Identifying Global Exceptional Patterns in Multi-database Mining

Chengqi Zhang[1], Meiling Liu[2], Wenlong Nie[3], and Shichao Zhang[1,2]

***Abstract*— In multi-database mining, there can be many local patterns (frequent itemsets or association rules) in each database. At the end of multi-database mining, it is necessary to analyze these local patterns to gain global patterns, when putting all the data from the databases into a single dataset can destroy important information that reflect the distribution of global patterns. This paper develops an algorithm for synthesizing local patterns in multi-database is proposed. This approach is particularly fit to find potentially useful exceptions. The proposed method has been evaluated experimentally. The experimental results have shown that this method is efficient and appropriate to identifying exceptional patterns.**

***Index Terms*— multi-database mining; local pattern evaluation; local pattern; global pattern; exceptional pattern**

I. Introduction

With the increasing development and application of distributed database technique and computer network, there exist many distributed databases in a business or financial organization. For example, a large company has many subsidiary companies, and each subsidiary company has its own database, all of the databases from each subsidiary company are relevant or irrelevant in logic, but they are distributed in different places. Different subsidiary company has different functions in helping the head company to make decisions. To make decisions for the development of company, the decision maker of the head company needs to know every database's interesting pattern or regulation and then synthetically evaluate these local patterns to generate global patterns.

It would appear to be unrealistic to collect data from different branches for centralized processing because of the potentially volume of data [20]. For example, different branches of Wal-Mart collect 20 million transactions per day. This is more than the rate at which data can feasibly be collected and analyzed by using today's computing power.

On the other hand, because of data privacy and related issues, it is possible that some databases of an organization can share their association rules but not their original data. Therefore, mining association rules from different databases and forwarding the rules (rather than the original raw data) to the central company headquarter provides a feasible way dealing with multiple database problems [19].

However, current data mining researches focus on mining in mono-database, but mono-database mining is different from multi-database mining because of their different data structure. So we need to come up with other solutions to analyze the data in multi-databases instead of using the technique in mono-database mining. This paper mainly discusses the pattern evaluation process at the end of data mining process and presents a method for identifying exceptional patterns.

[1]Faculty of Information Technology, University of Technology Sydney, PO Box 123, Broadway NSW 2007, Australia. {chengqi, zhangsc}@it.uts.edu.au

[2]Department of Computer Science, Guangxi Normal University, Guilin, 541004, P. C. China. hsp01wl@zsu.edu.cn

[3]The Institute of Logic and Cognition, Zhongshan University, Guangzhou, 510275, P. C. China. hsp01wl@zsu.edu.cn

This paper is organized as follows. Section II describes the process of multi-database mining and the patterns that exist in multi-database. Section III proposes a model for identifying exceptional patterns. Section IV designs an algorithm for identifying global exceptional patterns. In Section V, several experiments have be conducted for evaluating the proposed approach. In the last section we conclude this paper.

II. Description of Multi-Database Mining Problem

For description, this section states multi-database mining problem in a simple way.

Multi-database mining is the process of analyzing the data in multi-databases, and finding useful and novel knowledge, which is highly supported by most of databases or individual databases. Different from mono-database mining, there maybe exist semantic conflicts in multi-databases. The conflicts consist of synonym and homonyms. Synonym means that different field names in different databases denote the same data object. The head company must observe the semantic equivalence of the fields and translate the different local fields into a single global field name. Another kind of conflict is homonym, which means different data objects have the same name in different databases. The head company must recognize the semantic difference between the same field names and translate the same name into different field names. Because of these conflicts in different databases, the preprocessing in multi-database mining is very important. If we combine all the data in different databases into a single one and mine the large single database, then it may hide some features in a database and lose some useful pattern, moreover the techniques for integrating multi-databases is not perfect and it will take a large amount of efforts to integrate all the databases. The huge dataset after integrating will be difficult to deal with and its data may not be stored into memory at a time. So we cannot use traditional multi-database mining technique to analyze the data in multi databases.

