A Density-based Quantitative Attribute Partition Algorithm for Association Rule Mining on Industrial Database

Hui Cao, Gangquan Si, Yanbin Zhang, and Lixin Jia

Abstract— **Quantitative attribute partition is an important work of association rule mining, which is widely applied in industrial control at present, and the current partition methods are not suitable for the industrial database, which is generally large, high-dimensional and coupling. The paper proposes a density-based quantitative attribute partition algorithm for industrial database. The proposed algorithm uses an improved density-based clustering algorithm to detect the clusters. The clusters are agglomerated to form the new clusters according to the proximity between clusters and the new clusters are projected into the domains of the quantitative attributes. So the fuzzy sets and the membership functions used for partition are determined. We performed the experiments on a test database and a real industrial database. The experiments results verify the proposed algorithm not only can partition the quantitative attributes of industrial database successfully but also has the higher partition effectiveness.**

I. INTRODUCTION

ASSOCIATION rule mining is useful for discovering interesting association or correlation relationships hidden in large data sets [1]. For industry process, we can use it on field data to extract some unknown knowledge and many studies of this work have been introduced recently, such as, the identification of the fuzzy model of a controlled plant, the establishment of the control rules bases and the determination of the operation optimization values [2]-[6]. Therefore, the development of association rule mining will be benefit for a future control design of supervisory controller in order to optimize the plant performance.

Apriori is a classical algorithm of association rules mining and it is to the binary databases, in which values of each attribute are Boolean [7]. However, many variables of an industrial process are quantitative, so the algorithm can not be used directly. Reference [8] presents the quantitative association rules mining, which partitions the value of the quantitative attributes, and the quantitative association rule problem is mapped into the Boolean association rule problem. To deal with the problem "sharp boundary", which may under-emphasize or over-emphasize the elements near the boundaries of intervals, fuzzy sets are used in quantitative attribute partition process [9]. In this process, a database containing quantitative attributes is replaced by one with values from [0,1]. However, because the fuzzy sets are always provided by domain experts, it is difficult to know the fuzzy sets will be appropriate. Reference [10] presents that the fuzzy sets can be determined automatically from data by using clustering techniques and a model-based clustering algorithm, CLARANS, is adopted. Reference [11] and reference [12] use k-means and c-means, which are also the model-based clustering algorithm, to discover the clusters. Since there is not a suited model for industrial data, the algorithms can not be used effectively. Reference [13] adopts the CURE algorithm, which is a hierarchical clustering method, and it can not undo the completed step and correct erroneous decisions. Moreover, the clustering result of CURE mainly relies on the representative objects, and we can not estimate that the representative objects are correct. Reference [14] presents the notion of "density" and uses the grid-based method to capture the characteristics of quantitative attributes. Nevertheless, since creating a grid structure for high-dimension database is difficult, the algorithms can not be fit for the industrial database.

This paper proposes a density-based quantitative attribute partition algorithm for association rule mining on industrial database. The proposed algorithm uses an improved density-based clustering algorithm to detect the clusters in industrial database. The clusters are agglomerated to form the new clusters according to the proximity between clusters and the new clusters are projected into the domains of the quantitative attributes. So the fuzzy sets and the membership functions used for partition are determined. The organization of this paper is as follows. In section II, some related works are discussed. The proposed algorithm is explained in detail in section III. In section IV, the experiments are presented to verify the effectiveness and the practicability of the proposed algorithm. Finally, section V concludes the paper.

II. RELATE WORK

Density-based clustering locates region of high density that are separated from one another by regions of low density. DBSCAN is an effective density-based clustering algorithm and it is based on the concept of dense areas to form data clustering [15]. DBSCAN can handle the clusters of arbitrary shapes and it is relatively resistant to the outliers. Industrial database is used to record the values of the variables of an industrial process. If a variable is deemed as an attribute of an object, namely, a dimension of the database, the industrial database is high-dimensional because an industrial

The research is financially supported by National High Technology Research and Development Program of China (2006AA04Z180).

