Front. Electr. Electron. Eng. China (2006) 2: 177–182
DOI 10.1007/s 11460-006-0024-1

RESEARCH ARTICLE

**LU Jian-jiang, XU Bao-wen, ZOU Xiao-feng,
KANG Da-zhou, LI Yan-hui, ZHOU Jin**

Parallel mining and application of fuzzy association rules

Abstract Quantitative attributes are partitioned into several fuzzy sets by using fuzzy *c*-means algorithm. Fuzzy *c*-means algorithm can embody the actual distribution of the data, and fuzzy sets can soften the partition boundary. Then, we improve the search technology of apriori algorithm and present the algorithm for mining fuzzy association rules. As the database size becomes larger and larger, a better way is to mine fuzzy association rules in parallel. In the parallel mining algorithm, quantitative attributes are partitioned into several fuzzy sets by using parallel fuzzy *c*-means algorithm. Boolean parallel algorithm is improved to discover frequent fuzzy attribute set, and the fuzzy association rules with at least a minimum confidence are generated on all processors. The experiment results implemented on the distributed linked PC/workstation show that the parallel mining algorithm has fine scaleup, sizeup and speedup. Last, we discuss the application of fuzzy association rules in the classification. The example shows that the accuracy of classification systems of the fuzzy association rules is better than that of the two popular classification methods: C4.5 and CBA.

Keywords Data mining, Association rules, Fuzzy, Parallel, Classification

1 Introduction

Agrawal et al. presented the problem of mining Boolean association rules [1–2]. Srikant and Agrawal presented the problem of mining quantitative association rules [3]. This algorithm finds quantitative association rules by partitioning the attribute domain, and then transforming the problem into a binary one. Although this algorithm can solve the problems introduced by quantitative attributes, it brings the other two problems at the same time: 1) equi-depth partitioning cannot embody the actual distribution of the data. On the one hand, it may not work very well on highly-skewed data and tends to split adjacent values with high support into separate intervals though their behavior would typically be similar. On the other hand, it is not easy to distinguish the grade of membership. 2) The second problem is caused by the sharp boundary between intervals. The algorithm either ignores or over-emphasizes the elements near the boundary of the intervals in the mining process.

The contributions of this paper consist of three parts: 1) quantitative attributes are partitioned into fuzzy sets by fuzzy *c*-means (FCM) algorithm. FCM algorithm can embody the actual distribution of the data, and fuzzy sets can soften partition boundary. Then, we improve the search technology of apriori algorithm and present the algorithm for mining fuzzy association rules. 2) The parallel algorithm for mining fuzzy association rules is presented. In this algorithm, quantitative attributes are partitioned into several fuzzy sets by using parallel fuzzy *c*-means (PFCM) algorithm and Boolean parallel algorithm is improved to discover frequent fuzzy attribute sets, and the fuzzy association rules with at least a minimum confidence are generated on all processors. The parallel mining algorithm is implemented on the distributed linked PC/workstation. The experiment results show that the parallel mining algorithm has fine scaleup, sizeup and speedup. 3) We discuss the application of fuzzy association rules in the classification. The example shows that the accuracy of classification systems of the fuzzy association rules is better than that of the two popular classification methods: C4.5 and CBA.

2 Related works

Kuok et al. used fuzzy sets to soften partition boundary of

Translated from *Journal of Southeast University* (*Natural Science Edition*), 2005, 35(2): 165-170 (in Chinese)

LU Jian-jiang (✉), ZOU Xiao-feng
Institute of Command Automation, PLA University of Science and Technology, Nanjing 210007, China
E-mail: jjlu@seu.edu.cn
XU Bao-wen, KANG Da-zhou, LI Yan-hui, ZHOU Jin
Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China

the domains, and presented the concept of fuzzy association rules [4]. But he did not present the partitioning algorithm that can embody the actual distribution of the data, and the mining algorithm for fuzzy association rules cannot fit to the large databases. Lu et al. used the fuzzy clustering algorithm to partition the quantitative attributes into several fuzzy sets, and presented the algorithm for mining fuzzy association rules [5-6]. Au and Chan presented the problem of mining changes in fuzzy association rules [7].

As the database size becomes larger and larger, a better way is to mine association rules in parallel. Some parallel algorithms for mining Boolean association rules have been proposed in Ref. [8]. In this paper, quantitative attributes are partitioned into several fuzzy sets by using PFCM algorithm, and the parallel algorithm for mining fuzzy association rules is presented.