The existing techniques for dealing with multi-databases first will classify all the databases into several group, databases in each group are relevant [23]. The classification of database is necessary, if not, the mined patterns may not be understood because there are some irrelevant data. For example, a large chain store has 10 subsidiary stores, some of them mainly sell groceries, and others sell electrical appliances. When mining these databases, one should classify the transaction databases in order to find out the databases that are relevant in the product categories. If we integrate all the transaction databases into a single one, at last perhaps we will not find rules because integrating all the databases involves in some irrelevant information. For example, the data in the food transaction database and the electrical appliances transaction database are put together, and then the association between the two kinds of product is difficult to understood by user because these product are not sold together and are distributed in different places. So it is very necessary to classify all the databases before mining data in multi-databases. After classifying, we can apply the techniques for mining mono-database to multi-database, and then find all the local patterns in every database. At last, all the local patterns will be analyzed and evaluated in order to find out the valuable information.

Patterns in multi-database can be divided into 4 categories [20], [21]:

(1) Local patterns. In a large interstate company, its branches has its own databases, it is impossible for the head company to analyze all its branches' database, so the branches need to analyze its data in its own local database and submit the mined patterns to head company. The mined patterns from each branch is called local patterns. Their function is to provide local databases' data features to the head company to make decision.
(2) High-voting patterns [21]. This kind of patterns is supported by most subsidiary companies. They reflect the common features among subsidiary companies. According to these patterns, the head company can make decisions for the common profits of branches.
(3) Exceptional patterns. These patterns are highly supported by only a few branches, that is to say, these patterns have very high support in these branches and zero support in other branches. They reflect the individuality of branches. And according to these patterns, the head company can adjust measures to local conditions and make special policies for these branches.
(4) Suggesting patterns. These patterns have fewer votes than the minimal vote but are very close to minimal vote. The minimal vote is given by users or experts. If a local pattern has votes equal to or greater than minimal vote, the local pattern is said to be a global pattern, called as high-voting pattern. If a local pattern has votes less than the minimal votes but are very close to the minimal vote, it is called suggesting patterns and sometimes it is useful for decision making.

The definitions of these patterns in multi-database indicate that there are differences between multi-database and mono-database mining. The final purpose of multi-database is to analyze and evaluate the common or special features in all the databases. In this paper, we only describe the exceptional patterns.

A. Related Work

Data mining techniques (see [1], [16], [22]) have been successfully used in many diverse applications. These include medical diagnosis and risk prediction, credit-card fraud detection, computer security break-in and misuse detection, computer user identity verification, aluminum and steel smelting control, pollution control in power plants and fraudulent income tax return detection. Developed techniques are oriented towards mono-databases.

Multi-database mining has been recently recognized as an important research topic in the KDD community. One article [24] proposed a means of searching for interesting knowledge in multiple databases according to a user query. The process involves selecting all interesting information from many databases by retrieval. Mining only works on the selected data.

Liu, Lu and Yao [10] proposed another mining technique in which relevant databases are identified. Their work has focused on the first step in multi-database mining, which is the identification of databases that are most relevant to an application. A relevance measure was proposed to identify relevant databases for mining with an objective to find patterns or regularity within certain attributes. This can overcome the drawbacks that are the result of forcedly joining all databases into a single very large database upon which existing data mining techniques or tools are applied. However, this database classification is typically database-dependent. Therefore, Zhang and Zhang have proposed a database-independent database classification in [23], which is useful for general-purpose multi-database mining.

Zhong et al [25] proposed a method of mining peculiarity rules from multiple statistical and transaction databases based on previous work. A peculiarity rule is discovered from peculiar data by searching the relevance among the peculiar data. Roughly speaking, data is peculiar if it represents a peculiar case described by a relatively small number of objects and is very different from other objects in a data set. Although it appears to be similar to the exception rule from the viewpoint of describing a relatively small number of objects, the peculiarity rule represents the well-known fact with common sense, which is a feature of the general rule.

Other related research projects are now briefly described. Wu and Zhang advocated an approach for identifying patterns in multi-database by weighting [19]. Ribeiro et al. [14] described a way of extending the INLEN system for multi-database mining by incorporating primary and foreign keys as well as developing and processing knowledge segments. Wrobel [18] extended the concept of foreign keys to include foreign links since multi-database mining also involves accessing non-key attributes. Aronis et al. [4] introduced a system called WoRLD that uses spreading activation to enable inductive learning from multiple tables in multiple databases spread across the network. Existing parallel mining techniques can also be used to deal with multi-databases [2], [6], [7], [12], [13], [15].

The above efforts provide a good insight into multi-database mining. However, there are still some limitations in traditional multi-database mining that are discussed in next subsection.