Hui Cao is with The Electrical Engineering School of Xi'an Jiao Tong University, Xi'an, Shaanxi, 710049, P. R. China (corresponding author to provide phone: +86-029-82668666-212; fax: +86-029-83237910; e-mail: huicao@mail.xjtu.edu.cn).

Gangquan Si is with The Electrical Engineering School of Xi'an Jiao Tong University (e-mail: gangquan@mail.xjtu.edu.cn).

Yanbin Zhang is with The Electrical Engineering School of Xi'an Jiao Tong University (e-mail: ybzhang@mail.xjtu.edu.cn).

Lixin Jia is with The Electrical Engineering School of Xi'an Jiao Tong University (e-mail: lxjia@mail.xjtu.edu.cn).

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

process generally includes many variables. Moreover, some variables of industrial process are nonlinear and coupling mutually. Therefore, the industrial database is large, high-dimensional, coupling and complex. Using DBSCAN on industrial databases directly would result in some mistakes because the distances between objects become more uniform in high dimensional data sets [16]. To deal with this issue, reference [16] uses shared nearest neighbor(SNN) similarity as the measure of distance metric between objects. Let p and q are two objects and if and only if they have each other in their k-nearest neighbor lists. Their SNN similarity is defined by the following equation:

$$SNNsimilarity = size(NN(p) \bigcap NN(q)) \tag{1}$$

where $NN(p)$ and $NN(q)$ are the k-nearest neighbor lists of p and q, respectively. $size(\bullet)$is the operator for calculating the size of data set.

SNN similarity provides us with a more robust measure of similarity and it works well for data with high dimension. However, in SNN similarity approach, calculating the k-nearest neighbor list of an object is based on the distance between objects and Euclidean distance are generally adopted as the distance metric, so the coupling of industrial database would affect the results. We cite some examples to explain that.

Let D be a three-dimension industrial database, and the three dimensions are d_1, d_2, and d_3, respectively. d_2 and d_3 are coupled with each other. An object in D can be written as $\{v_{d1}$, v_{d2}, $v_{d3}\}$, and v_{di} is the value of the i-th dimension of the object.

Example I: Let p, q_1, and q_2 are three objects in D, which are $\{0, 0, 0\}$, $\{0, 1, 1\}$ and $\{0, \sqrt{2}, 0\}$, respectively. Because $dist(p, q_1)$=$dist(p, q_2)$=2, q_1, and q_2 may be deemed as the neighbor of p. However, because d_2, and d_3 are coupled with each other, namely, in general, d_2, and d_3 may reflect the same system state in industrial process. Therefore, according to the experience of experts, q_1 is more "close" to p,. The reason of this contradiction is the duplicating calculation of d_2, and d_3.

Example II: Let p, q_1, and q_2 are three objects in D, and they are $\{0, 0, 0\}$, $\{1, 10, 10\}$ and $\{2, 9, 9\}$, respectively. $dist(p, q_1)$=14.18 and $dist(p, q_2)$=12.88, so we can estimate that q_2 is more close to p. Nevertheless, in industrial process, compared with other variables, some variables reflect the system state more clearly. In this example, q_1 is more close to p actually. The reason is that the absolute figure of v_{d2} and v_{d3} are larger than that of v_{d1}, so v_{d1} is almost "ignored" in the process of computing the distance between objects.

In general, after the clusters are detected, the fuzzy sets used for quantitative attribute partition are determined. Then, the clusters are projected into the domains of a quantitative attributes to identify the partition interval and establish the membership functions. However, since projecting the clusters would form some overlap intervals, identifying the membership value of an object based on the intervals will make some mistakes, and we cite example III to explain that.