Classification is an important topic in data mining technology. The decision tree is the most popular approaches to solve the classification problem [9]. Association rules can also be used for classification (CBA) [10], when dealing with quantitative attributes; their domains are usually divided into equal-width or equal-frequency intervals [3]. But it will introduce some problems. The first problem is that equi-width partitioning cannot embody the actual distribution of the data. The second problem is caused by the sharp boundary. In most cases, the resulting intervals are not too meaningful and are hard to understand.

3 Mining fuzzy association rules

Let $T=\{t_1,t_2,\cdots,t_n\}$ be a relational database, t_j represent the jth record. Let $I=\{i_1,i_2,\cdots,i_m\}$ be the attribute set, where i_j denotes a Boolean, categorical or quantitative attribute, $t_j[i_k]$ represents the value of jth record in attribute i_k.

For mining fuzzy association rules, values of the record in attribute are first partitioned into several fuzzy sets. Boolean attribute and categorical attribute with fewer values can be partitioned into several fuzzy sets easily. Each quantitative attribute is partitioned into several fuzzy sets represented by triangular fuzzy numbers by using FCM algorithm [11].

We construct a new database through the database T to mine fuzzy association rules. In the new database, attributes are fuzzy sets, which are called fuzzy attributes next. Values of the record in fuzzy attributes are obtained as follows: let i_k^1 be a fuzzy set of i_k, and then i_k^1 is a fuzzy attribute in the new database. Value of the jth record in fuzzy attribute i_k^1 is $i_k^1(t_j[i_k])$, $i_k^1(t_j[i_k])$ is the grade of membership $t_j[i_k]$ with respect to i_k^1. All fuzzy attributes are still denoted as I, let $t_j(y_k)$ represent the value of the jth. record in fuzzy attribute y_k, $t_j(y_k)$ falls in [0, 1]. Let $X=\{y_1,y_2,\cdots,y_p\}\subset I$, $Y=\{y_{p+1},y_{p+1},\cdots,y_{p+q}\}\subset I$. A fuzzy association rule is an implication of the form $X\Rightarrow Y$. In order to mine association rules, we need to define the support and confidence of fuzzy association rules.

Definition 1 Let $X=\{y_1,y_2,\cdots,y_p\}\subset I$, the support of fuzzy attribute set X is defined as follows

$$\mathrm{Sup}(X)=\frac{\sum_{j=1}^{n}\prod_{m=1}^{p}t_j(y_m)}{n}$$

where $\sum_{j=1}^{n}\prod_{m=1}^{p}t_j(y_m)$ is called the support count of fuzzy attribute set X. Fuzzy attribute sets with at least a minimum support are called frequent fuzzy attribute sets.

Definition 2 The support of $X\Rightarrow Y$ is defined as follows

$$\mathrm{Sup}=\frac{\sum_{j=1}^{n}\prod_{m=1}^{p+q}t_j(y_m)}{n}$$

Definition 3 The confidence of $X\Rightarrow Y$ is defined as follows

$$\mathrm{Conf}=\frac{\mathrm{Sup}}{\mathrm{Sup}(X)}$$

The association rules with at least a minimum support and a minimum confidence respectively are called interesting fuzzy association rules. The interesting fuzzy association rules can be easily generated from frequent fuzzy attribute sets. Because $t_j(y_k)$ falls in [0, 1], we can easily know that all subsets of a frequent fuzzy attribute set must also be frequent according to Definition 1. With the above findings, it is easy to modify apriori algorithm [2] to discover all frequent fuzzy attribute sets.

4 Parallel mining fuzzy association rules

In this section, the parallel algorithm for mining fuzzy association rules is discussed. In the parallel mining algorithm, quantitative attributes are first partitioned into several fuzzy sets by using PFCM algorithm [12]. Secondly, the parallel algorithm for mining Boolean association rules is improved to discover frequent fuzzy attribute sets. Finally, the fuzzy association rules with at least a minimum confidence are generated on all processors.

4.1 Parallel partitioning the quantitative attributes

As the database size becomes larger and larger, FCM algorithm requires lots of computation power, main memory and disk I/O to partition quantitative attributes.

Lamehamedi et al. presented PFCM algorithm [12]. PFCM algorithm is developed following a master/slave approach. The computation is iterative and consists of slaves controlled by the master. In order to implement the parallel mining algorithm on the distributed linked PC/workstation, we improve the master/slave approach to the single program/multi data approach. Indeed, PFCM algorithm is shown in Algorithm 1.