B. Limitations of Previous Multi-database Mining

As have seen, traditional multi-database mining is fascinated with mono-database mining techniques. It consists of a two-step approach. The first step is to select the databases most relevant to an application. All the data is then pooled together from these databases to amass a huge dataset for discovery upon mono-database mining techniques that can be used. However, there are still some limitations discussed below.

1) Putting all the data from relevant databases into a single database can destroy some important information that reflect the distributions of patterns. The statement "85% of the branches within a company agree that a customer usually purchases sugar if he/she purchases coffee" is an example of such a piece of information. These patterns may be more important than the patterns present in the mono-database in terms of global decision-making within a company. Hence, existing techniques for multi-databases mining are inadequate for applications.

In some contexts, each branch of an interstate company, large or small, has equal power in voting patterns for global decisions. For global applications, it is natural for the company headquarters to be interested in the patterns voted for by most of his/her branches. It is therefore inadequate in multi-database mining to utilize existing techniques for mono-databases mining.

2) Collecting all data from multi-databases can amass a huge database for centralized processing using parallel mining techniques.

It may be an unrealistic proposition to collect data from different branches for centralized processing because of the huge data volume. For example, different branches of Wal-Mart receive 20 million transactions a day. This is more than the rate at which data can be feasibly collected and analyzed using today's computing power.

3) Because of data privacy and related issues, it is possible that some databases of an organization may share their patterns but not their original databases.

Privacy is a very sensitive issue, and safeguarding its protection in a multi-database is of extreme importance. Most multi-database designers take privacy very seriously, and allow some protection facility. For source sharing in real-world applications, sharing patterns is a feasible way of achieving this.

From the above observations, it is clear that traditional multi-database mining is inadequate to serve two-level applications of an interstate company. This prompts the need to develop new techniques for multi-database mining.

Based on the above analysis, the problem for our research can be formulated as follows.

> Let $D_1, D_2, \cdots, D_m$ be m databases in the m branches $B_1, B_2, \cdots, B_m$ of a company, respectively; and LI_i be the set of local patterns (local instances) from D_i ($i = 1, 2, \cdots, m$). We are interested in the development of new techniques for identifying global exceptional patterns of interest in the local patterns.

III. Identifying Exceptional Patterns of Interest

Given n databases $D_1, D_2, \cdots, D_n$, they represent the databases from n branches of a large company. Let $LP_1, LP_2, \cdots, LP_n$ be the corresponding local patterns which are mined from every database; And $minsupp_i$ be the user specified minimal support in the database D_i $(i = 1, 2, \cdots, n)$. For each pattern P, its support in D_i is denoted by $Supp_i(P)$. We define the average vote of local patterns in the databases as follows.

Formula 1: $$AverVotes = \frac{\sum_{i=1}^{Num(GP)} Num(P_i)}{Num(GP)}$$

Where GP means the Global Patterns, it is the set of all patterns from each database, that is $GP = \{LP_1 \cup LP_2 \cup \cdots \cup LP_n\}$, and $Num(GP)$ is the number of patterns in GP. We regard the $AverVotes$ as a boundary to identify exceptional patterns and high-voting patterns. If a pattern's votes is less than the $AverVotes$, then it will be considered as an candidate exceptional pattern, otherwise as an high-voting pattern. We use CEP to denote the set of Candidate Exceptional Patterns and define the the global support of a pattern as follows.

Formula 2: $$Supp_G(P) = \frac{\sum_{i=1}^{Num(P)} \frac{Supp_i(P) - minsup_i}{1 - minsup_i}}{Num(P)}$$

where, $Supp_G(P)$ means the global support of a pattern; $Num(P)$ is the number of databases which support the pattern P. In this formula, we assume that the n databases play the same role in helping the head company to make decisions, that is to say that they have the same authority in providing their patterns to the head company. So we don't consider the weight of every database. Because

$$\frac{Supp_i(P) - minsup_i}{1 - minsup_i} \leq 1$$

therefore,

$$\sum_{i=1}^{Num(P)} \frac{Supp_i(P) - minsup_i}{1 - minsup_i} \leq Num(P)$$

The value $Supp_G(P)$ will be equal to or less than 1, as a result the closer $Supp_G(P)$ is to 1, the more significant the pattern will be.