Example III: D is a two-dimension databases and a and b are the two dimensions of D, that is shown in Fig.1. C_1 and C_2 are two clusters and projecting them into attribute a forms two intervals $[l_1, u_1]$ and $[l_2, u_2]$, respectively. c_1 and c_2 are two objects in C_1 and C_2, respectively. c_1^a and c_2^a are the projections of c_1 and c_2 in attribute a, respectively. Therefore, for the attribute a, the general algorithm will determine that the membership values of c_1 equals that of c_2. However, since they belong to different clusters, they need be distinguished in partition process, otherwise the results of future association rule mining will be wrong.

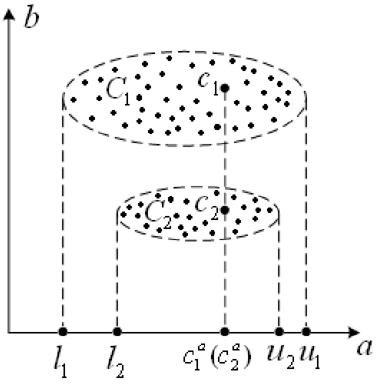

Fig. 1. Projecting the clusters into the attribute a.

III. THE ALGORITHM

In this section, we describe our algorithm for partitioning quantitative attribute. The proposed algorithm is divided into several phases, where phase I is a clustering process of the improved density-based clustering algorithm, phase II is agglomerating the clusters, detected in phase I, to form the new clusters and calculating the centroids of the new clusters, and phase III is establishing the membership functions based on the projections of the new clusters and identifying the membership values of each quantitative attribute of the objects.

Let D=$\{D_1, D_2, \cdots, D_n\}$ be an industrial database, n is the number of objects. D_i is an object in the database. If the number of dimensions of D is m, then D_i can be presented as follows:

$$D_i = \{v_i^i, v_2^i, \cdots, v_m^i\}$$

where v_i^j, $j = 1, 2, \cdots, m$, is the value of the j-th dimension of the ith object.

We assume that a and b, $a \in [1, m]$, $b \in [1, m]$, $a \neq b$, be a pair of coupling dimensions of D, where the coupling dimensions is the dimension which is coupled with other dimensions in industrial database.

Phase I

Step1: Calculating the fuzzy weighted distance between objects [17].

Let p and q are two objects in D, the weighted distance between them is defined by:

$$wdist(p,q)=\sqrt{\sum_{j=1}^{m} w_j(v_j^p-v_j^q)^2} \qquad (2)$$

where $j=1,2,\cdots,m$. w_j is weight of j-th dimension.

Although the weights of the dimensions are generally set beforehand, the weights of the coupling dimensions would be adjusted according to the objects in database, which reflects the working states of the industrial process. The approach of weight adjustment is based on the fuzzy control strategy.

Let v_a^p and v_a^q represent the value of p and q in a, respectively, then the absolute distance between of p and q in a is $ad_a^{pq}=|v_a^p-v_a^q|$. In like manner, ad_b^{pq} represents the absolute distance between p and q in b. AD_a and AD_b are linguistic variables of ad_a^{pq} and ad_b^{pq}, respectively, and the set of them is {ZO, PS, PM, PB}, where ZO, PS, PM, and PB represent zero, positive small, positive middle, and positive big, respectively.

Let $w_a^{''}$ and $w_b^{''}$ are the adjusted weights of a and b. $W_a^{''}$ and $W_b^{''}$ are linguistic variables of $w_a^{''}$ and $w_b^{''}$, respectively. The set of $W_a^{''}$ and $W_b^{''}$ are adopted as {NB, NS, ZO, PS, PB}, where NB, NS, ZO, PS, and PB represent negative big, negative small, zero, positive small, and positive big, respectively. The fields of AD_a and AD_b is [0,1] and that of $W_a^{''}$ and $W_b^{''}$ is [-1,1]. Based on the knowledge and the experience of experts, the discrete values of membership respected to them are assigned and a series of fuzzy logic rules can be obtained. For example,

Rules I: IF AD_a is PS and AD_b is PS, **THEN** $W_a^{''}$ is NB and $W_b^{''}$ is ZO.

