Clustering Method Study on High-Dimensional Trading Data

Peng Yong
Xi'an International University, Shaanxi Xi'an, China
Email: 18966805526@163.com

Di Li
University of Liverpool, Liverpool, Merseyside L69 3BX, United Kingdom
Email: creekline017@hotmail.com

Abstract—Existing clustering algorithms are not designed specially for the features of trading data and most clustering analyses lack scalability for large-scale transactions. Therefore, a rapid and scalable clustering algorithm using little space is proposed by us, to effectively process high-dimensional trading data without setting parameters manually. The improved method introduces weighted coverage density as similarity metrics of data. On this basis, the clustering criterion function is established for clustering analysis. We assume further implementation is to find association rules in clustering rules. Then two transaction-oriented evaluation measures for clustering quality are put forward. The large item size ratio is based on the concept of big data, which is used to measure the percentage in clustering; the average pair-clusters merging index is adopted to indicate the difference among clustering results with coverage density. The experimental results of artificial data and real data sets have shown that the improved method for clustering analysis can generate high-grade clustering results on most of the experimental data sets, compared to traditional algorithms.

Index Terms—Clustering; Coverage Density; Trading Data; SCALE; Dataset; Metrics

I. INTRODUCTION

With the development of information and Internet technology, people have entered the age of information explosion. We not only have huge data in quantity but the data types become more complicated, while the data structure is more diversified. Rapid development of free trading brings huge challenge to relevant data platform and decision support system [1,2]. For large-scaled huge data, traditional data analysis tools can only provide simple operations, while they cannot obtain internal relations and hidden information. In order to get out of the dilemma of rich data with poor knowledge, people are urgent for the technologies and tools that can intelligently and automatically convert data into useful information and knowledge [3]. Trading data is a kind of Boolean data and it usually has two obvious features: (1) Huge data quantity, such as the American retailer system Walmart. It can produce more than 200 million trading data in one day; (2) High data dimension. We can suppose that one trading data set $D$ which contains $N$ transactions and each transaction includes different items in quantity. If the item in trading data is seen as one attribute, and corresponding trading data is seen as one line, one trading dataset can be converted into a traditional category dataset and each attribute value is Boolean one. After the trading dataset is converted into category dataset, universal clustering algorithms of category data can be performed clustering analysis on conversed data [4]. Above two features of conversed data cannot make effective clustering analysis with current clustering algorithms of category data. If it is conversed into Boolean data, the data dimension will increase from dozens to thousands. Thus, whether current clustering algorithm of category data is effective to process high-quality data has become a great challenge. It is significant to develop professional, rapid and high-qualified clustering algorithm for trading data.

Recently there have emerged some clustering analysis algorithms for trading data with transaction features, such as LargeItem, CLOPE, and CCCD [5–7]. However, there are two obvious disadvantages in these algorithms. First, all these algorithms need users to manually adjust needed parameters from one to two algorithms, to change cluster quantity and seek optimum clustering results. LargeItem needs to set support and weight. There is a repulsion parameter in CLOPE and a threshold parameter of merged cluster in CCCD. These parameter settings need manual adjustment and the settings of different dataset are also different. It is very difficult to adjust the parameters, as is known, there is no algorithm to provide corresponding direction in current algorithms. Secondly, existing algorithms do not aim at data field application to propose quality evaluation method for clustering result. But clustering is an unsupervised process. Some general metrics or interactive visualization method based on statistical or geometrical features [8] have been applied to provide evaluation for numerical data clustering. However, for category data, due to the lack of distance function among meaningful data, the measuring based on entropy is taken as general metrics to be broadly implemented on clustering algorithms [9,10]. However,
evaluation metrics for trading data-oriented clustering result is still absent in current algorithms.