The formula gives a method to compute a pattern's significance value. It uses the distance between a pattern's support and the corresponding database's minimal support as a measure. Because different database have different data information, We cannot simply say that 0.5 is greater than 0.22 in two databases whose minimal support is 0.48 and 0.13 respectively; This is because the two databases' minimal supports are different. According to the formula, we can obtain the significance of a pattern P in D_i. The greater the value $\frac{Supp_i(P) - minsup_i}{1 - minsup_i}$ is, the more significant the pattern P will be in D_i.

We only need to calculate the $Supp_G(P)$ values of patterns in CEP because these patterns' votes are less than the $AverVotes$, they will possibly be exceptional patterns. If a pattern has very high support in few databases and zero support in other databases, then its global support will be high. This pattern is referred to an exceptional pattern defined in Section 2. To evaluate the highness of the support of a pattern P in a database, we define a metrics as follows.

Formula 3: $$S(P) = \frac{Supp_i(P) - minsup_i}{1 - minsup_i}$$

where, $S(P)$ is the highness of the support of P in the database D_i, $Supp_i(P)$ is the support of P in D_i, $minsup_i$ is the user-specified minimum support for mining D_i.

This formula means, the higher the support of a pattern in a subsidiary company, the more interesting the pattern will be. We define the formula to compute the deviation of a pattern from the corresponding minimal support $minsup_i$, The value will be used to draw plots to show how far the patterns deviate from the same level.

IV. Algorithm Design

Exceptional patterns reflect the individuality of branches within an interstate company. This section presents an algorithm, *Identify-ExPattern*, for identifying exceptional patterns.

Algorithm 1: IdentifyExPattern

Input: LP_i: set of local patterns; $minsup_i$: minimal support threshold in D_i $(i = 1, 2, \cdots, n)$;

Output:EP: the set of exceptional patterns;

begin

(1) $GP \leftarrow \{LP_1 \cup LP_2 \cup \cdots \cup LP_n\}$; CEP=Ø;

(2) For each pattern P in GP do
Count P's votes, $Num(P)$; And Record which database support it, using $from$ to note them.
Calculate the average votes using Formula 1: $AverVotes = \frac{\sum_{i=1}^{Num(GP)} Num(P_i)}{Num(GP)}$

(3) For each pattern P in GP do
if $(Num(P) < AverVotes)$ $CEP = CEP \cup P$

(4) For each candidate exceptional pattern P in CEP do
$$Supp_G(P) \leftarrow \frac{\sum_{i=1}^{Num(P)} \frac{Supp_i(P) - minsup_i}{1 - minsup_i}}{Num(P)}$$

(5) Rank all the patterns P in CEP by their $Supp_G(P)$;

(6) Output the high rank patterns in CEP and the databases which support them;

End.

The algorithm $IdentifyExPattern$ is to search all the significant exceptional patterns from the given n local patterns.

Step (1) generates the set of patterns from each database. Step (2) counts each pattern's votes, and the average votes of patterns $AverVotes$. Step (3) generates the candidate exceptional patterns. Step (4) is to calculate all the candidate exceptional patterns's $Supp_G(P)$ values. Step (5) ranks the candidate exceptional patterns by their $Supp_G(P)$. Step (6) outputs all the exceptional patterns which satisfy the user's requirement and have high rank.

Example 1: Consider 5 databases $D_1, D_2, \cdots, D_5$, their corresponding patterns is in the following. Patterns are denoted by A-F, the value after each colon is the pattern's support; $minsup_1 = 0.49$, $minsup_2 = 0.48$, $minsupp_3 = 0.82$, $minsupp_4 = 0.20$, $minsupp_5 = 0.13$ are 5 databases' minimal support respectively.

$$\begin{aligned}
LP_1 &= \{\{A:0.69\}; \{C:0.68\}; \{F:0.52\}\} \\
LP_2 &= \{\{A:0.50\}; \{B:0.62\}; \{C:0.91\}; \{E:0.82\}; \\
&\quad \{F:0.76\}; \{G:0.86\}\} \\
LP_3 &= \{\{A:0.87\}; \{C:0.85\}; \{D:0.86\}; \{E:0.86\}; \\
&\quad \{F:0.95\}\} \\
LP_4 &= \{\{B:0.36\}; \{C:0.31\}; \{E:0.28\}\} \\
LP_5 &= \{\{E:0.22\}\}
\end{aligned}$$

We now use the algorithm $IdentifyExPattern$ to search all the exceptional patterns from the given local patterns. According to the Step (1) and Step (2), we can get $GP = \{A, B, C, D, E, F, G\}$,and the $AverVotes = \frac{18}{7} = 2.57$. Because Pattern B, D and G have less votes than the $AverVotes$. After pruning by $AverVotes$, $CEP = \{B, D, G\}$. The $Supp_G(P)$ value of each pattern in CEP are shown as follows.