The max-min algorithm is used in fuzzy logic inference, and the defuzzification is accomplished by the largest of maximum method. So, the querying table of adjusted weights can be calculated during the period of design, which is shown as Table I.

TABLE I

QUERYING TABLE OF ADJUSTED WEIGHTS

IF		THEN	
ad_a^{pq}	ad_b^{pq}	$w_a^{''}$	$w_b^{''}$
0	0	0	0
0	0.3	0	-0.5
0	0.7	0	-0.5
0	1.0	0	-1.0
0.3	0	-0.5	0
0.3	0.3	-1.0	0
0.3	0.7	-1.0	0.5
0.3	1.0	-1.0	1.0
0.3	0	-0.5	0
0.7	0.3	0.5	-1.0
0.7	0.7	-1.0	0
0.7	1.0	-1.0	0.5
1.0	0	-1.0	0
1.0	0.3	1.0	-1.0
1.0	0.7	0.5	-1.0
1.0	1.0	-1.0	0

In actual clustering process, $w_a^{''}$ and $w_b^{''}$ can be obtained directly by querying the table. The weights, w_a and w_b, can be figured out with the following equation:

$$w_j=w_j^{'}+k_{w_j}\cdot w_j^{''} \qquad (3)$$

where $j=a,b$, $w_j^{'}$ is the initial weight, and k_{w_j} is adjusted weight coefficients, which can let the range of $w_j^{'}$ to be set with considering actual demand. Finally, the fuzzy weighted distance between objects can be gotten.

Step2: According to the fuzzy weighted distance between objects, k-nearest neighbor lists of each object in D are established.

Step3: Calculating the SNN similarity between objects, and the pseudocode of this step as follow:

```
IF two objects, $x$ and $y$ are not among the $k$-nearest
    neighbor lists of each other THEN
    $SNNsimilarity(x,y)$=0
ELSE
    $SNNsimilarity(x,y)$=number of shared neighbors
END IF
```

Step4: Performing the clustering process, that is similar as that of DBSCAN. The algorithm begins with the arbitrary object o in industrial database D, and retrieves all neighbors of o, whose SNN similarity between them and o is greater than Eps. In the proposed algorithm, Eps is a threshold of SNN similarity between objects. If the number of neighbors is greater than $MinPts$, a cluster is created. The point o and its neighbors are assigned into this cluster. Then, let the neighbors be the expand objects to iterate the same process until all of the points have been labeled.

Step5: Collect the clusters and output.

Phase II

The improved density-based clustering algorithm, mentioned in phase I, is more suitable for industrial data and it provides a more reasonable basis for forming the fuzzy sets, which are used in quantitative attribute partition. However, the number of clusters detected by phase I is not limited. More is the number of clusters, more is that of linguistic terms. It would increase the time complexity of the future association rule mining. To treat with the problem, we agglomerate the clusters based on the proximity between clusters. The proximity between clusters usually includes three different metrics, MIN, MAX and Group Average, come from a graph-based view of clusters. MIN defines cluster proximity as the proximity between the closest two objects that are in different clusters. MAX takes the proximity between the farthest two objects in different clusters to be the cluster proximity, and Group Average defines cluster proximity to be the average pairwise proximities of all pairs of objects from different clusters. In phase II, we choose Group Average as the proximity between clusters and the proximities of a pairs of objects is the SNN similarity between objects, which has been calculated in phase I. The steps of phase II as follows:

Step1: If the number of clusters is greater than ck, then do the next step, otherwise, got to Step 5, where ck is a integer and the threshold of the number of clusters.

Step2: Calculating the Group Average between clusters.

Step3: Agglomerating the nearest two clusters to form the new cluster, and updating the information of the clusters.

Step4: Repeating Step1 to 3.

Step5: Calculating the centroids of the clusters.

Let the j-th cluster C_j=$\{d_1, d_2, \cdots d_i\}$ be a set of objects and s is the number of objects in C_j. The centroid of C_j is denfined as

$$C_j = \frac{1}{s}\sum_{r=1}^{s} d_r \tag{4}$$

Step6: Output the new clusters and the centroids of them.

Phase III

In phase II, a set of clusters $\{C_1, C_2, \cdots C_{ck}\}$ are detected, and in phase II, we use them as the linguistic terms for establishing the fuzzy membership functions, namely, the fuzzy set is $\{C_1, C_2, \cdots C_{ck}\}$.

We project the clusters into the domains of the h-th quantitative attribute, where $h \in [1, m]$, and the projections of the clusters will form the intervals. $[l_g^h, u_g^h]$ represents the interval formed by the projections of C_g in the h-th quantitative attribute. c_g is the centroid of C_g and its projection in the h-th quantitative attribute is c_g^h, where $g \in [1, ck]$ and $c_g^h \in [l_g^h, u_g^h]$. The triangular function is adopted as membership function, and l_g^h, u_g^h and c_g^h are the leftmost value, the rightmost value and the center value of the triangular membership function, respectively, that are shown in Fig.2. For the h-th quantitative attribute, the membership function of an object belonged to the linguistic term C_g can be defined as

$$f_{c_g}^h(x) = \begin{cases} (x - l_g^h)/(c_g^h - l_g^h) & l_g^h \le x \le c_g^h \\ (u_g^h - x)/(u_g^h - c_g^h) & c_g^h \le x \le u_g^h \end{cases} \tag{5}$$

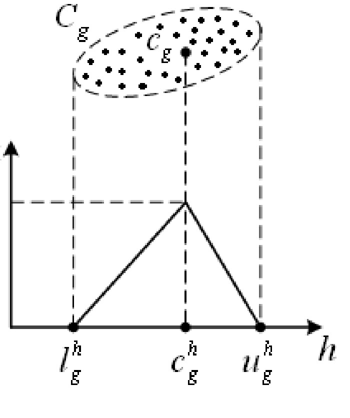

Fig. 2. Membership function of the linguistic term C_g in the h-th quantitative attribute.

To avoid the mistakes of example III in section II, we use the weighted membership function. For the h-th quantitative attribute, the weighted membership function of an object belonged to the linguistic term C_g is defined as

$$wf_{c_g}^h(x) = w_{c_g}^h \cdot f_{c_g}^h(x)$$

$$= \frac{SNNsimilarity(x, c_g)}{\sum_{r=1}^{ck} SNNsimilarity(x, c_r)} \cdot f_{c_g}^h(x) \tag{6}$$

where w_{c_g} is the weight of the linguistic term C_g and it represents the relationship degree between the object and the linguistic term C_g.

In the same way, we can establish the weighted membership functions of other linguistic terms in the h-th quantitative attribute, namely, we finish partitioning the h-th quantitative attribute. After all quantitative attributes in industrial database are partitioned and the weighted membership functions of them are established, so we can perform the quantitative association rule mining algorithm to find a set of association rules. For example,

IF attribute $a \in C_1$ and attribute $b \in C_2$, **THEN** attribute $c \in C_3$.

In the following, we analyze the time complexity of the proposed algorithm. The time complexity of calculating SNN similarity is $O(n^2)$, and the time complexity of querying table and recomputing the weights are both $O(1)$, which is smaller than that of other steps. Moreover, the complexity of DBSCAN is $O(n^2)$ also, so the time complexity of phase I is $O(n^2)$. The complexity of the agglomerating algorithm is $O(n^2)$ and the complexity of calculating the centroids is $O(n^2)$, then the time complexity of phase II is $O(n^2)$. In addition, the complexity of phase III is $O(n)$. Therefore, the time complexity of the proposed algorithm is $O(n^2)$.

