SL_{k-1}\neq\varnothing$; $k++$) do {
(3) $SL_k=\{S=P[1]P[2]\ldots P[k-1]Q[k-1]\mid P[1]=Q[1],\ldots,P[k-2]=Q[k-2]$ and $P\in SL_{k-1}$, $Q\in$
      $SL_{k-1}$ and $\forall\ s\in SL_{k-1}$, where $s$ is a $(k-1)$-subsequence of $S\}$;
(4) for all $S\in SC_k$ do
(5) calculate $\sup(S)$ by scanning the tidlists of each $S[i]$;
(6) $SL_k=\{S\in SC_k\mid\sup(S)\leq$min-sup$\}$; }
(7) $SL=\bigcup SL_k$
```

F_SeqSet algorithm uses the same approach to determine the candidate fuzzy sequences before each pass begins. The difference from F_Seq is that support of candidate sequences is obtained by scanning the tidlist set SS in Set_Seq, instead of scanning the database. Since the elements in tidlists are stored in an increasing order of cus_tids and rec_tids, the scanning process is very efficient. The efficiency of this technique is higher than the method of getting support by scanning the database adopted in F_Seq, especially when the database is huge.

## 4 Performance

Some experiments have been carried out to evaluate the performance of the algorithms and study their scale-up properties. All the experiments were performed on a Pentium 586 personal computer running Windows98 with main memory of 32MB. We applied the algorithms to a relational database which contains 10,000 diagnostic records of 500 pneumonic patients in a hospital during 3 months. Each record consists of 20 attributes including patient number, patient age, diagnostic time, and some symptoms such as temperature, pulses/minute, palpitations/minute and other attributes. In order to study the relationship among these attributes, the algorithms proposed above are performed on the database. We define some fuzzy concepts on the attributes. For example: 'old' and 'young' are defined on the attribute 'age' as follows:

$$\text{'young'}=\begin{cases}1 & 0\leqslant u\leqslant 25\\ \dfrac{1}{1+\left(\dfrac{u-25}{5}\right)^2} & u>25\end{cases},\quad \text{'old'}=\begin{cases}0 & 0\leqslant u\leqslant 50\\ \dfrac{1}{1+\left(\dfrac{u-50}{5}\right)^2} & u>50\end{cases}$$

We define 'high fever', 'low fever' and 'normal' on attribute temperature as:

$$\text{'high fever'}=\begin{cases}1 & u>40\\ \dfrac{1}{1+(40-u)^2} & 37\leqslant u\leqslant 40,\\ 0 & u<37\end{cases}\quad \text{'low fever'}=\begin{cases}\dfrac{1}{1+(u-38)^2} & u>38\\ 1 & 37\leqslant u\leqslant 38,\\ \dfrac{1}{1+(36-u)^2} & u<37\end{cases}$$

$$\text{'normal'}=\begin{cases}0 & u>40\\ \dfrac{1}{1+(u-36)^2} & 36\leqslant u\leqslant 40\\ 1 & u<36\end{cases}$$

Other fuzzy concepts also can be defined such as 'fast', 'a little fast', 'normal', 'slow' on pulse and palpitation, and 'good health', 'bad health', and 'common health' on attributes: temperature, pulse and palpitation. Due to lack of space, we do not list these fuzzy concepts in the paper. We applied the algorithms proposed in Section 2 to the patient database and got the association rules. For example, age= 'old' and temperature= 'high'⇒ palpitation= 'fast'. If the records of a patient are arranged according to the increasing order of diagnostic time, fuzzy sequential rules can be obtained using the algorithms proposed in Section 3, such as, temperature= 'high'⇒ pulse= 'fast'.

### 4.1 Relative performance

Figure 1 shows the execution time of the four algorithms for decreasing values of min-sup. As min-sup decreases, the execution time of all algorithms increases because of increases in the total number of candidate and large fuzzy patterns. In F_AprioriSet, the cost of generating the support decreases during later passes as the number and length of tidlists become smaller. This improvement is more obvious for small min-sup because the number of passes increases and the tidlists are much smaller than the original database. On the other hand, our algorithm exhibits disadvantage compared with AprioriAll when the min-sup is high as it spends much time initializing the data structures without deriving much benefit in processing cost. Similarly, F_SetSeq performs better than F_Set because of the advantage of set operation.

