A DISTRIBUTED ALGORITHM FOR MINING FUZZY ASSOCIATION RULES

George Stephanides
University of Macedonia, Department of Applied Informatics
156 Egnatia Street, 540 06 Thessaloniki GREECE

Mihai Gabroveanu, Mirel Cosulschi, Nicolae Constantinescu
University of Craiova, Computer Science Department
13 A.I. Cuza Street, 200585 Craiova ROMANIA

Keywords: data mining, fuzzy association rules, distributed mining.

Abstract: Data mining, also known as knowledge discovery in databases, is the process of discovery potentially useful, hidden knowledge or relations among data from large databases. An important topic in data mining research is concerned with the discovery of association rules. The majority of databases are distributed nowadays. In this paper is presented an algorithm for mining fuzzy association rules from these distributed databases. This algorithm is inspired from DMA (**D**istributed **M**ining of **A**ssociation rules) algorithm for mining boolean association rules.

1 INTRODUCTION

Data mining, also known as knowledge discovery in databases, is the process of discovery potentially useful, hidden knowledge or relations among data from large databases. An important task in data mining process is the discovery of association rules. An association rule describes an interesting relationship among different attributes.

The task of discovering association rules was first introduced in (Agrawal R., 1993). Many of proposed algorithms for mining association rules are sequential algorithms. The most popular are: Apriori (Rakesh Agrawal, 1994), DHP, DIC. The basic problem of finding fuzzy association rules was introduced in (Chan Man Kuok, 1998).

Mining association rules based on fuzzy sets can handle quantitative and categorical data, providing the necessary support to use uncertain data types with existing algorithms. Today the majority of databases are distributed. The records of transactions corresponding to each customer operation registered in a stores chain distributed in many locations form an example of such databases. The main problem here is to discover the association rules from this distributed data.

In this paper we introduce an algorithm for mining fuzzy association rules from these distributed databases. This algorithm is an adaptation of DMA algorithm used here for mining fuzzy association rules.

2 PROBLEM DEFINITION

2.1 Sequential problem definition

The formal problem definition as in (Chan Man Kuok, 1998) is the following:

Let $\mathcal{DB} = \{t_1, \ldots, t_n\}$ a transactional database. We consider that this database is characterized by a set of categorical or quantitative attributes (items). Let $\mathcal{I} = \{i_1, \ldots, i_m\}$ the set of these attributes. We note with $dom(i_k)$ the domain of values for the attribute i_k. For each attribute i_k, $(k = 1, \ldots, m)$ we will consider $n(k)$ associated fuzzy sets. Let $F_{i_k} = \{f^1_{i_k}, \ldots, f^{n(k)}_{i_k}\}$ be the set of fuzzy sets. For an attribute i_k and a fuzzy set $f^j_{i_k}$, the membership function is $\mu_{f^j_{i_k}}$.

Definition 2.1. *We call **fuzzy itemset** the tuple $\langle X, F_X \rangle$, where $X \subseteq \mathcal{I}$, and F_X is a set of fuzzy sets associated with items from X. A fuzzy itemset $\langle X, F_X \rangle$ is called **k-fuzzy itemset** if the number of attributes from X is k.*

Definition 2.2. *A **fuzzy association rule** is an implication with following form $X \in A \Rightarrow Y \in B$, where $X, Y \in \mathcal{I}$, $X \cap Y = \emptyset$, $X = \{x_1, \ldots, x_p\}$, $Y = \{y_1, \ldots, y_q\}$. $A = \{a_1, \ldots, a_p\}$ and $B = \{b_1, \ldots, b_q\}$ are fuzzy sets related to attributes from X, respectively Y. More exactly, $a_i \in F_{x_i}$, $(i = 1, \ldots, p)$, and $b_i \in F_{y_i}$, $(i = 1, \ldots, q)$.*

Stephanides G., Gabroveanu M., Cosulschi M. and Constantinescu N. (2005).
A DISTRIBUTED ALGORITHM FOR MINING FUZZY ASSOCIATION RULES.
In *Proceedings of the First International Conference on Web Information Systems and Technologies*, pages 206-209
DOI: 10.5220/0001228802060209

We denote this rule with $\langle X, A\rangle \Rightarrow \langle Y, B\rangle$.

The intuitively signification of this fuzzy association rule $\langle X, A\rangle \Rightarrow \langle Y, B\rangle$ is: "if a transaction (tuple) satisfies the property $X \in A$ then it will satisfy the property $Y \in B$ with a high probability also".