Emphasizing on above problems, this paper proposes a high-dimension trading data-oriented clustering analysis method, with the function of sampling, clustering structure assessment, domain-specific evaluation (SCALE for short). There are four consistent steps to complete the clustering analysis. At first, large dataset will be sampled. Then the clustering structure is estimated based on sampling set to generate candidate optimum Ks. At clustering analysis stage, the clustering algorithm based on weighted covering density can be used to accomplish clustering of large datasets. This step outputs candidate clustering result in small quantity. We mainly introduce weighted coverage density clustering algorithm to save storage rapidly and two specific clustering quality evaluation metrics of trading data: AMI and LISR. At the stage of clustering quality evaluation, domain specific metrics AMI and LISR are applied to estimate the quality of clustering results, so as to select the optimum clustering result. Experiment results show that our improved algorithm can efficiently generate trading data clustering results with high quality, under the assistance of SCALE structure.

II. ANALYSIS ON WEIGHTED COVERAGE DENSITY

A. Problem in Coverage Density

First, the centralized trading data in transaction dataset will be mapped in a 2D grid graph. We suppose that abscissa in grid graph represents the data item and ordinate represents trading serial number. Filled unit grid \((i, j)\) in grid diagram denotes that trading data \(t_j\) contains data item \(i\). When checking figure 1, we can see two naturally formed categories. They are \(\{abcd, bcd, ac\}\) and \(\{de, def\}\), which are indicated by two matrices in this figure. There are 16 unit grids not to be filled in original figure. But in two subgraphs, there are only four unit grids which are not filled. The problem of clustering trading data is transformed into how to divide the grid graph into proper number of subgraphs to minimize unfilled unit grid numbers. Similarly, if we attempt to use co-clustering algorithm [11] based on dual graph to divide the dataset, the division result is shown as the right two lines in figure 2. Obviously, co-clustering will lead to correlation lose between data item \(c\) and \(d\). It is defined as row-clustering of clustering problem under the background of this paper, instead of co-clustering. Above samples also show that when transaction has been arranged in some specific order, the clustering structure of trading dataset becomes very simple and visualized. Secondly, when two categories have the same coverage density but with different filling unit grid distribution, they will not be distinguished.

B. Weighted Coverage Density

The contribution of unit grid \((i, j)\) is made up of two parts: trading data contribution and data item contribution. In covering density, the trading data contribution and data item contribution are uniformed as \(T_i = T = \frac{1}{N_k}\) and \(W_j = W = \frac{1}{M_k}\). The coverage density can be express by another form:

\[
\rho(C_j) = T \sum_{j=1}^{M_k} \frac{\text{occur}(I_j) \times W_j}{S_k \times N_k}
\]  

(1)

Figure 1. 2D grid example

This paper proposes a heuristic rule: if two categories have the same coverage density, the unit filling distribution which concentrates on part data item is better than the even distribution that dispersed on each data item. Therefore, the definition of weighted coverage density is proposed by us. Specifically, high-frequency data item in category has more weighted value than low-frequency data item. It means that the weighted value of each data item is not fixed but it is determined by current data item among the categories during clustering. Therefore, data item contribution \(W_j\) is no longer uniformed and it is defined as proportion of each data item frequency and the occurring time data item. Its calculation formula is

\[
W_j = \frac{\text{occur}(I_j)}{S_k} \times \sum_{j=1}^{M_k} W_j = 1
\]  

(2)

Without changing the trading data contribution \(T\), the calculation formula of a category weighted coverage density can be acquired. We substitute formula 3 in formula 2 to get the result:

\[
\rho(C_j) = \frac{\sum_{j=1}^{M_k} \text{occur}(I_j)^2}{S_k \times N_k}
\]  

(3)

Therefore, the data item in right side of the graph is superior to that in the left side. It is consistent with previously proposed heuristic rule.

III. CLUSTERING ALGORITHM FOR HIGH-DIMENSIONAL TRADING DATA

A. Clustering Criterion Function

Weighted coverage density reflects the compactness inside trading data, and it is a kind of similarity metrics based on set. When defining the similarity of clustering result in one class, we should take into account the size of each class, instead of simple overlay or means. For a clustering result like \(C^t = \{C_1, C_2, ..., C_t\}\, t < N\), its expected weighted coverage density \(\rho^t\) works as the
clustering criterion function of the clustering algorithm. The calculation formula is

$$\rho'(C^i) = \frac{1}{N} \sum_{i=1}^{l} \sum_{j=1}^{m} \rho(C_j) = \frac{1}{N} \sum_{i=1}^{m} o(H_i) S_i$$  \hspace{1cm} (4)$$