Algorithm 1 PFCM algorithm

PFCM1. The values taken by each attribute are regarded as the initial set of patterns. To partition the initial set of patterns among the processes, each process will get *n*/*s* patterns. Where *n* is the number of patterns and *s* is the number of the processes launched.

$$\{x_1, x_2, \cdots, x_{n/s} \mid x_{(n/s)+1}, \cdots, x_{2n/s} \mid \cdots \mid x_{(s-1)(n/s)+1}, \cdots, x_n\}$$

Process 1 Process 2 Process s

PFCM2. Initialize the $v_i, i = 1, \cdots 2, \cdots, c$ on the root process, and broadcast them to all processes.

PFCM3. Each process receives $v_i, i = 1, 2, \cdots, c$, and computes the grade of memberships it holds. Each process j operates separately on its subset of data

$$\{x_k, k = (j-1)n/s + 1, \cdots, jn/s\}$$

This step is the end of the initialization part.

$$u_{ik,0} = \left[\sum_{j=1}^{c} (\| x_k - v_{i,0} \|_A / \| x_k - v_{j,0} \|_A)^{2(m-1)} \right]^{-1} \quad \forall i, k$$

For t=1 to T

PFCM4. Each process computes:

$$\alpha_{i,j} = \sum_{k=1}^{\text{size}x} (u_{ik,t-1})^m x_k$$

$$\beta_{i,j} = \sum_{k=1}^{\text{size}x} (u_{ik,t-1})^m$$

where $\text{size}x = n/s$ is the number of patterns received by each process.

PFCM5. Each process j sends these results ($\alpha_{i,j}$ and $\beta_{i,j}$) to the root process, which then aggregates to compute $v_{i,t}$ and broadcasts to all processes.

$$v_{i,t} = (\sum_{j=1}^{s} \alpha_{i,j}) / (\sum_{j=1}^{s} \beta_{i,j}), \quad i = 1, 2, \cdots, c$$

PFCM6. Each process receives the value of the cluster centers and computes the grade of membership. Each process operates separately on its subset of data.

$$u_{ik,t} = \left[\sum_{j=1}^{\text{size}x} (\| x_k - v_{i,t} \|_A / \| x_k - v_{j,t} \|_A)^{2(m-1)} \right]^{-1} \quad \forall i, k$$

PFCM7. Each process computes the error. The portion of the error at each process j is computed and then sent to the root process.

$$\text{error}_j = \sum_{k=1}^{\text{size}x} \sum_{i=1}^{c} (u_{ik,t-1} - u_{ik,t})^2 \qquad j = 1, 2, \cdots, s$$

PFCM8. The errors are aggregated at the root process:

$$E_t = (\sum_{j=1}^{s} \text{error}_j)^{1/2}$$

if $E_t < \varepsilon$, then stop.

4.2 Parallel mining fuzzy association rules

Agrawal and Shafer presented the parallel algorithm (Count Distribution) for parallel mining Boolean association rules[8]. The Count Distribution algorithm scales linearly and has excellent speedup and sizeup behavior with respect to the number of transactions. In this paper, we adopt the similar idea of the Count Distribution algorithm to design parallel algorithm for mining fuzzy association rules.

The key step to generate frequent fuzzy attribute sets is to obtain the whole support count by exchanging the local support count of the candidate fuzzy attribute sets. First, each process obtains local support count of the candidate fuzzy attribute sets from the hash tree asynchronously and put into the array LcntArr. Secondly, each process aggregates to compute the components of the array by function MPI_Reduce_Scatter () and broadcasts them to each process, which only receives the whole support count with respect to its partition. Third, each process gathers the whole support count of the candidate fuzzy attribute sets to the array GcntArr by function MPI_AllGatherv (). The parallel algorithm based on the Message Passing Interface (MPI) is shown in Algorithm 2.

Algorithm 2 The parallel mining algorithm

Input: Subset of data D_i ($i = 1, 2, \cdots, s$) ; minsup; minconf.