Definition 2.3. *The **fuzzy support value** of itemset $\langle X, F_X\rangle$ in $\mathcal{DB}$ is:*

$$FS_{\langle X,F_X\rangle} = \frac{\sum_{t_i\in\mathcal{DB}} \prod_{x_j\in X} \alpha_{a_j}(t_i[x_j])}{|\mathcal{DB}|}$$

where

$$\alpha_{a_j}(t_i[x_j]) = \begin{cases} \mu_{a_j}(t_i[x_j]), & \text{if } \mu_{a_j}(t_i[x_j]) \geq \omega \\ 0, & \text{otherwhise} \end{cases}$$

and ω is a user specified minimum threshold for the membership function. Thus, the values of membership functions lesser than this minimum threshold are ignored.

Definition 2.4. *A fuzzy itemset $\langle X, F_X\rangle$ is called a **large (frequent) fuzzy itemset** if its fuzzy support value is greater than or equal to the minimum support threshold (minsup), namely $FS_{\langle X,F_X\rangle} \geq minsup$.*

An association rule is considered as *interesting* if it has enough support and high confidence value. This association rule can be encountered under the name *strong rule*.

Problem 1 (Sequential Mining Fuzzy Association Rules)**.** *Given the database $\mathcal{DB}$ characterized by a set of attributes $\mathcal{I}$, the fuzzy sets associated with attributes from $\mathcal{I}$, ω the minimum support threshold for membership function, the minimum support threshold (minsup) and the minimum confidence threshold (minconf), extract all interesting fuzzy association rules.*

Definition 2.5. *Let $\langle X, A\rangle \Rightarrow \langle Y, B\rangle$ be a fuzzy association rule. The **fuzzy support value of the rule** is defined as fuzzy support value of the itemset $\langle\{X, Y\}, \{A, B\}\rangle$:*

$$FS_{\langle X,A\rangle\Rightarrow\langle Y,B\rangle} = FS_{\langle\{X,Y\},\{A,B\}\rangle}$$

Definition 2.6. *A fuzzy association rule is called a **frequent rule** if its fuzzy support value is greater than or equal to the minimum support threshold ($minsup$), namely $FS_{\langle X,A\rangle\Rightarrow\langle Y,B\rangle} \geq minsup$.*

Based on discovered *large fuzzy itemsets* we can generate all possible frequent rules, but in order to be *interesting* they must have a high confidence value.

Definition 2.7. *Let $\langle X, A\rangle \Rightarrow \langle Y, B\rangle$ a fuzzy association rule. The **fuzzy confidence value** of the rule is defined as:*

$$FC_{\langle X,A\rangle\Rightarrow\langle Y,B\rangle} = \frac{FS_{\langle Z,C\rangle}}{FS_{\langle X,A\rangle}}$$

where $Z = \{X, Y\}$ and $C = \{A, B\}$

The confidence of the rule is defined as the fraction between the value of fuzzy support of the fuzzy itemset $\langle Z, C\rangle$ and the value of fuzzy support of the fuzzy itemset $\langle X, A\rangle$.

Lemma 1. *If a fuzzy itemset $\langle X, F_X\rangle$ is a large fuzzy itemset in $\mathcal{DB}$, $Y \subseteq X$, $F_Y \subseteq F_X$, then also fuzzy itemsets $\langle Y, F_Y\rangle$ are large in $\mathcal{DB}$.*

From the above lemma we can draw the conclusion that any fuzzy subitemset of a large fuzzy itemset is also large.

The problem of sequential mining of fuzzy association rules can be decomposed in two subproblems:

1. *find all large fuzzy itemsets.*
2. *generate the fuzzy association rules from the large fuzzy itemsets founded.*

The majority of algorithms for mining fuzzy association rules (see (Gyenesei, 2000), (Hong T.P., 2000)) are based on the algorithm Apriori (Rakesh Agrawal, 1994).

2.2 Distributed problem definition

Let $\mathcal{DB} = \{\mathcal{DB}_1, \mathcal{DB}_2, \ldots, \mathcal{DB}_n\}$ be a distributed database over n sites $S_1, S_2, \ldots, S_n$. We denote with D the number of transactions from $\mathcal{DB}$, and with D_i the number of transactions from $\mathcal{DB}_i$, for all $i = 1, \ldots, n$.