One clustering algorithm based on expected weighted coverage density tends to maximize the value of clustering results. For example, there is a transaction set \{abc, abcd, abce, bdfg, dgh, dgi\}. Its possible clustering schemes include \{\{abc, abcd, abce\};\{bdfg, dgh, dgi\}\} and \{\{abc, abcd, abce\};\{bdfg, dgh, dgi\}\}. Figure 2 describes the original data set and item distribution of clustering results.

$$E(1) = \frac{2^2 + 2^2 + 4 + 1^2}{6} = \frac{2^2 + 4^2 + 2^2 + 1^2 + 1^2}{6} = 0.733$$

$$E(2) = \frac{1^2 + 2^2 + 1^2 + 1^2 + 1^2 + 1^2 + 3^2 + 3^2 + 1^2 + 1^2}{10} = 0.806$$

$E(2) < E(1)$ is consistent with aforementioned heuristic principles. Firstly, the unfilled cells of the second result are less than the first result; Secondly, the fourth data has more overlap with the frequent items in the second result. But if the expected weighted coverage density is the only metrics during clustering process, and the clustering amount of is not limited, then abnormal event will occur. So the clustering amount should be controlled by other parameters explicitly or implicitly. When weighted coverage density algorithm is used in the third step of SCALE structure [12], we can use clustering structure analysis and output candidate clustering amount to control the clustering amount.

B. Clustering Algorithm based on Weighted Coverage Density

The weighted coverage density algorithm adopts division-based strategy [13] to cluster high-dimensional data. It repeatedly scans the data and distributes them to the closest class, so as to maximize the expected weighted coverage density of clustering results. The whole process has two phases: initial clustering distribution phase and clustering adjustment repeatedly phase:

At the phase of initial clustering distribution, the algorithm uses the candidate clustering number output and the clustering seeds to generate an initial result, during clustering structure analysis. In detail, corresponding clustering seeds of candidate clustering number $K$ will form $K$ initial classes. Then weighted coverage density algorithm scan the data waiting for sampling to distribute them into one of $K$ classes, which can make the maximum expected weighted coverage density of current clustering result.

Since the clustering result generated at initial distribution phase is not optimum, the algorithm will repeatedly read and scan the data from the dataset, at the phase of clustering adjustment. It adjusts the data class label to raise the value of expected weighted coverage density as high as possible. In specific implementations, the algorithm reads each trading data randomly and checks whether the data of current class is optimum. If not, the data will be transmitted to the optimum class to make the largest increment of expected weighted coverage density value. The null class after one transmission is eliminated. When all the data finish scanning and there is not any data to be transmitted the program will stop; otherwise a new round of scanning begins. We should pay attention that the times of scanning make difference due to different dealing sequence, clustering structure and clustering amount. Related experiments have shown that the small dataset of most clustering structure need two or three times of adjustment, while that for large-scale dataset with noise need more iterations.

The pseudocode for weighted coverage density algorithm is described as Algorithm 1 shows. For the two stages of algorithm, the key step is to seek objective category for each trading data. Each possible distribution calculation or updating expected weighted coverage density value is needed during the seeking. In order to avoid unnecessary data access and calculation, weighted coverage density algorithm will store some statistical $I_i$. The statistical information of stored category $C_i$ includes: trading data quantity $N_i$ in $C_i$, data item quantity $M_j$, occurring times of all data items $S_i$ and square of occurring times of all data items, that is, $S_i^2 = \sum_{j=1}^{M_i} occur(I_{ij})^2$. $C_i$ contains data item set $I_i$ and occurring time of each data item is $I_i[j] occur$.
Algorithm 1 wcd.main()
Input: File $D$ of transaction; clustering amount $K$; initial seeds
Output: $K$ classes

1. Phase 1 - initial division */!
Initialize the seeds to form initial $K$ classes
While reading $D$ is not finished do
Read one piece of trading data $t$ from $D$;
Add $t$ to class $C_t$ to acquire the maximum expected weighted coverage density;
Write $<t, t>$ back to $D$;
End while

2. Phase 2 - adjust repeatedly */!
While Movemark=true do
Movemark=false;
Generate random sequence of reading file $R$;
while all the data is checked do
read $<t, t>$;
if expected weighted coverage density is raised by transmitting $t$ to $C_i$ and $i \neq j$ then
Movemark=true;
Write $<t, t>$ back to $D$;
End if
End while
End while

After above statistical information is recoded, the expected weighted coverage density can be calculated for current clustering result increment. Specifically, we design two sub-functions DeltaAdd and DeltaRemove to respectively compute the value of expected weighted coverage density, after one category increases a trading data or reduces a trading data. Because two functions are similar, the pseudocode of DeltaAdd function is given in following algorithm. We suppose $t, I$, is data item set of trading data $t$.