Output: Fuzzy Association Rules (FARs).

```
/* Parameter: nproc (the number of process), myid (process
   label), mysize (data size of the subset), pPMat (partial
   partition matrix), pFARs (partial fuzzy association rules) */
Methods:
{
  MPI_Init (&argc,&argv);
  MPI_Comm_size (MPI_COMM_WORLD, &nproc);
  MPI_Comm_rank (MPI_COMM_WORLD, &myid);
  Data_input (Di);
  MPI_Barrier (MPI_COMM_WORLD);
  If (myid==0)
      startwtime=MPI_Wtime ();
 for (cl=0; cl<COLUMN; cl++)
  /* Clustering by PFCM */
 {
     PFCM (Di, mysize, cl, pPMat[cl]);
  }
  /* Construct a new database */
  Transform (pPMat);
  istree=ist_create (m, minsup, minconf);
  /* Discover frequent fuzzy attribute sets */
  do {
     for (i=0;i<mysize;i++)
         ist_count (istree, pPMat, mysize, i);
```

```
        extract_count (istree, LCntArr);
       MPI_Reduce_scatter (LCntArr, PartGCntArr, PartSize,
           MPI_SUM, MPI_COMM_WORLD);
      MPI_Allgatherv (PartGCntArr, PartSize[myid], GCntArr,
          PartSize, disp, MPI_COMM_WORLD);
      Writeback_count (GCntArr, istree);
      l=ist_addlvl (istree);
        } while (l==0);
/* Generating fuzzy association rules*/
    Gen_rule (istree, pFARs);
    If (myid==0) {
         MPI_Gather (pFARs, FARs, MPI_COMM_WORLD);
          endwtime = MPI_Wtime ();
         printf ("wall clock time = %f\n",endwtime-startwtime);
                        }
    MPI_Finalize ();
    If (myid==0){
        Return FARs;
                        }
}
```

4.3 Experimental analysis

We implement our parallel algorithm to mining fuzzy association rules on the distributed linked PC/workstation. This workstation consists of six computers with 128 MB RAM, which are interconnected via a 10 M/100 M hub. We use the parallel message passing software MPICH1.2.4. The experiment is implemented on an abalone dataset from UCI Machine Learning Repository, which has nine attributes, 4 177 instances. We first copy the abalone dataset 32 times, and obtain a 5.72 MB dataset with 133 664 records, which is regarded as the initial dataset. The performance of scaleup, sizeup and speedup is analyzed.

To see how well our parallel algorithm handles large datasets when more computers are available, we perform scaleup experiment, where the dataset is copied from the initial dataset in direct proportion to the number of computers in the workstation. The number of maximal computers is set to 6. In the experiment, attributes are partitioned into three fuzzy sets. Let minimum support be 0.01, minimum confidence be 0.1. The performance results of scaleup are shown in Fig. 1. In addition to the absolute response times as the number of computers is increased, Fig. 2 also plots scaleup, which is the response time normalized with respect to the response time for a single computer. The pfarm curve is our parallel algorithm; another curve is the ideal condition.

We fix the size of the computers at 4 on the workstation, while increasing the dataset from 1.5 MB per computer to 9 MB. Fig. 3-4 show the performance results of sizeup. The results show sublinear performance for our parallel algorithm. Our parallel algorithm is actually more efficient as the dataset size is increased, since increasing the size of the dataset simply makes the noncommunication portion of the code take more time due to more I/O and more transaction processing. This will lead to reduce the percentage of the overall time spent in communication.

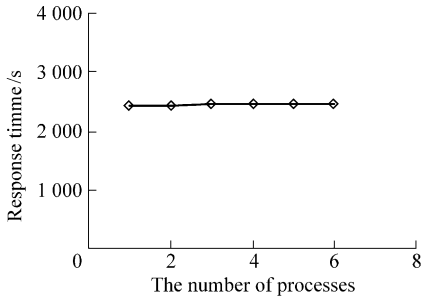

Fig. 1 Scaleup

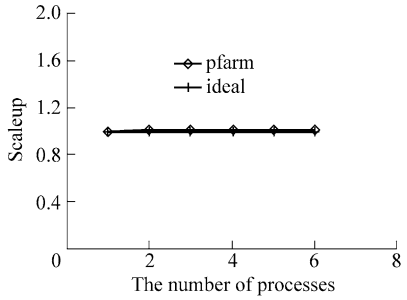

Fig. 2 Relative scaleup

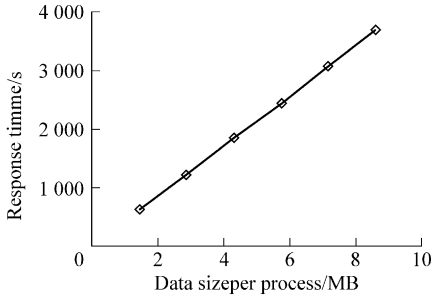

Fig. 3 Sizeup

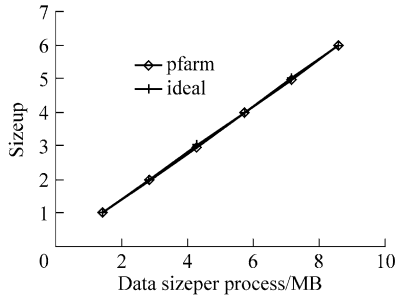

Fig. 4 Relative sizeup

We perform speedup experiments, for where we keep the dataset constant and vary the number of computers. We use the initial dataset (5.72 MB), and the number of maximal processors is set to 6. Fig. 5-6 show the performance results of speedup.