$Supp_G(B) = 0.235$, Pattern B comes from $\{D_2, D_4\}$
$Supp_G(D) = 0.222$, Pattern D comes from $\{D_3\}$
$Supp_G(G) = 0.73$, Patterns G comes from $\{D_2\}$

After rank the patterns in CEP by their $Supp_G(P)$, the order will be $\{G, B, D\}$. It is obvious that pattern {G} has the highest global

support and it is supported by only a database. So it can be regarded as an exceptional pattern. After finding such exceptional patterns, the head company can use the patterns to assist making special decision for the corresponding subsidiary company.

From the example, we can see that this approach is reasonable and when the manager of head company makes decisions for the development of his company, he can not consider only the number that supported a certain pattern but also the pattern's support value in these databases.

In the practical application of multiple database, such as chain stores and interstate company , because it maybe generate large amount of patterns, it is necessary to find an approach to evaluate all the patterns.

V. Experiments

In this section, we evaluate the function of the approach. The following experiments were conducted on Pentium 4 personal computer with 256 MB main memory running Microsoft Windows 2000. Our intention is not to evaluate the running time of our approach, So the experiment environment is not important. The dataset used in one experiment were generated randomly, we considered the data as mined patterns. And the other experiment was conducted on real dataset downloaded from the Internet (http://www.ics.uci.edu/ mlearn/MLSummary.html).

A. Random Patterns

First, we present our experiment on the randomly generated patterns. These patterns were generated randomly by our patterns generator and were assigned certain support. In addition, the minimal support of each database was also assigned randomly. Of course, when designing the algorithm for generating patterns' support, we assigned that each pattern's support must equal to or greater than the corresponding database's minimal support because we considered these patterns were pruned by minimal support threshold. Table 1 shows the parameter setting in Experiment 1. And Table 2 shows the number of patterns in each database and the minimal support of each database.

Table 1: Parameters setting in Experiments

Number of datasets	10
Average number of patterns in all datasets	10
Patterns Symbols	1-15

Table 2: Number of patterns and the minimal support in each dataset

Dataset	Number of patterns	Patterns	minsupp
D0	3	{11}:0.57 {5}:0.63 {4}:0.21	0.19
D1	9	{6}:0.87 {13}:0.77 {12}:0.80 {3}:0.75 {7}:0.78 {10}:0.82 {4}:0.75 {8}:0.88 {5}:0.82	0.74
D2	5	{7}:0.46 {4}:0.47 {15}:0.49 {2}:0.54 {14}:0.51	0.45
D3	11	{5}:0.80 {7}:0.85 {14}:0.81 {6}:0.87 {13}:0.81 {2}:0.84 {3}:0.81 {1}:0.88 {10}:0.81 {9}:0.83 {12}:0.81	0.80
D4	2	{10}:0.50 {14}:0.50	0.05
D5	2	{13}:0.22 {12}:0.40	0.10
D6	5	{4}:0.89 {1}:0.88 {12}:0.88 {7}:0.88 {6}:0.89	0.88
D7	10	{10}:0.39 {4}:0.52 {13}:0.71 {1}:0.88 {7}:0.27 {3}:0.38 {5}:0.86 {8}:0.81 {11}:0.74 {12}:0.74	0.22
D8	3	{3}:0.74 {4}:0.85 {15}:0.86	0.54
D9	2	{4}:0.61 {2}:0.49	0.38