Algorithm 2 wcd.deltaAdd($C_t$, $t$)

Float deltaAdd($C_t$, $t$)

{ $\Delta S_{+}$ = $S_{+}$ + $|t|$; $\Delta S_{-}$ = 0;
for ($i = 0; i < |t|; i++$) {
if ($t[I[i]]$ not exist in $I_i$ then
$\Delta S_{+}$ = $\Delta S_{+}$ + ($I[I[j].occur] + 1)^2 - (I[I[j].occur])$
else
$\Delta S_{-} = (I[I[j].occur] + 1)^2 - (I[I[j].occur])$
}
return $((\Delta S_{+} + \Delta S_{-})/S_{-ocorr}) - (\Delta S_{+} / S_{+})$
}

C. Evaluation Method of Clustering Quality

Clustering is an unsupervised process to attempt to optimize one objective function. Its objective function may be domain specific, complex and even difficult to be assigned sometimes. Therefore, the design strategy of SCALE is not to optimize the clustering objective function simultaneously, but to be optimized by stage. In detail, the clustering stage of the whole large dataset will optimize objective function which has lower cost. While other metrics contain the domain specific metrics which is used to select the optimized result from a group of candidate clustering results.

Subsequent experiments use four kinds of candidate clustering results of the evaluation metrics. Two of them are domain specific metrics in trading data, that is, LISR and AMI, which are used in clustering quality evaluation stage in SCALE framework [14, 15].

The more large data item quantity in clustering results, the more possible to find out more frequent item sets in category. If occurring times of one dataset in category are higher than users’ assigned ratio value, it is called a large item. The ratio value assigned by user is defined as the minimum support. This definition has kind of relationship with the minimum support in association rule mining.

For one clustering result $C_t$, its calculation of Large Item Size Ratio is:

$$\alpha = \sum \frac{N_s}{N} \times \frac{LS_s}{S_t}$$ (5)

$LS_s$ denotes all occurring times of large data item and $S_t$ denotes occurring times of all the data item. Above formulas have considered the trading data quantity of each category, to reduce the influence of small noise category on the whole clustering result. In practical application, users can offer their interested minimum support to seek association rules. Then, according to these support degrees, different clustering LISR value can be compared to select the most interested one clustering result as final result. Average pair-clusters merging index is used to measure dissimilarity degree between categories. As is shown by previous contents, weighted covering density metrics evaluates homogeneity inside categories and attempts to store more frequent item sets. AMI is a metrics based on covering density which reflects the structural difference between categories.

Assume there are two class $C_i$ and $C_j$, the dissimilarity between them is

$$d(C_i, C_j) = \frac{N_i}{N_i + N_j} \rho(C_i) + \frac{N_j}{N_i + N_j} \rho(C_j) - \rho(C_i \cup C_j)$$ (6)

Formula 7 can be simplified as

$$d(C_i, C_j) = \frac{(S_i(\frac{1}{M_i} - \frac{1}{M_y}) + S_j(\frac{1}{M_j} - \frac{1}{M_y}))}{N_i + N_j}$$ (7)

$M_y$ denotes the amount of different data items after two classes are merged and $M_y \geq \max\{M_i, M_j\}$.

Since $\frac{1}{M_y} \leq \frac{1}{M_i}$ and $\frac{1}{M_y} \leq \frac{1}{M_j}$, $d(C_i, C_j)$ must be a real number in [0,1]. $S_i(\frac{1}{M_i} - \frac{1}{M_y})$ represents the structure change caused by combined class $C_t$. Obviously, when two classes have the same data items, that is, $M_i = M_j = M_y$ and the value of $d(C_i, C_j)$ is 0. For two very different classes whose data item overlap are less,
the combination will acquire a large \(d(C_i, C_j)\). Therefore, above metrics reflect the structure difference among the classes. Figure 3 and 4 describe this situation.