Definition 2.8. *For a given fuzzy itemset $\langle X, F_X\rangle$ we call **global fuzzy support value** the fuzzy support value of $\langle X, F_X\rangle$ in $\mathcal{DB}$ defined as:*

$$FS_{\langle X,F_X\rangle} = \frac{\sum_{t_i\in\mathcal{DB}} \prod_{x_j\in X} \alpha_{a_j}(t_i[x_j])}{|\mathcal{DB}|}$$

*and **global fuzzy support count** in $\mathcal{DB}$ is defined as:*

$$CFS_{\langle X,F_X\rangle} = \sum_{t_i\in\mathcal{DB}} \prod_{x_j\in X} \alpha_{a_j}(t_i[x_j])$$

Definition 2.9. *For a given fuzzy itemset $\langle X, F_X\rangle$ and a database $\mathcal{DB}_i$ we call **local fuzzy support value in** DB_i the fuzzy support value of $\langle X, F_X\rangle$ in DB_i defined as:*

$$FS^i_{\langle X,F_X\rangle} = \frac{\sum_{t_i\in\mathcal{DB}_i} \prod_{x_j\in X} \alpha_{a_j}(t_i[x_j])}{|\mathcal{DB}_i|}$$

*and **local fuzzy support count** in $\mathcal{DB}_i$ is defined as:*

$$CFS^i_{\langle X,F_X\rangle} = \sum_{t_i\in\mathcal{DB}_i} \prod_{x_j\in X} \alpha_{a_j}(t_i[x_j])$$

Let $minsup$ be the minimum support threshold.

Definition 2.10. *A fuzzy itemset $\langle X, F_X\rangle$ is called **global large fuzzy itemset** if $FS_{\langle X,F_X\rangle} \geq minsup$.*

Definition 2.11. *A fuzzy itemset $\langle X, F_X \rangle$ is called* ***local large fuzzy itemset at site*** S_i *if* $FS^i_{\langle X, F_X \rangle} \geq minsup$.

Definition 2.12. *If a fuzzy itemset $\langle X, F_X \rangle$ is both globally large and locally large at a site S_i, it is called* ***gl-large fuzzy itemset at site*** S_i.

In the following, we will denote with L the set of all globally large fuzzy itemsets in $\mathcal{DB}$, and with $L_{(k)}$ the set of all globally large k-fuzzy itemsets in $\mathcal{DB}$.

Problem 2 (Distributed Mining Fuzzy Association Rules)**.** *Given the set of items $\mathcal{I}$, the distributed database $\mathcal{DB} = \{\mathcal{DB}_1, \mathcal{DB}_2, \ldots, \mathcal{DB}_n\}$, the fuzzy sets associated with attributes from $\mathcal{I}$, the minimum support threshold ($minsup$) and the minimum confidence threshold ($minconf$), extract all global fuzzy association rules.*

3 THE DISTRIBUTED ALGORITHM

In (Cheung D.W., 1996), the authors proposed a DMA algorithm for mining boolean association rules from distributed databases.

3.1 Generate set of candidate fuzzy itemsets

The candidate fuzzy itemsets reduction is made on the basis of the properties of the global large fuzzy itemsets and local large fuzzy itemsets subsequently presented:

Lemma 2. *If a fuzzy itemset $\langle X, F_X \rangle$ is locally large at a site S_i, then all its subsets are also locally large at site S_i,*

Lemma 3. *If a fuzzy itemset $\langle X, F_X \rangle$ is globally large, then there exist a site S_i, $(1 \leq i \leq n)$, such that $\langle X, F_X \rangle$ is locally large at site S_i.*

Lemma 4. *If a fuzzy itemset $\langle X, F_X \rangle$ is gl-large fuzzy itemset at a site S_i, $(1 \leq i \leq n)$, then all its sub-fuzzy itemsets, $\langle Y, F_Y \rangle$, $Y \subseteq X$, are also gl-large fuzzy itemsets at site S_i.*

We use GL^i to denote the set of all gl-large fuzzy itemsets at site S_i, and $GL^i_{(k)}$ to denote all k-gl-large fuzzy itemsets at site S_i.