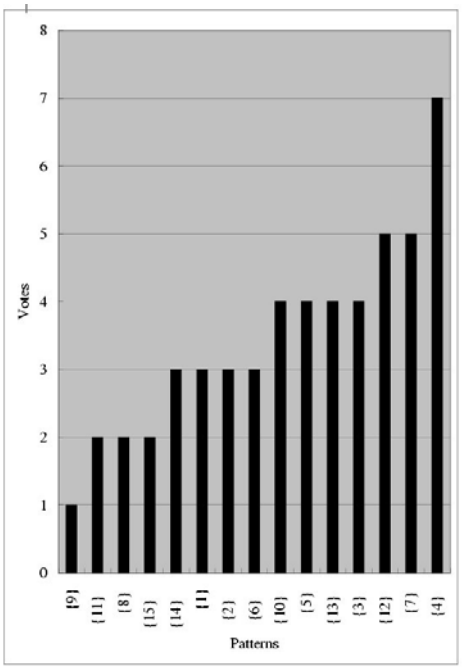

Fig. 1. Votes of patterns

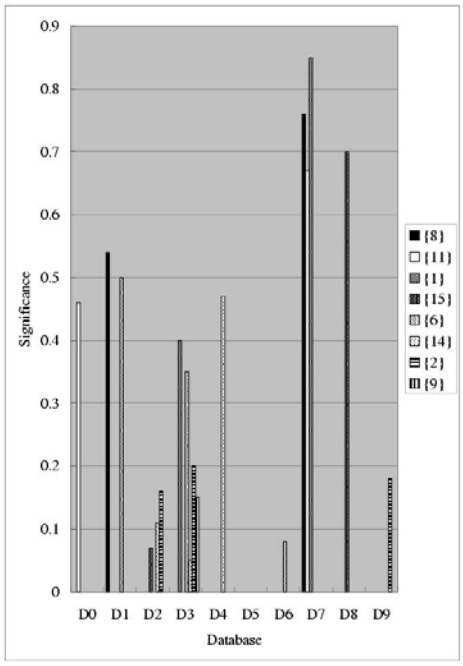

Fig. 2. Each pattern's significance calculated by formula 3.

The distributions of patterns in GP are shown in Figure 1. X-coordinate denotes the patterns in GP, and Y-coordinate are the patterns' votes. In this experiment, $averVotes = 3.3$, from Figure 1, we can see that pattern $\{\{9\}, \{11\}, \{8\}, \{15\}, \{14\}, \{1\}, \{2\}, \{6\}\}$ are candidate exceptional patterns because their votes are less than 3.3.

After executing the $IdentifyExPattern$ algorithm, we can get the global support of all candidate exceptional patterns. The $Supp_G(P)$ values are shown in Table 3.

Table 3 shows that the candidate exceptional pattern $\{8\}$ has the highest global support and it is supported by $D1$ and $D7$. When searching Table 2, we find that pattern $\{8\}$ has the highest support in $D1$ and the second highest support in $D7$ comparing to their corresponding minimal support. The experiment results show that this method can be used to find exceptional patterns when there exist exceptional patterns in Multiple databases. In section 5.2, we will present an experiment in which we can not find any exceptional patterns because there doesn't exist any exceptional patterns in the specified multi-database.

B. Real Datasets

For real-life applications, we have also evaluated our approach using the database downloaded from the Internet (please see http://www.ics.uci.edu/ mlearn/MLSummary.html/). We choose the Zoo Database containing 101 instances and 18 attributes (animal name, 15 boolean attributes, 2 numerics). The boolean attributes are "hair", "feathers", "eggs", "milk", "airborne", "aquatic", "predator", "toothed", "backbone", "breathes", "venomous", "fins", "tail", "domestic" and "catsize". And the numeric attributes are "legs" and "type", where the "type" attribute appears to be the class attribute. All the instances are classified into 7 classes.

To obtain multiple and relevant databases, we vertically partitioned the Zoo Database into 7 subset datasets according to the "type" attribute. Each dataset contained 18 attributes. When preprocessing, we used different number to denote different attribute values. After preprocessing, we mined the 7 datasets respectively and obtained their own frequent itemsets. Table 4 shows the 7 datasets' corresponding information.

Because of the large amount of frequent itemsets, to better illustrate the efficiency of our approach, we only selected some special frequent itemsets which were relevant to the specified attribute. We selected 97 frequent itemsets and their votes are shown in Figure 3. In this experiment, the $AverVotes = 2.4$.

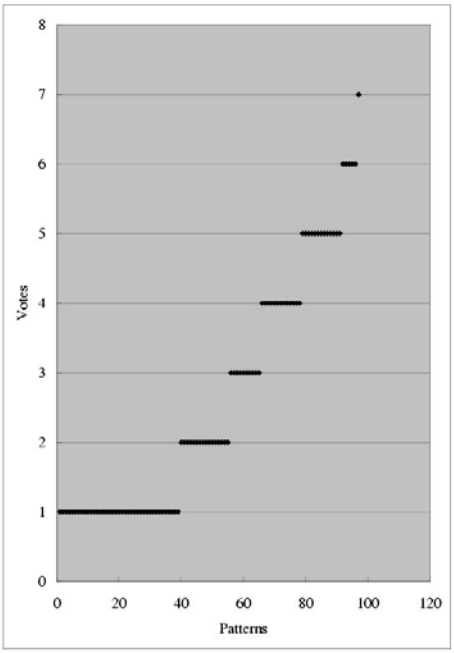

Fig. 3. Votes of 97 frequent itemsets

From Figure 3, we can see that there are about 55 frequent itemsets whose votes are less than the $AverVotes$. Table 5 shows the mined typical exceptional patterns using our approach.

From Table 5, we can easily see that the animals in D1 are characteristic with Pattern P1 and those animals in other datasets have no the character. So it can be regarded as an exceptional pattern owned by D1. This is because the animals in D1 are mammals which are different from other datasets. And D4's instances are all fish, only fish have fins, so the results is reasonable. For other patterns showed in Table 5, they are also considered as exceptional patterns. In this experiment, we partitioned the original database into 7 datasets by their "type" attribute. This partition makes that each database belongs to a certain class, So we can find the potential exceptional patterns. From the experiment, we can draw a conclusion that our approach is useful to identify exceptional patterns.

At last, we simply presented another experiment. In this experiment, we only selected 3 datasets (D4, D5, D6, D7) and 3 attributes in the Zoo Database ("feathers", "eggs", "milk"). The experiment result shows that most of the animals in the 4 datasets have the common features: most of them have no feathers, and can lay eggs but have no milk. That is to say, there doesn't exist potential exceptional patterns. As a result, we can not find any exceptional patterns.

Table 3: Exceptional patterns analysis

Pattern	Patterns	Supported by which dataset	Patterns' Meaning
P1	{hair=1}	D1	The animals in D1 usually have hair
P2	{eggs=0}	D1	The animals in D1 usually can not lay eggs
P3	{milk=1}	D1	The animals in D1 usually have milk
P4	{legs=4}	D1	The animals in D1 usually have 4 legs
P5	{feathers=1}	D2	The animals in D2 usually have feathers
P6	{legs=2}	D2	The animals in D2 usually have 2 legs
P7	{fins=1}	D4	The animals in D4 usually have fins
P8	{legs=0}	D4	The animals in D4 usually have no legs
P9	{hair=0 and legs=4}	D5	The animals in D5 usually have 4 legs, but no hair. These characters are different from those in D1, in D1, the animals also have 4 legs but they have hair.
P10	{predator=0}	D6	The animals in D6 are not predators.
P11	{legs=6}	D6	The animals in D6 usually have 6 legs.
P12	{hair=0 and backbone=0}	D7	The animals in D7 usually have no hair and no backbones.

VI. Summary

In this paper, we studied an approach for identifying exceptional patterns from multiple databases. It can be considered as an postprocessing work after mining multiple, relevant databases. We conducted several experimental studies, one was experimented on patterns which were generated randomly and the other was experimented on real Zoo Database. We found that our approach can identify potential exceptional patterns from multiple databases' patterns. On one hand, if there exists potential exceptional patterns in multiple databases, the approach can be used to find them out. On the other hand, if there does not exist any potential exceptional patterns in multiple databases, no exceptional patterns can be found. Therefore, the approach is fit to find potential exceptional patterns. It seems that the datasets used in the experiments are not relevant to the business data, but our intention is to illustrate the function of our approach. In the practical application, when faced with the patterns of multiple databases, we can use the method to find exceptional patterns from the multiple databases and make special decisions.

However, if more information about the multiple databases can be considered, the experiment results will be more perfect. There are one direction for ongoing work by weighting. If each subsidiary company plays different roles in assisting making decision for the head company. We can assign weights for each database.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments on the first version of this paper.

References

[1] R. Agrawal, T. Imielinski, and A. Swami, Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, 6(1993): 914-925.