Based on the dissimilarity metrics of the above, this paper proposes AMI to evaluate the integral dissimilarity of clustering results with K-division.

Traditional clustering algorithm [16-18] attempts to combine within-class similarity with between-class dissimilarity to form quality evaluation and optimization. However, it is very difficult in practical application because users need to assign some domain specific weighted parameter to realize this combination. Thus, SCALE adopts the hierarchical optimization, that is, the value of expected weighted coverage density in category is optimized. But AMI value is used at clustering result evaluation stage.

IV. EXPERIMENTAL ANALYSIS

A. Performance Test

Large-scale artificial data and real data are used to test the performance of weighted coverage density. Because the improved algorithm saves more memory space, the test mainly focuses on related factors of time complexity. The time complexity is \(O(\lambda \times N \times K \times |r|)\). Since the times of adjustment \(\lambda\) is uncontrollable, the testing experiments only study three factors: amount of trading data \(N\), amount of clustering amount \(K\) and the average length of data \(|r|\). The following experiment result describe that weighted coverage density algorithm has scalability for three factors. The work is first performed on 5 groups of TxI4Dx series of dataset. The size of dataset is from 100K to 500K and the amount of clustering is \(K = \{10,50,100\}\). There are dataset generated randomly by the same parameters in each data.

So the running time of each group is the average time of 10 dataset. Figure 5 and 6 show the work of weighted coverage density algorithm nearly has a linear relationship with the number clustering and the size of dataset.

We test the relation between running time and the average length of data on other 4 groups of TxI4DlooK series dataset. In figure 7, for the small clustering amount such as \(K \leq 50\), we find the running time has approximate linear relationship with the average length of trading data; when the amount of clustering becomes large, such as \(K = 100\), the relationship becomes nonlinear. People have more interest on small amount of clustering, so the average of weighted coverage density algorithm is scalable for data.

CLOPE method is tested on 10%, 50% and 100% of Mushroom [19] 100k, \(r = 2.0\). The acquired clustering amount is \(\{22,23,30\}\). Then the clustering amount is used to run weighted coverage density algorithm on corresponding dataset, to acquire the running time. Figure 8 shows the improved algorithm is much faster than CLOPE.

We test the performance of weighted coverage density algorithm on real dataset and its clustering amount changes from 2 to 100. The relation between clustering amount and time is described as figure 9. It testifies previous analyses on time complexity. Because the source code provided by the author of CLOPE can not handle changing data, the weighted coverage density algorithm can not be compared to CLOPE on Retail dataset.
B. Quality Evaluation for Clustering Results

We adopt three metrics: LISR, AMI and expected entropy to evaluate the clustering result quality, generated by improved algorithm and by CLOPE. The experiments show that: SCALE-assisted algorithm has higher clustering quality than that of CLOPE. Apriori Net is also used to find frequent itemsets in the results of two schemes. It finds that SCALE-assisted algorithm has more amount of frequent item sets than that of CLOPE. Figure 10 describes LISR value under different minimum support degrees, in [0.6-10]. As for reserved large data item, the more value of LISR is, the better quality we can acquire from the results. LISR curves explain the clustering results in weighted coverage density algorithm are prior to CLOPE, especially for some higher support degrees.

When drawing the AMI index graph for the results of different clustering amount for some datasets, its pointed optimum clustering amount is consistent with that suggested by BKPlot method [22]. Figure 11 describes the AMI curve of Tc30a6r and Zoo respectively with different $K$. The global highest peak means there is the biggest dissimilarity between classes and it exists in “Ks” recommended by BKPlot. These optimum values are also consistent with predefined amount of categories.