Lemma 5. *If $\langle X, F_X \rangle \in L_{(k)}$, (i.e. is a globally large fuzzy k-itemset), then there exists a site S_i, $(1 \leq i \leq n)$ such that $\langle X, F_X \rangle$ and all its (k-1) sub-fuzzy itemsets are gl-large fuzzy itemsets at site S_i.*

Like in the DMA algorithm, which is an adaptation of the Apriori algorithm, at k-th iteration, the set of candidate sets is obtained by applying the *Fuzzy_Apriori_Gen* function on $L_{(k-1)}$. We denote this set by $CA_{(k)}$. More exactly,

$$CA_{(k)} = \textit{Fuzzy\_Apriori\_Gen}(L_{(k-1)}).$$

For each site S_i, $(1 \leq i \leq n)$, we denote with $CG^i_{(k)}$ the set of candidate fuzzy itemsets generated applying *Fuzzy_Apriori_Gen* on $GL^i_{(k-1)}$, i.e.,

$$CG^i_{(k)} = \textit{Fuzzy\_Apriori\_Gen}(GL^i_{(k-1)}).$$

Because $GL^i_{(k-1)} \subseteq L_{(k-1)}$, then $CG^i_{(k)}$ is a subset of $CA_{(k)}$. Following, we denote $CG_{(k)} = \bigcup_{i=1}^{n} CG^i_{(k)}$.

Theorem 1. *For every $k > 1$, the set of all globally large k-fuzzy itemsets $L_{(k)}$ is a subset of $CG_{(k)} = \bigcup_{i=1}^{n} CG^i_{(k)}$.*

Applying the Theorem 1 the result is that we can use the set $CG_{(k)}$, which is a superset of $L_{(k)}$, as a candidate set instead of $CA_{(k)}$, and could be much smaller that $CA_{(k)}$.

Thus the candidate set for $L_{(k)}$ will be generated at k-th iteration in the following manner: first the set of candidate sets $CG^i_{(k)}$ can be generated locally at each site S_i. After this step, sites exchange fuzzy support count and compute the set of gl-large fuzzy itemsets $GL^i_{(k)}$. Based on $GL^i_{(k)}$, the candidate fuzzy itemsets at S_i for $(k+1)$-st iteration can then be generated.

3.2 Local pruning of candidate sets

The Lemma 3 can be used to perform a local pruning of the set of candidate fuzzy item sets. At a site S_i, after the set of candidate fuzzy itemsets $CG_{(k)}$ is generated, in order to find if a candidate fuzzy itemset $\langle X, F_X \rangle \in CG^i_{(k)}$ is gl-large fuzzy itemset, the fuzzy support count must be requested from all other sites. We can prune this request for fuzzy support count for some candidates using a local pruning technique. The basic idea is that at site S_i, if a candidate fuzzy itemset $\langle X, F_X \rangle \in CG^i_{(k)}$ is not locally large at site S_i, there is no need for S_i to compute global support to find out if it is globally large. This is possible because in this case, either $\langle X, F_X \rangle$ is not globally large, or it will be locally large at some other site, and hence only the sites where $\langle X, F_X \rangle$ is locally large need to be responsible to find its global support count. We use $LL^i_{(k)}$ to denote those fuzzy candidate items in $CG^i_{(k)}$ which are locally large at site S_i.

3.3 The algorithm outline

In Algorithm 1 is presented in detail the FUZZY-DMA algorithm for distributed mining of association

Algorithm 1 FUZZY-DMA

INPUT:

$\mathcal{DB}_1, \ldots, \mathcal{DB}_n$ - the database partition at each site.

$minsup$ - the minimum support threshold.

$\mathcal{F}$ - the set of fuzzy sets associated with attributes from $\mathcal{I}$.

OUTPUT:

L - the set of all globally large fuzzy itemsets in $\mathcal{DB}$.