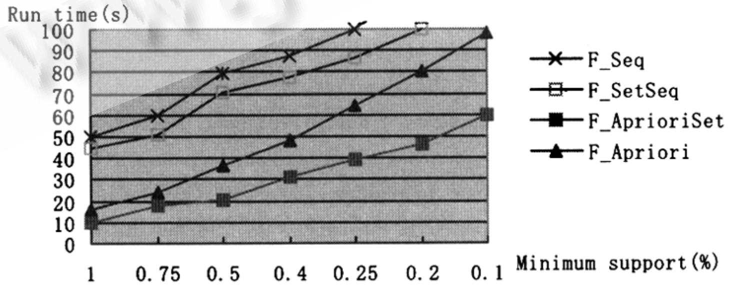

Fig. 1 Execution time

### 4.2 Scale-Up experiment

We present the results of scale-up experiments for the algorithms in this section. Figure 2 shows how

F_Apriori and F_AprioriSet scale up as the number of records increases ten times from 1000 to 10,000. The minsup is set to 0.5. The execution times are normalized with respect to the times for 1000 records. As shown, the execution time of two algorithms scales up linearly with the number of records.

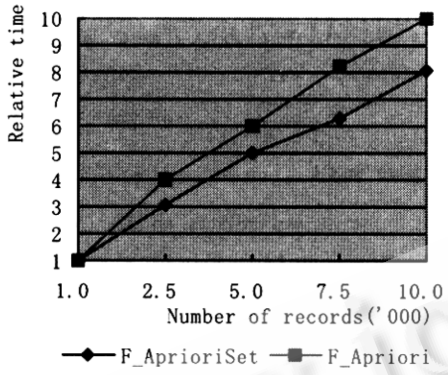

Fig. 2 Scale-Up: number of records

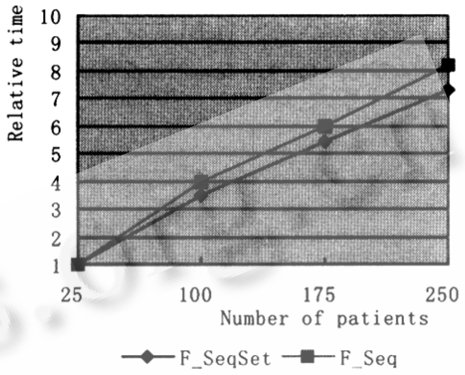

Fig. 3 Scale-Up: number of patients

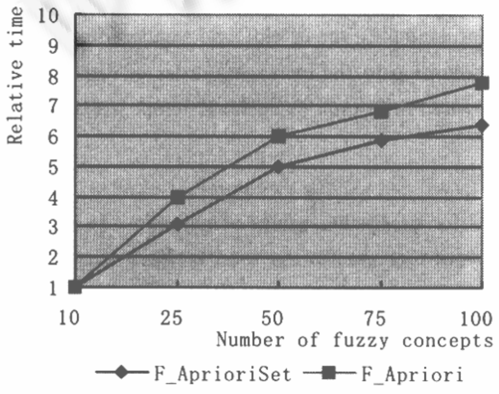

Fig. 4 Scale-Up: number of fuzzy concepts

Figure 3 shows the scale-up property of F_Seq and F_SetSeq as the number of patients increases from 25 to 250 with each patient having 20 records. The minimum support is set to 0.5%. As expected, the execution time scales up linearly with the number of patients.

We also examine how the algorithms scale up as the number of fuzzy concepts is increased from 10 to 100. Figure 4 shows that as the number of fuzzy concepts increases, the execution time increases dramatically at first because more candidate and large fuzzy patterns are generated and calculating their support cost more time. The increasing trend becomes much slower for large number of fuzzy concepts because no more large fuzzy patterns are generated.

## 5 Conclusion

The use of fuzzy techniques has been considered as one of the key components of data mining systems because of the affinity with the human representation[10]. Fuzzy set theory in particular is more and more frequently used in expert systems because of its simplicity and similarity to human reasoning. Fuzzy sets handle numerical values better than existing methods because fuzzy sets soften the effect of sharp boundaries. Using fuzzy concepts, the discovered rules are more understandable to human. In addition, the use of fuzzy techniques makes the algorithms resilient to noise and missing values in the databases. In this paper, we introduce the problem of mining fuzzy association rules and sequential rules from large relational databases. We also propose algorithms to solve the problem. Experiments demonstrate that the algorithms based on fuzzy set operation perform better. In addition, the algorithms have excellent scale-up property.

In this paper, we offer the following contributions:

• We introduce the definition of fuzzy association rules and fuzzy sequential rules expressed by fuzzy concepts.

• We propose two algorithms for mining fuzzy association rules.

• We extend the above algorithms to discover fuzzy sequential rules.

In the future, we plan to extend this work along the following directions:

• Extend our work to parallel and incremental association rules and sequential rules mining.

• The fuzzy rules discovered strongly depend on the fuzzy concepts defined. Defining reasonable membership function of fuzzy concepts needs further work.

**References:**

[1] Chen, Ming-Syan, Han, Jia-wei, Yu, Philip. Data mining: an overview from a database perspective. IEEE Transactions on Knowledge and Data Engineering, 1996,8(6):866～883.

[2] Agrawal, Rakesh, Imielinski, Tomasz, Swami Arun. Mining association rules between sets of items in large databases. In: Peter Buneman, Sushil Jajodia, eds. Proceedings of the ACM SIGMOD International Conference on Management of Data. Washington D.C.: ACM Press, 1993. 207～216.

[3] Houtsma, M., Swami, A. Set-Oriented mining of association rules. Research Report RJ 9567, IBM Almaden Research Center, San Jose, California, 1993.

[4] Agrawal, Rakesh, Srikant, Ramakrishnan. Fast algorithm for mining association rules. In: Jorge B. Bocca, Matthias Jarke, Carlo Zaniolo, eds. Proceedings of the 20th International Conference on Very Large Data Bases. Santiago de Chile Chile: Morgan Kaufmann Publishers, 1994. 487～499.

[5] Srikant, Ramakrishnan, Agrawal, Rakesh. Mining quantitative association rules in large relational tables. In: Jagadish H. V., Mumick Inderpal Singh, eds. Proceedings of the ACM SIGMOD International Conference on Management of Data. Montreal, Canada: ACM Press, 1996. 1～12.

[6] Agrawal, Rakesh, Srikant, Ramakrishnan. Mining sequential patterns. In: Philip S. Yu, Arbee L. P. Chen, eds. Proceedings of the 11th International Conference on Data Engineering. Taiwan: IEEE Computer Society, 1995. 3～14.

[7] Chan, Keith C.C., Au Wai-Ho. Mining fuzzy association rules. In: Forouzan Golshani, Kia Makki, eds. Proceedings of the 6th International Conference on Information and Knowledge Management (CIKM'97). Las Vegas, Nevada, 1997. 209～215.

[8] Chan, Man Kuok, Fu, Ada, Won, Man Hon. Mining fuzzy association rules in database. SIGMOD Record, 1998,27(1): 41～46.

[9] Cheng, Ji-hua, Shi, Peng-fei, Guo, Jian-sheng. An algorithm for mining fuzzy association rules. Mini-micro Systems, 1999,20(4):270～274 (in Chinese).

[10] Maeda, A., Ashida, J., Taniguchi, Y. Data mining system using fuzzy rule induction. In: Proceedings of the IEEE International Conference on Fuzzy Systems. Yokohama, Japan, 1995. 45～46.

**附中文参考文献：**

[9] 程继华,施鹏飞,郭建生.模糊关联规则及挖掘算法.小型微型计算机系统,1999,20(4):270～274.

## 关系数据库中模糊规则的快速挖掘算法

陈 宁[1]，陈 安[2]，周龙骧[1]

[1](中国科学院 数学与系统科学研究院 数学研究所,北京 100080);

[2](中国科学院 科技政策与管理科学研究所,北京 100080)

**摘要**：关联规则和时序规则是数据挖掘的任务之一.在以往的算法中,规则通常用确定的数值或概念来表示,往往不具有实际意义,而且不容易被用户理解.研究了从大型关系数据库中挖掘模糊关联规则和模糊时序规则的问题.基于模糊集合的理论,提出了两个模糊关联规则的挖掘算法,然后把它们分别扩展为模糊时序规则的挖掘算法.用模糊概念表示的规则更符合人的思维和表达习惯,增强了规则的可理解性.

**关键词**：模糊概念;模糊关联规则;模糊时序规则

**中图法分类号**：TP311 **文献标识码**：A