IV. EXPERIMENT EVALUATION

To evaluate the proposed algorithm, experiments are done on a test database and a real industrial database. The test database is a three-dimension database and the number of objects of the database is 100. Moreover, the second dimension and the third dimension of the test database are a pair of coupling dimensions. The fuzzy c-means algorithm and the proposed algorithm are used on the test database and we compare the partition effectiveness based on the clustering results of them. At first, we use the fuzzy c-means algorithm on the test database and k is set as 3. The clustering result is illustrated in Fig.3. There are three clusters marked by "$+$", "$\times$", and "$\circ$", respectively, and the centroids of them are also marked. Since all objects are labeled as different cluster by this method, some far point, who have higher outlier-ness, affect the clusters and the calculating cluster centroids. The cluster "$+$" and the cluster "$\times$" are much near and some objects in boundary between the two clusters are not evaluated clearly. In addition, in fuzzy c-means algorithm, the initial objects chosen randomly also affect the final results.

On the same test database, we use the proposed algorithm to form the clusters. The initial weights of the first dimension, the second dimension, and the third dimension are 0.3, 0.35, and 0.35, respectively. The adjusted weight coefficients of the second dimension and the third dimension are both set as 0.35. Table I, which was shown in section III, is used as the querying table of adjusted weights. The

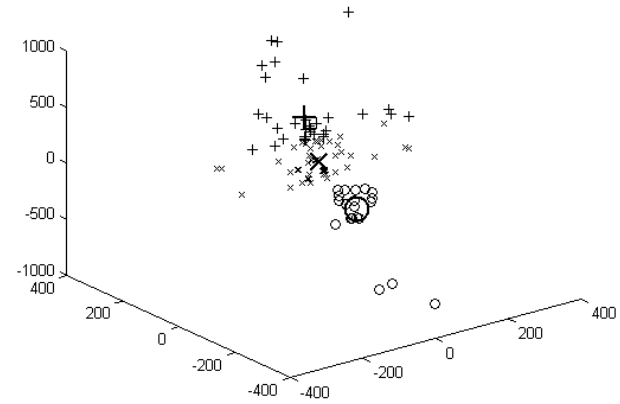

Fig. 3. Clustering result of fuzzy c-means algorithm on test database.

parameter k of k-nearest neighbor list is set as 4. Both Eps and $MinPts$ are 3, and ck is set as 3. The clustering result is shown in Fig.4. Three clusters are marked by "+", "×", and "○", respectively, and the centroids of them are also marked. Moreover, the outliers are marked by "•". Compared with Fig.3, the outliers can be detected and they can not impact the clustering result. The boundaries of three clusters are distinct and all objects are estimated correctly. Therefore, the clustering result of the proposed algorithm is more reasonable. Based on this clustering result, the partition for quantitative attributes would be more proper.

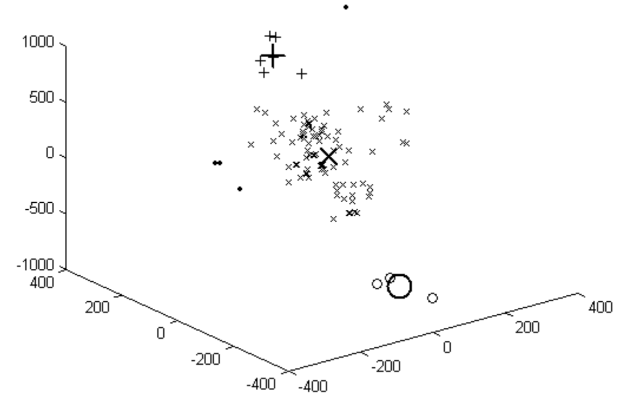

Fig. 4. Clustering result of the proposed algorithm on test database.

The experiments on a real industrial database are presented to verify the effectivity of the proposed algorithm. The real database is the database of the Pulverizing System of Power Plant, which is used to record the service data. To facilitate our experiment, we simplify the real database. The simplified database includes five dimensions, the ball mill load, the mill current, the outlet temperature, the entrance negative pressure, and the inlet-outlet pressure difference, and they are all quantitative attributes. The initial weights of them are set as 0.3, 0.2, 0.1, 0.2, and 0.2. Since the entrance negative pressure and the inlet-outlet pressure difference is a pair of coupling dimensions, the querying table of adjusted weights, which is shown in Table I, is used for them, and the adjusted weight coefficients of them are set as 0.2. The parameter k of k-nearest neighbor list is set as 4. Eps and $MinPts$ are 2 and 3, respectively, and ck is set as 3. We use the proposed algorithm on the simplified database with the number of objects increasing. During the experiment process, all parameters are not changed. The experiments results are shown in Table II. Since ck is 3, namely, there are three linguistic terms. We use $[l_i, u_i, c_i]$to represent the triangular membership function of the linguistic term RC_i, where $i = 1, 2, 3$ and l_i, u_i, and c_i represent the leftmost value, the rightmost value and the center value of the triangular membership function, respectively. The parameters of membership function of the ball mill load are shown in the second column of Table II and all the parameters are normalized. With the number of objects increasing, the partition of the ball mill load becomes more reasonable and the increasing degree of running time satisfies the analysis of the time complexity mentioned in section III. The experiments results show that the proposed algorithm can partition the quantitative attribute of industrial database successfully and do not increase the algorithm complexity.

TABLE II

EXPERIMENTS RESULTS OF THE PROPOSED ALGORITHM

Number of objects	Parameters of membership function of ball mill load $\{RC_1; RC_2; RC_3\}$	Running time (sec.)
100	{[0.1954, 0.9577, 0.3596]; [0.2844, 1, 0.6269]; [0, 0.7668, 0.2388]}	0.0870
200	{[0.2844, 1, 0.6269]; [0.1843, 0.9577, 0.2780]; [0.0158, 0.7668, 0.2472]}	0.4676
400	{[0.2844, 1, 0.6269]; [0, 0.7668, 0.2518]; [0.0897, 0.9577, 0.2493]}	3.2827
600	{[0.2011, 0.9577, 0.3532]; [0.2928, 1, 0.6840]; [0,0.5787, 0.2489]}	10.8088
800	{[0.2928, 1, 0.6373]; [0.0158,0.5787, 0.2472]; [0, 0.3764, 0.2398]}	25.3868
1000	{[0.2537,0.8980, 0.4418]; [0.4515, 1, 0.7287]; [0, 0.5117, 0.2514]}	49.0990
2000	{[0.0986, 0.2751, 0.1860]; [0.2278, 1, 0.7484]; [0, 0.9750,0.49137]}	380.5010
3000	{[0.7540, 1, 0.8651]; [0.2278, 0.9111,0.7354]; [0, 0.9767, 0.5466]}	1261.0932

V. CONCLUSIONS

In this paper, a density-based quantitative attribute partition algorithm for association rule mining on industrial database is proposed. The proposed algorithm has some advantages as follows. First, it can partition the quantitative attributes of industrial database effectively. Second, it uses the SNN similarity based on the fuzzy weighted distance between objects to instead of the general distance metric, and the approach is more suitable for the industrial databases. Third, it agglomerates the detected clusters to identify the fuzzy sets for quantitative attributes partition, and decreases the complexity of the future association rule mining process. Fourth, based on the projections of clusters, the proposed algorithm can establish the triangular membership functions of quantitative attributes conveniently. Fifth, in the proposed algorithm, the weighted membership functions is defined, and it determines the membership value of an object more reasonably. The experiments results also verify the effectiveness of the proposed algorithm and the time complexity are not increased. Since our work is partitioning the quantitative attributes of industrial database, we would integrate the proposed algorithm and the quantitative association rule mining algorithm to further verify the partition effectivety in our future work.

REFERENCES

[1] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, *Introduction to Data Mining*. Addison Wesley Higher Education, USA, 2006.