METHOD: For all $k \geq 1$, iterates the following algorithm distributively at each site S_i. At the end of each step a synchronization is required to develop global count. The algorithm terminates when either $L_{(k)}$ returned is empty or candidate $CG_{(k)} = \emptyset$.

```
if $k = 1$ then
    $T^i_{(1)} = Get\_Local\_Fuzzy\_Count(DB_i, \emptyset, 1)$
else
    $CG_{(k)} = \cup^n_{i=1} CG^i_{(k)} =$
    $= \cup^n_{i=1} Fuzzy\_Apriori\_Gen(GL^i_{(k-1)})$
    $T^i_{(k)} = Get\_Local\_Fuzzy\_Count(DB_i, CG_{(k)}, i)$
for all $\langle X, A \rangle \in CG^i_{(k)}$ do
    if $CFS^i_{\langle X,A \rangle} \geq minsup \times D_i$ then
        insert $\langle X, A \rangle$ into $LL^i_{(k)}$
{Broadcast support count request to compute global fuzzy support count}
for $j = 1, \ldots, n; j \neq i$ do
    $Broadcast\_Count\_Request(LL^i_{(k)}, S_j)$
{Receive support count request}
for $j = 1, \ldots, n; j \neq i$ do
    receive $LL^j_{(k)}$ extract $CFS^i_{\langle X,A \rangle}$ from $T^i_{(k)}$ and send to $S_j$
{Compute global fuzzy support count}
for all $\langle X, A \rangle \in LL^i_{(k)}$ do
    receive $CFS^j_{\langle X,A \rangle}$ from sites $S_j$, where $j \neq i$
    $CFS_{\langle X,A \rangle} = \sum^n_{p=1} CFS^p_{\langle X,A \rangle}$
    if $CFS_{\langle X,A \rangle} \geq minsup \times D$ then
        insert $\langle X, A \rangle$ into $G^i_{(k)}$
broadcast $G^i_{(k)}$ {Compute global $L_{(k)}$}
receive $G^j_{(k)}$ from all other sites $S_j$, $(i \neq j)$
$L_{(k)} = \cup^n_{i=1} G^k_{(k)}$
return $L_{(k)}$
```

rules. At every iteration (k-th iteration), each site S_i computes the set of gl-large fuzzy itemsets $GL^i_{(k)}$ on the site, and from these computes the set of all globally large fuzzy itemsets $L_{(k)}$.

Initially, each site S_i generates the complete global candidates fuzzy itemsets $CG_{(k)}$ using the globally $(k-1)$-fuzzy itemsets, $L_{(k-1)}$, generated at the end of step $k-1$, and locally large candidate fuzzy itemsets based on gl-large fuzzy itemsets found at site S_i at $(k-1)$ step applying function *Fuzzy_Apriori_Gen* on $GL^i_{(k-1)}$ (candidate sets generation).

For each $\langle X, A \rangle \in CG_{(k)}$, scan the database $\mathcal{DB}_i$ to compute the local fuzzy support count $CFS_{\langle X,A \rangle}$ and store it into the hash tree $T^i_{(k)}$ using function *Get_Local_Fuzzy_Count*, and generate set of locally large fuzzy itemsets $LL^i_{(k)}$. After this, S_i broadcasts the candidate fuzzy itemsets from $LL^i_{(k)}$ to other sites to collect fuzzy support counts. The fuzzy support counts are needed to compute global support counts and generate set of all gl-large k-fuzzy itemsets at site S_i.

Finally computed gl-large fuzzy itemsets are broadcasted to all other sites, and these can compute $L_{(k)}$.

The algorithm is stopped when either $L_{(k)}$ returned is empty or candidate set $CG_{(k)}$ is empty.

4 CONCLUSION

In this article, it is proposed an algorithm for mining fuzzy association rules from distributed databases more efficiently than a sequential algorithm. In the future, we will study the means of automatically finding of fuzzy sets associated with database attributes. The other direction of improvement is related to the study of new relationships between local and global large itemsets in order to reduce the number of messages exchanged among sites.

REFERENCES

Agrawal R., Imiclinski T., S. A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD Conference Washington DC, USA.*

Chan Man Kuok, Ada Fu, M. H. W. (1998). Mining fuzzy association rules in databases. *SIGMOD Rec.*, 27(1):41–46.

Cheung D.W., Jiawei Han, N. V. F. A. Y. F. (1996). A fast distributed algorithm for mining association rules. In *In 4th International Conference on Parallel and Distributed Information Systems (PDIS '96)*, pages 31–43. IEEE Computer Society Technical Committee on Data Engineering, and ACM SIGMOD.

Gyenesei, A. (2000). Mining weighted association rules for fuzzy quantitative items. In *Principles of Data Mining and Knowledge Discovery*, pages 416–423.

Hong T.P., Kuo C.S., C. S. W. S. (2000). Mining fuzzy rules from quantitative data based on the apriotitid algorithm. In *Proceedings of the 2000 ACM symposium on Applied computing*, pages 534–536.

Rakesh Agrawal, R. S. (1994). Fast algorithms for mining association rules. In Bocca, J. B., Jarke, M., and Zaniolo, C., editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann.