An Effective Clustering Algorithm for Transaction Databases Based on K-Mean

Dingrong Yuan
College of Computer Science and Information Technology, Guangxi Normal University, Guilin, China
Faculty of Information Technology, University of Technology, Sydney, P.O. Box 123, Broadway NSW 2007, Australia
E-mail: dryuan@mailbox.gxnu.edu.cn

Yuwei Cuan and Yaqiong Liu
College of Computer Science and Information Technology, Guangxi Normal University, Guilin, China
E-mail: cuanyuwei@163.com; yaqiongliu99@163.com

***Abstract*—Clustering is an important technique in machine learning, which has been successfully applied in many applications such as text and webpage classifications, but less in transaction database classification. A large organization usually has many branches and accumulates a huge amount of data in their branch databases called multi-databases. At present, the best way of mining multi-databases is, first, to classify them into different classes. In this paper, we redefine related concepts of transaction database clustering, and then in connection to the traditional clustering method, we propose a strategy of clustering transaction databases based on the k-mean. To prove that our strategy is effective and efficient, we implement the proposed algorithms. The results showed that the method of clustering transaction databases based on the k-mean is better than present methods.**

***Index Terms*—transaction databases, database clustering, k-mean, multi-database**

I. Introduction

With the development of networks and database technology, large datasets have accumulated in various industries. The manner of acquisition of knowledge from these datasets has become a popular topic, while multiple source database mining has been considered as a challenging subject in data mining. So far, many studies had been conducted, including Wu and Zhang who advocated an approach for identifying interesting patterns hidden in multi-databases by weighting [1]. Liu et al. proposed a multi-database mining technique that can search relevant databases [2]. The multiple-database mining mode is divided into local, high votes, and exception modes. Zhang et al. recommended this model in their paper [3].

Clustering techniques mainly include hierarchical and classificatory clustering. Cohesion and k-mean algorithms are representations of two clustering methods [4], and these techniques have been successfully applied to cluster numeric and text databases, among others.

Unlike text and webpages, the value of attributes in transaction databases is Boolean, such as in transaction records of supermarkets or banks. The traditional clustering approach may lose its effectiveness or may be incorrect when applied to transaction databases. Therefore, Wu et al. proposed a transaction database classification method based on similarity [5]. Animesh Adhikari et al. and Yuan et al. subsequently investigated the problem further [6, 7]. They respectively introduced different strategies for transaction database classification. After comparing and analyzing these studies, we find that their strategies are effective but inefficient. In this paper, we propose a new strategy for classifying transaction databases.

The rest of this paper is organized as follows. In Section II, we introduce the related concepts of transaction database classification. Section III proposes a clustering strategy based on the k-mean, and then the related algorithm is designed. In Section IV, we conduct a number of experiments to validate our strategy. The summary is presented in Section V.

II. Problem Description

Let $D = \{D_i \mid i=1,2,...,n\}$ be a transaction database set called a multi-database. $I(D_i)=\{I_j \mid j=1,2,...,k\}$ is the item set of D_i. To cluster databases in D, several classes are clustered according to a strategy.

For example, let $D=\{D_1,D_2,D_3,D_4,D_5\}$ be a transaction database set, which is composed of five different databases: $D_1=\{a, b, c, d\}$, $D_2=\{b, c, d\}$, $D_3=\{e\}$, $D_4=\{f, h\}$, and $D_5=\{f, g, h\}$. The five databases can be clustered into three classes as $cluster(D)=\{\{D_1,D_2\},\{D_3\},\{D_4,D_5\}\}$.

Clustering transaction databases is one of the techniques in multi-database classification. The related concepts are as follows.

Definition 1: Let $I(D_i)=\{I_j \mid j=1,2,...,k\}$ be the item sets of database D_i. The support of I_j is $sup(I_j)$. Under the

Manuscript received May 15, 2013; revised September 28, 2013;

Corresponding author: Dingrong Yuan, College of Computer Science and Information Technology, Guangxi Normal University, Guilin, 541004, China.

doi:10.4304/jcp.9.4.812-816

supporting threshold α, the definition of frequent item sets of D_i is as follows:

$$FI(D_i)^{\alpha} = \{I_j \mid I_j \in I(D_i), \sup(I_j) \geq \alpha\}, \quad \alpha \in [0,1]$$

TABLE I.
DATABASE SET

database	$I(D_i)$ and $sup(I_i)$
D_1: {(a,b),(b)}	a:0.5, b:1.0, (a,b):0.5
D_2: {(b,c),(a,c),(a)}	a:0.6, b:0.3, c:0.6, (a,b):0.0, (b,c):0.3, (a,c):0.3, (a,b,c):0.0
D_3: {(a,b,c),(a,c),(b,d),(a,c,d)}	a:0.7, b:0.5, c:0.7, d:0.5, (a,b):0.2, (b,c):0.2, (a,c):0.7, (a,d):0.2, (b,d):0.2, (c,d):0.2, (a,b,c):0.2, (a,b,d):0.0, (b,c,d):0.0, (a,c,d):0.2, (a,b,c,d):0.0

To illustrate, the given databases are shown in TABLE I.

If α is 0.5,
$FI(D_1)^{0.5}$={a:0.5,b:1.0,(a, b):0.5},
$FI(D_2)^{0.5}$={a:0.6,c:0.6},
$FI(D_3)^{0.5}$={a:0.7,b:0.5,c:0.7,d:0.5,(a, c):0.7}.

Definition 2: Let $D=\{D_1, D_2, ..., D_n\}$ be a set of transaction databases. Under the supporting threshold α, the definition of distance between D_1 and D_2 is:

$$dis(D_1,D_2)^{\alpha} = 1 - \frac{\sum_{X \in \{FI(D_1)^{\alpha} \cap FI(D_2)^{\alpha}\}} \text{mininum}\{\sup(X,D_1), \sup(X,D_2)\}}{\sum_{X \in \{FI(D_1)^{\alpha} \cup FI(D_2)^{\alpha}\}} \text{maxinum}\{\sup(X,D_1), \sup(X,D_2)\}}$$

where ∩ and ∪ denote the intersection and union of two sets, respectively.

If X does not belong to D_i, then $sup(X, D_i) = 0$.

As an example, in TABLE I, we have
$dis(D_1,D_2)^{0.5}$=1-0.185=0.815
$dis(D_1,D_3)^{0.5}$=1-0.243=0.757

III. TRANSACTION DATABASE CLUSTERING

In this section, we define the related concepts of transaction database clustering, and then propose a clustering strategy for transaction databases based on the k-mean. The related algorithm is also designed in this section.

A. Related Concepts

The purpose of database clustering is to classify transaction databases into different classes. First, we define the related concepts as follows.

Definition 3: Let $D=\{D_1, D_2, ..., D_n\}$ be a set of transaction databases. Under the supporting threshold α, the definition of a class in D is defined as:

$$class(D) = \{D_i \mid D_i \in D\}$$

For example, let $D=\{D_1,D_2,D_3,D_4,D_5\}$ be a set of transaction databases. D_1,D_2,D_5 can then be clustered as a class, which is denoted as $class(D)=\{D_1,D_2,D_5\}$. Thus, $class(D)$ is a class of multi-database set D.

Definition 4: Let $D=\{D_1, D_2, ..., D_n\}$ be a set of transaction databases. Under the supporting threshold α, an *m-cluster* of D is defined as follows:

$$cluster(D)^m = \{class_k(D) \mid k = 1,2,...,m\} \quad , m \in [1,n]$$

which satisfies

(1) $class_1(D) \cup class_2(D) \cup ... \cup class_m(D) = D$

(2) $class_i(D) \cap class_j(D) = \varnothing, \ i,j \in \{1,2,...m\}$

From the above formula, the total number of clustering can be deduced as

$$total-number(D) = 1^n/1! + 2^n/2! + ... + m^n/m!$$

where $n=|D|$ and m is the number of classes after clustering.

In all these classes, the best classification should be selected as the final classification of the multi-database. Consequently, we define the measurement of the best complete classification as follows.

Definition 5: Let $D=\{D_1, D_2, ..., D_n\}$ be a set of transaction databases and $cluster^m(D)=\{class_1(D), class_2(D), ..., class_m(D)\}$ be an *m-cluster* of D. The inner distance of the classification is defined as follows:

$$\text{innerDis}(cluster^m(D)) = \frac{\sum_{class_i \in cluster^m(D)} \frac{\sum_{D_p \in class_i, D_q \in class_i, D_p \neq D_q} dis(D_p,D_q)}{|class_i| \cdot (|class_i|-1)/2}}{m}$$

where $|class_i|$ denotes the number of database in $class_i$. Inner class distance reflects the average cohesion degree of a classification. Cohesion is the mean distance between databases in one class.

Definition 6: Let $D=\{D_1, D_2, ..., D_n\}$ be a set of transaction databases, and $cluster^m(D)=\{class_1(D), class_2(D), ..., class_m(D)\}$ be an *m-cluster* of D. The outer distance of the clustering is defined as follows:

$$\text{outerDis}(cluster^m(D)) = \frac{\sum_{class_i, class_j \in cluster^m(D)} \frac{\sum_{D_p \in class_i, D_q \in class_j} dis(D_p,D_q)}{|class_i| \cdot |class_j|}}{m(m-1)/2}$$

where $|class_i|$ and $|class_j|$ denote the number of item sets of $class_i$ and $class_j$, respectively. Outer distance reflects the average coupling degree of a classification. Coupling is the mean distance between databases in different classes.

Usually, we consider a classification with smaller *innerDis* and larger *outterDis* as a better one. In light of this, we evaluate a classification based on its cohesion and coupling.

Definition 7: Let $D=\{D_1, D_2, ..., D_n\}$ be a set of transaction databases, and $cluster^m(D)=\{class_1(D), class_2(D), ..., class_m(D)\}$ be an *m-cluster* of D. The goodness of the clustering is defined as follows:

$$\text{goodness}(cluster^m(D)) = \text{outerDis}(cluster^m(D)) - \text{innerDis}(cluster^m(D)) - \frac{m}{n}$$

The discussion above illustrates that the best clustering holds the highest value of goodness.

For example, consider a database set $D=\{D_1, D_2, D_3, D_4\}$, as shown in TABLE II.

TABLE II.
DATABASE SET

database	$FI(D)^{0.5}$
D_1:{a, b, c, (a,c), (b, d), (a,b,c), (a,c,d), (a,b,c,d)}	{a:0.625, b:0.500, c:0.625, (a,c):0.500}
D_2:{b, c, (b,c), (d,c)}	{b:0.500, c:0.750}
D_3:{b, e, g, (e,g), (a,e,g), (b,e,g)}	{e:0.666, g:0.666, (e,g):0.500}
D_4:{d, e, g, (b,c), (c,e), (d,e,g), (c,e,g), (c,d,e)}	{c: 0.500, e:0.625}

We can obtain the distance matrix based on Definition 2.

$$\begin{array}{c} \\ D_1 \\ D_2 \\ D_3 \\ D_4 \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & D_2 & D_3 & D_4 \end{array} \\ \begin{bmatrix} 0.000 & 0.526 & 1.000 & 0.826 \\ 0.526 & 0.000 & 1.000 & 0.733 \\ 1.000 & 1.000 & 0.000 & 0.731 \\ 0.826 & 0.733 & 0.731 & 0.000 \end{bmatrix} \end{array}$$

According to Definition 4, the *m-cluster* of D will be

$cluster^4(D)=\{\{D_1\}, \{D_2\}, \{D_3\}, \{D_4\}\}$

$cluster^3(D)=\{\{D_1\}, \{D_3\}, \{D_2,D_4\}\}$

$cluster^2(D)=\{\{D_1,D_2,D_4\}, \{D_3\}\}$

$cluster^1(D)=\{\{D_1,D_2,D_3,D_4\}\}$

The goodness of each clustering is

$$goodness(cluster^4(D))=\frac{2}{4\times 3}\cdot(0.526+1+0.826+1+0.733+0.731)-0-\frac{4}{4}$$
$$=-0.197$$

$$goodness(cluster^3(D))=\frac{2}{3\times 2}(\frac{1}{1}+\frac{0.526+0.826}{2}+\frac{1+0.731}{2})-\frac{1}{3}(\frac{0.733}{1})-\frac{3}{4}$$
$$=-0.147$$

$$goodness(cluster^2(D))=0.063,\quad goodness(cluster^1(D))=-1.053$$

As we have seen, $cluster^2(D)=\{\{D_1,D_2,D_4\}, \{D_3\}\}$ is the best clustering because it has the highest goodness value.

B. K-mean Clustering Model

We define the mean distance between a database and a class as follows.

Definition 8: Let $D=\{D_1, D_2, ..., D_n\}$ be a set of transaction databases, and $cluster^m(D)=\{class_1(D), class_2(D), ..., class_m(D)\}$ be an *m-cluster* of D. The definition of the mean distance between D_k and $class_i$ is

$$dis(D_k, class_i)=\frac{1}{|class_i|}\cdot\sum_{D_q\in class_i} dis(D_k, D_q)$$

We construct a clustering model based on the k-mean as follows:

We classify the databases into m classes. For any database in one class, if the average distance between the database and any other class is the minimum, then we redistribute the database into the class, until no database remains to be redistributed. In our model, we can obtain the best *m-cluster* by reallocating each database to the nearest class.

Given a database set $D=\{D_1, D_2,..., D_n\}$, we can obtain the best clustering of D based on the goodness of all different *m-clusters*.

Definition 9: Let $D=\{D_1, D_2, ..., D_n\}$ be a set of transaction databases. The best clustering of D is defined as follows:

$$endCluster(D)=\arg[\text{mininum}\{goodness(bestCluster^m(D))]$$

where $bestClass^m(D)$ denotes the best *m-cluster* in D, $m=1,2,...,n$.

In Section C, the process and the algorithms of clustering transaction databases are described.

C. Clustering Algorithm Based on the K-Mean

A two-step approach is proposed in this section to identify the best clustering from the given transaction databases. In the first step, a procedure to generate the best *m-cluster* is designed. In the second step, an algorithm to search for the best clustering is developed.

Given the set of databases $D_1, D_2, ... D_n$, suppose the number of classes is m, the best *m-cluster* can be obtained by redistributing each database into the nearest class. The procedure *meanCluster* for generating the best *m-cluster* is designed as Procedure 1.

Procedure1 *meanCluster*

Input: $D_i(1\le i\le n)$: databases, m: number of classes;
Output: $bestCluseter^m$: best *m-cluster* of databases;

```
begin
(1) construct the distance matrix DIS[n][n] by computing
    the distance between different databases, Di and Dj;
    let clusterList[m]←Ø;
    for i:=0 to n do{
        create random integer k(k<=m);
        add i to clusterList[k];
    }
(2) while flag:=true and count<100 do{
        let flag←false;
        for i:=0 to n do{
            for k=0 to m do{
                if(clusterList[k] contains i and size of
                   clusterList[k] >1)
                {
                    find mininum distance between Di and
                    clusterList[t] in all clusterList;
                    if(t!=k){
                        remove i from clusterList[k];
                        add i to clusterList[t];
                        let flag←true;
                    }
                    break;
                }
            }
        }
        let count←count+1;
    }
(3) let bestCluster^m←{clusterList[0],cluserList[1],...,
    clusterList[m-1]};
    return bestCluster^m;
end
```

Step (1) initializes clustering, and the time complexity is $O(mn^2/8-n/8+mn)$. Step (2) updates the clustering result based on the k-mean. Step (3) returns the best *m-cluster*. The time complexity of the algorithm is $O(mn^2)$.

We design an algorithm to search for the best clustering, as shown below.

Procedure2 *bestCluster*
Input: $D_i(1\le i\le n)$: databases;
Output: *endCluseter*: best clustering of databases;
begin
construct the array goodness[n+1] to store different goodness of different best m-cluster;
for m:=0 to n **do{**
compute $bestCluster^m$ by Procedure 1;
let goodness[m]←goodness($bestCluster^m$);
}
Search maximum goodness[k] in goodness[n];
let $endCluseter=bestCluster^k$;
return *endCluseter*;
end

The time complexity of computing for the best *m-cluster* in Procedure 2 is $O(mn^2)$.

Section IV details the experiments conducted to prove the effectiveness of our algorithms.

IV. Experiment

We carried out several experiments to demonstrate the efficiency of the approach. One is to investigate the result of clustering transaction databases with our method, using the strategy of Zhang and Animesh Adhikari. The other experiment is to examine the time complexity of these algorithms. The experiments are implemented on a 1.6GHz Pentium processor with 2GB of memory. The Java Edition 6 platform is used as the development tool.

A. Dataset Preparation

We use the same datasets as those in [6], namely, synthetic two datasets T10I4D100K and T40I10D100K, which are derived from the IBM Almaden Quest research group. The attribute tables of the two datasets are shown in TABLE III.

TABLE III.
Attribute Table of Dataset

Dataset	NT	ALT	NI	AFI
T10I4D100K	1,00,000	11.102280	870	1276.124138
T40I10D100K	1,00,000	40.605070	942	4310.516985

NT is the number of transactions. ALT denotes the average length of a transaction. NI and AFI are the number of items and the average frequency of an item, respectively.

B. The Results of the Clustering Transaction Databases

First, we divided T10I4D100K into 10 different sub-datasets as our transaction database. We then mined the frequent items from these databases, with different threshold α separately. Finally, the items were clustered according to the three methods. The results are shown in Figure 1.

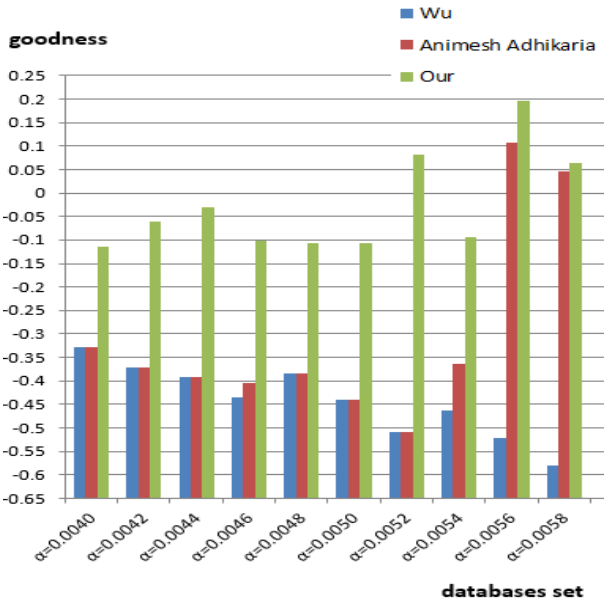

Figure 1. Goodness of Best Clustering by different Methods

In the figure, the horizontal axis denotes different threshold α. The vertical axis is the goodness of the best clustering. The column stands for the results of the clustering by the three methods.

The figure also shows that our method can obtain better results than those with using the strategies of Wu and Animesh Adhikari because the value of our goodness is largest under the same threshold α. The result of the study by Adhikari is better than that of Wu in some cases, such as α=0.0054. In our experiment, we found that the result obtained by the method by Wu can achieve complete classification with zero coupling.

C. The Result of Time Complexity

In this experiment, we divided the T40I4D100K into different sub-datasets as our transaction database, and then mined the frequent items from these databases under the given threshold α=0.054. At varying numbers of transaction databases, the time consumption of the different algorithms is obtained, as shown in Figure 2.

In the figure, the horizontal axis denotes the number of transaction databases. The vertical axis is the time consumption, and the curve represents the results of time consumption variation along with the number of databases.

Figure 2 also shows that the time consumption is amplified with the increasing number of databases. In terms of time consumption, when the number of databases increases, the results of our method become

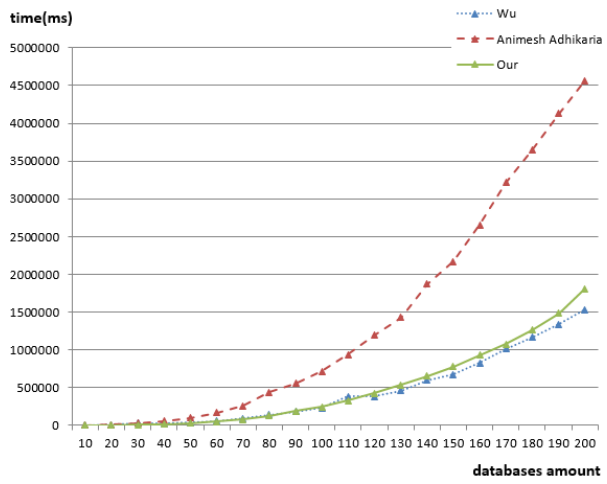

Figure 2. Time Consumption of Different Methods

similar to those obtained by the method of Wu, whereas those by the method of Adhikari rapidly increase.

Obviously, the method by Wu and our method are more appropriate for large multi-databases (i.e., when a multi-database contains many classes) than the strategy proposed by Adhikari.

Consequently, the clustering result of our method is better than that of the others, and the time complexity achieved by our method is satisfactory.

V. Conclusion

Clustering transaction databases is a valuable topic in the area of multi-database mining. Based on previous studies, we proposed a k-mean clustering method. We can obtain ideal classification results according to the mean distance between a database and a class. The experiments demonstrated that our method is more effective and efficient than other present clustering algorithm for transaction databases.

Acknowledgment

This work was supported in part by a grant from the BaGui scholar team project“Multi-source data mining and information security” , partially supported by the Guangxi innovation team project under grant GA060004, universities science and technology key research projects.

The authors would like to thank the anonymous reviewers for their constructive comments, and also thank the editors for their checking on the final version of this paper.

References

[1] X. Wu, S. Zhang. Synthesizing high-frequency rules from different data sources. *Knowledge and Data Engineering, IEEE Transactions on*,15(2), pp.353-367, 2003.

[2] H. Liu, H. Lu, J. Yao. Identifying relevant databases for multidatabase mining. *Research and Development in Knowledge Discovery and Data Mining*, pp.210–221, 1998.

[3] S. Zhang,X. Wu,C. Zhang.Multi database mining[J]. *IEEE Computational Intelligence Bulletin*, 2(1), pp.5213, 2003.

[4] Duda, R.O., Hart, P.E. Pattern Classification and Scene Analysis. John Wiley & Sons (1973).

[5] X. Wu, C. Zhang, S. Zhang, Database classification for multidatabase mining, *Information Systems*,30(1), pp.71–88,2005.

[6] Animesh Adhikaria, P.R. Rao. Efficient clustering of databases induced by local patterns. *Decision Support Systems*, 44, pp.925 – 943, 2008.

[7] D. Yuan, H. Fu, Z. Li, H. Wu. An application-independent database classification method based on high cohesion and low coupling. *Journal of Information & Computational Science* ,7(1), pp.1-6,2012.

[8] Frequent itemset mining dataset repository, http://fimi.cs.h-elsinki. fi/data.

[9] H. Jiang, J. Gu, Y. Liu, et al. Study of Clustering Algorithm based on Fuzzy C-Means and Immunological Partheno Genetic[J]. Journal of Software, 8(1), pp.134-141, 2013.

[10] X. Li. A New Text Clustering Algorithm Based on Improved K_means[J]. Journal of Software, 7(1), pp.95-101, 2012.

[11] Q. Niu, X. Huang. An improved fuzzy C-means clustering algorithm based on PSO[J]. Journal of Software, 6(5), pp.873-879, 2011.

Dingrong Yuan is a Professor in the College of Computer Science and Information Technology, Guangxi Normal University, Guilin, China. He received his Ph.D. in Computer Science from the Beijing University of Technology, Beijing, China. His recent research interests include database classification, data analysis, and data mining.

Yuwei Cuan is a master’s degree student in the College of Computer Science and Information Technology, Guangxi Normal University, Guilin, China. His recent research interests include database classification, data analysis, and data mining.

Yaqiong Liu is a master’s degree students in the College of Computer Science and Information Technology, Guangxi Normal University, Guilin, China. Her recent research interests include database classification, data analysis, and data mining.