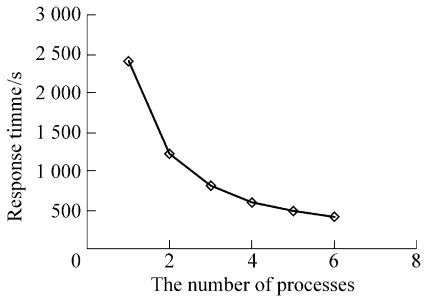

Fig. 5 Speedup

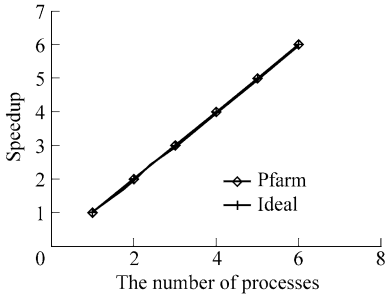

Fig. 6 Relative speedup

5 Classification system of fuzzy association rules

Classification is an important research issue of data mining technology. There are a number of classification methods [9-10]. Liu et al. proposed a classification method of the association rules (CBA) [10]; they partitioned quantitative attributes into several intervals for mining association rules. But the partition method of intervals introduces some problems. The first problem is that equi-depth partition cannot embody the actual distribution of the data. The second problem is caused by the sharp partition boundary. Next, we will use the fuzzy association rules to build the classification system.

Let $I=\{i_1,i_2,\cdots,i_m,i\}$ be the attribute set of classification databases. Attribute i is a categorical attribute with values $C_1,C_2,\cdots,C_q$, which are all class labels. Let $y=(y_1,y_2,\cdots,y_m)$ be a sample, where $y_1,y_2,\cdots,y_m$ are the values taken by attributes $i_1,i_2,\cdots,i_m$. In this section, we will discuss how to use fuzzy association rules to classify the sample y. We use interesting fuzzy association rules to build a classification system. Suppose that we use the algorithm in Sect. 3 to discover M interesting fuzzy association rules as following:

R_k : if i_1 is A_1^k and $\cdots$ and i_m is A_m^k, then i is $C_j, k=1,2,\cdots,M$ where $A_1^k,A_2^k,\cdots,A_m^k$ are fuzzy sets of attribute $i_1,i_2,\cdots,i_m$ respectively, C_j is the jth class label. We use these association rules to build the rule base of the classification system. When a sample y is to be classified, we can compute the discriminant function values of each class $g_h(y), h=1, 2,\cdots, q$ by the following formula

$$g_h(y)=\frac{\sum_{1\le k\le M, y=C_h}\prod_{j=1}^{m}A_j^k(y_j)}{\sum_{k=1}^{M}\prod_{j=1}^{m}A_j^k(y_j)}$$

We compare these discriminant function values, and take the class label corresponding to the maximum value as the classification result of the sample y. In this inference method, the information provided by each rule for sample classification is considered. At the same time, because fuzzy association rules are easy to be understood, the classification system has a better interpretability.

In order to check the accuracy of our classification system, this paper discusses Diabetes dataset from UCI Machine Learning Repository. Each quantitative attribute is partitioned into three fuzzy sets represented by triangular fuzzy numbers by FCM algorithm. In the experiment, ten-fold cross-validation method is applied to estimate the classification accuracy. The dataset is randomly divided into ten disjoint subsets, with each subset containing approximately the same number of records. Sampling is stratified by the class labels to ensure that the subset class proportions are roughly the same as those in the whole dataset. For each subset, a classifier is built using the records not in it. The classifier is then tested on the withheld subset to obtain a cross-validation estimate of its accuracy. Then, ten cross-validation estimates are averaged to provide an estimate for the classifier built from all the data. The cross-validation estimate in each subset is obtained as follows.

The number of interesting fuzzy association rules decides the complexity and accuracy of the classification system. A perfect classification system has a few rules and a good accuracy. In order to control the complexity of the classification system, we first mine 1 000 interesting association rules on the withheld subset and rank these rules by their support. Then, some rules that have high support are selected to evaluate the accuracy. In order to save the computing time, we select the number of rules at a multiple of 50, such as 50 100, 150 and so on. We select 20 times and select the number of rules with the best accuracy. Table 1 shows the experimental results with the classification system of fuzzy association rules (FARCS).