We choose the best clustering results of two schemes and calculate the value of LISR, AMI and expected entropy. As shown in table 1: compared to the result of CLOPE, SCALE-assisted algorithm has smaller expected entropy, which means higher similarity inside the classes. Because LISR does not directly influence the amount of frequent items in clustering results, our experiments directly use Apriori Net to look for and gather statistics of

<table>
<thead>
<tr>
<th>DataSets</th>
<th>Method</th>
<th>LISR (MinSup=0.9)</th>
<th>AMI</th>
<th>Expected Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tc30a6r1000.2L</td>
<td>SCALE</td>
<td>0.866901</td>
<td>0.161347</td>
<td>5.319539</td>
</tr>
<tr>
<td></td>
<td>CLOPE</td>
<td>0.703372</td>
<td>0.094432</td>
<td>5.472992</td>
</tr>
<tr>
<td>Zoo</td>
<td>SCALE</td>
<td>0.704826</td>
<td>0.120255</td>
<td>4.281075</td>
</tr>
<tr>
<td></td>
<td>CLOPE</td>
<td>0.536510</td>
<td>0.060840</td>
<td>4.399753</td>
</tr>
<tr>
<td>Mushroom</td>
<td>SCALE</td>
<td>0.680028</td>
<td>0.129066</td>
<td>4.872726</td>
</tr>
<tr>
<td></td>
<td>CLOPE</td>
<td>0.644401</td>
<td>0.105209</td>
<td>4.776457</td>
</tr>
</tbody>
</table>

When drawing the AMI index graph for the results of different clustering amount for some datasets, its pointed optimum clustering amount is consistent with that suggested by BKPlot method [22]. Figure 11 describes the AMI curve of Tc30a6r and Zoo respectively with different $K$. The global highest peak means there is the biggest dissimilarity between classes and it exists in “Ks” recommended by BKPlot. These optimum values are also consistent with predefined amount of categories.
the frequent itemsets, which are generated by weighted coverage density algorithm and CLOPE. In specific operation, SCALE and CLOPE will generate the clustering results of Mushroom with \( k = 2 \), then frequent itemsets mining algorithm is processed. In short, under the structure of SCALE, we can acquire better clustering results than CLOPE and it can be finished under smaller parameter searching space.

V. CONCLUSION

Dataset similarity metrics based on dataset for trading data are provided in this paper: the covering density and weighted covering density. In addition, two metrics and important definition relevance in statistics are analyzed in theory. Besides, taking into account the weighted coverage density design and implementation of trading data clustering algorithm, this algorithm is verified to be rapid with small memory requirement. So it is very suitable to process large-scaled trading data with scalability. SCALE design has following characteristics:

- Covering density and weighted coverage density are introduced based on trading data. Two metrics can efficiently measure the overall similarity on a group of trading data.
- Weighted coverage density design and expected weighted coverage density trading data clustering algorithm can be completed.
- The clustering structure estimation is introduced in SCALE structure to efficiently reduce the search in clustering algorithm parameter space. Therefore, weighted coverage density algorithm does not need manual allocation parameters.
- Clustering result estimation in SCALE structure proposes that domain specific metrics selects the optimal clustering result. It indicates that domain specific metrics is essential to capture the meaning of trading data clustering analysis. Experiment results show that weighted coverage density algorithm under SCALE structure can efficiently generate trading data clustering result in high quality.

Our method can be further expanded to other clustering analysis of various datatype, to create a general platform for large scale data clustering. Current prototype system is only suitable for trading data, so the common platform of the next generation should be integrated with other types of data clustering analysis algorithms, clustering structure detection algorithms and clustering results evaluation metrics. A common platform for the next generation of integration with other data types should integrate clustering algorithm, clustering structure detection algorithms and clustering results evaluation metrics. The scheme can even be used to support customization capabilities, which means that the algorithms of each component under this structure can be specified by users.

REFERENCES


Peng Yong was born in Shaanxi China, October, 1978. He was awarded Master Degree in National Economics at Northwest University of China in 2012. His research interests are in the area of international trade and national economics. Mr. Peng has been keen on researching what he is interested in and devoted his most time and energy to it. He is often presented the title of the best teacher in college because of his excellent performance and challenging spirit.

Di Li was born in Chengdu China. He received his PhD. degree and M.Sc degree in Electrical Engineering at the University of Liverpool, the UK in 2010 and 2005, respectively. His current major field of study includes wireless networks, radio propagation and antennas.