[2] XIU-QUAN CHEN, YONG-FU WANG, and XIAO-LING HUANG, "Fuzzy Modeling Method Based on Data Mining," *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, Guangzhou, P. R. China, 2005, pp. 1924-1930.

[3] Dong Liu, Yunping Chen, Youping Fan, and Guang Shen "The Application of Association Rule mining in Power System Restoration," *IEEE Power Engineering Society General Meeting*, Montréal, Quebec, Canada, 2006, pp. 74-78.

[4] LI Jian-qiang, LIU Ji-zhen, ZHANG Luan-ying, and NIU Cheng-lin, The Research and Application of Fuzzy Association Rule Mining in Power Plant Operation Optimization, *Proceedings of the CSEE*, vol. 26, no. 20, 2006, pp 118-123.

[5] Andrew Kusiak and Zhe Song, Combustion Efficiency Optimization and Virtual Testing: A Data-Mining Approach, *IEEE Transactions on Industrial Informatics*, vol. 2, no. 3, 2006, pp 176-184.

[6] Zhe Song and Andrew Kusiak, Constraint-Based Control of Boiler Efficiency: A Data-Mining Approach, *IEEE Transactions on Industrial Informatics*, vol. 3, no. 1, 2007, pp 73-83.

[7] Rakesh Agrawal and Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules," *Proceedings of the 20th International Conference on very Large Databases*, Santiago de Chile, Chile, 1994, pp. 487-499.

[8] Ramakrishnan Srikant and Rakesh Agrawal, "Mining Quantitative Association Rules in Large Relational Tables," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Montreal, Quebec, Canada, 1996, pp. 1-12.

[9] Jee-Hyong Lee and Hyung Lee-Kwang, "An Extension of Association Rules Using Fuzzy Sets," *Proceedings of the Seventh IPSA World Congress*, Seoul, Korea, 1997, pp. 399-402.

[10] Ada Wai-Chee Fu, Man Hon Wong, Siu Chun Sze, Wai Chiu Wong, Wai Lun Wong, and Wing Kwan Yu, "Finding Fuzzy Sets For The Mining of Fuzzy Association Rules for Numerical Attributes," *Proceedings of the First International Symposium on Intelligent Data Engineering and Learning*, Hong Kong, P. R. China, 1998, pp. 263-268.

[11] Peng Yan, Guoqing Chen, Fuzzy Quantitative Association Rules and Its Applications, *Fuzzy Applications in Industrial Engineering*, vol. 201, 2006, pp 573-587.

[12] Hannes Verlinde, Martine De Cock, and Raymond Boute, Fuzzy Versus Quantitative Association Rules: A Fair Data-Driven Comparison, *IEEE Transactions on system, Man, and Cybernetics-Part B: Cybernetics*, vol. 36, no. 3, 2007, pp 679-684.

[13] M. Kaya and R. Alhajj, Genetic Algorithm Based Framework for Mining Fuzzy Association Rules, *Fuzzy Sets and Systems*, vol. 152, no. 3, 2005, pp 587-601.

[14] Wang Lian, David W. Cheung, and S.M. Yiu, An Efficient Algorithm for Finding Dense Regions for Mining Quantitative Association Rules, *Computers & Mathematics with Applications*, vol. 50, no. 3-4, 2005, pp 471-490.

[15] Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proceedings of the International Conference on Knowledge Discovering in Databases and Data Mining*, Oregon, United States, 1996, pp. 221-231.

[16] Levent Ertöz, Michael Steinbach, and Vipin Kumar, "Finding Clusters of Different Sizes, Shapes, and Densities in Noisy, High Dimensional Data," *SIAM International Conference on Data Mining*, San Francisco, USA, 2003, pp. 47-58.

[17] Hui Cao, Gangquan Si, Yanbin Zhang, and Lixin Jia, "A Fuzzy Weighted Density-Based Clustering Algorithm for Industrial Databases," *Proceedings of the International Colloquium on Information Fusion*, Xian, P. R. China, 2007, pp. 386-392.