We compare FARCS with two popular classification methods: C4.5 and CBA. The algorithms of C4.5 and CBA are from http://www.cse.unsw.edu.au/~quanlian and http://www.comp.nus.edu.sg/~dm2. Where the minimum support is set to 1%, the minimum confidence is set to 50 %, and other parameters are unchanged. The accuracy of C4.5 is 74.2 %, CBA is 74.5 %, FARCS is 77.2 078 %. It is obvious that the accuracy of FARCS is better than CBA and C4.5 in the Diabetes dataset.

Table 1 Experimental results

Folds	Rule numbers	Training accuracy	Test accuracy
1	800	0.855 491	0.763 158
2	700	0.852 388	0.779 221
3	100	0.759 768	0.753 247
4	100	0.742 402	0.766 234
5	450	0.829 233	0.792 208
6	600	0.843 705	0.779 221
7	650	0.843 705	0.818 182
8	150	0.771 346	0.766 234
9	450	0.824 891	0.766 234
10	100	0.754 335	0.736 842
Average	410	0.807 726	0.772 078

6 Conclusions

We partition the quantitative attributes into several fuzzy sets by using FCM algorithm. FCM algorithm can embody the actual distribution of the data, and fuzzy sets can soften partition boundary. Then, we improve the search technology of apriori algorithm and present the algorithm for mining fuzzy association rules. We also present the parallel algorithm for mining fuzzy association rules in the large database. The parallel mining algorithm is implemented on the distributed linked PC/workstation. The experiment results show that the parallel mining algorithm has fine scaleup, sizeup and speedup. Finally, we discuss the application of fuzzy association rules in the classification. The example shows that accuracy of the classification systems based on the fuzzy association rules is better than that of the two popular classification methods: C4.5 and CBA.

Acknowledgements This study was supported by the National Key Basic Research Program 973 (2002CB312000), National Natural Science Funds for Distinguished Young Scholar (60425206), Advanced Armament Research Project (51406020105JB8103).

References

1. Agrawal R., Imieliski T., Swami A., Mining association rules between sets of items in large databases, in Proc. of ACM SIGMOD Conference on Management of Data, Washington, DC, 1993, 207–216
2. Agrawal R., Srikant R., Fast algorithms for mining association rules, in Proc. of the 1994 International Conference on Very Large Databases, Santiago, Chile, 1994, 487–499
3. Srikant R., Agrawal R., Mining quantitative association rules in large relational tables, in Proc. of the ACM -SIGMOD Conference on Management of Data, Montreal, Canada, 1996, 1–12
4. Kuok C.-M., Fu A.-W.-C., Wong M.-H., Mining fuzzy association rules in Databases, SIGMOD Record, 1998, 27(1): 41–46
5. Lu Jianjiang, Song Ziling, Qian Zuping, Mining linguistic valued association rules, Journal of Software., 2001, 12(4): 607–611 (in Chinese)
6. Lu Jianjiang, Xu Baowen, Xu Lei, et al., Mining association rules with linguistic terms, in Proc. the 15th IEEE International Conference on Tools with Artificial Intelligence, California, USA, 2003, 129–133
7. Au W.-H., Chan K.-C.-C., Mining changes in association rules: a fuzzy approach, Fuzzy Sets and Systems., 2005, 149(1): 87–104
8. Agrawal R., Shafer J.-C., Parallel mining of association rules: design, implementation and experience, Special Issue on Data Mining, IEEE Transactions on Knowledge and Data Engineering., 1996, 8(6): 962–969
9. Quinlan J.-R., C4.5: Programs for machine learning, San Mateo, CA, Morgan Kaufmann, 1993
10. Liu B., Hsu W., Ma Y., Integrating classification and association rule mining, in Proc. of the International Conference on Knowledge Discovery and Data Mining, USA, New York, 1998, 80–86
11. Hathaway R.-J., Davenport J.-W., Bezdek J.-C., Relational dual of the *c*-means algorithms, Pattern Recognition., 1989,22(2): 205–212
12. Lamehamedi H., Bensaid A.-D., Kebbal E.-G., Adaptive programming: Application to a semi-supervised point prototype clustering algorithm, in Proc. Of the International Conference on Parallel and Distributed Processing Techniques, Nevada, USA, 1999, 2753–2759