[2] R. Agrawal, J. Shafer: Parallel Mining of Association Rules. *IEEE Transactions on Knowledge and Data Engineering*, 8(6) (1996): 962-969.

[3] J. Albert, Theoretical Foundations of Schema Restructuring in Heterogeneous Multidatabase Systems. In: *Proceedings of International Conference on Information and Knowledge Management*, 2000: 461-470.

[4] J. Aronis et al, The WoRLD: Knowledge discovery from multiple distributed databases. *Proceedings of 10th International Florida AI Research Symposium*, 1997: 337-341.

[5] P. Chan, An Extensible Meta-Learning Approach for Scalable and Accurate Inductive Learning. *PhD Dissertation*, Dept of Computer Science, Columbia University, New York, 1996.

[6] J. Chattratichat, et al., Large scale data mining: challenges and responses. In: *Proceedings of Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, (KDD-97), Newport Beach, California, USA, AAAI Press, August 14-17, 1997: 143-146.

[7] D. Cheung, V. Ng, A. Fu and Y. Fu, Efficient Mining of Association Rules in Distributed Databases, *IEEE Transactions on Knowledge and Data Engineering*, 8(1996), 6: 911–922.

[8] E. Han, G. Karypis and V. Kumar, Scalable Parallel Data Mining for association rules. In: *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1997: 277-288.

[9] A. Hurson, M. Bright, and S. Pakzad, *Multidatabase systems: an advanced solution for global information sharing.* IEEEComputer Society Press, 1994.

[10] H. Liu, H. Lu, and J. Yao, Identifying Relevant Databases for Multidatabase Mining. In: *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1998: 210–221.

[11] J. Park, M. Chen, P. Yu: Efficient Parallel and Data Mining for Association Rules. In: *Proceedings of International Conference on Information and Knowledge Management*, 1995: 31-36.

[12] A. Prodromidis, S. Stolfo. Pruning meta-classifiers in a distributed data mining system. In: *Proceedings of the First National Conference on New Information Technologies*, 1998: 151-160.

[13] A. Prodromidis, P. Chan, and S. Stolfo, Meta-learning in distributed data mining systems: Issues and approaches, In *Advances in Distributed and Parallel Knowledge Discovery*, H. Kargupta and P. Chan (editors), AAAI/MIT Press, 2000.

[14] J. Ribeiro, K. Kaufman, and L. Kerschberg, Knowledge discovery from multiple databases. In: *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (KDD-95), Montreal, Canada, AAAI Press, August 20-21, 1995: 240-245.

[15] T. Shintani and M. Kitsuregawa, Parallel mining algorithms for generalized association rules with classification hierarchy. In: *Proceedings of the ACM SIGMOD Conference on Management of Data*, 1998: 25-36.

[16] G. Webb, Efficient search for association rules. In: *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*, 2000: 99-107.

[17] D. Wolpert, Stacked Generalization. *Neural Networks*, 5(1992): 241–259.

[18] S. Wrobel, An algorithm for multi-relational discovery of subgroups. In: J. Komorowski and J. Zytkow (eds.) *Principles of Data Mining and Knowledge Discovery*, 1997: 367-375.

[19] Xindong Wu and Shichao Zhang, Synthesizing High-Frequency Rules from Different Data Sources. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 2, March/April 2003: 353-367.

[20] Shichao Zhang, Xindong Wu and Chengqi Zhang, Multi-Database Mining. *IEEE Computational Intelligence Bulletin*, Vol. 2, No. 1, June 2003: 5-13.

[21] Shichao Zhang, Chengqi Zhang and Xindong Wu, *Knowledge Discovery in Multiple Databases*. Springer, 2004.

[22] Shichao Zhang and Chengqi Zhang, Anytime Mining for Multi-User Applications. *IEEE Transactions on Systems, Man and Cybernetics (Part A)*, Vol. 32 No. 4(2002): 515-521.

[23] Chengqi Zhang and Shichao Zhang, Database Clustering for Mining Multi-Databases. In: *Proceedings of the 11th IEEE International Conference on Fuzzy Systems*, Honolulu, Hawaii, USA, May 2002.

[24] J. Yao and H. Liu, Searching Multiple Databases for Interesting Complexes. In: *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 1997: 198-210.

[25] N. Zhong, Y. Yao, and S. Ohsuga, Peculiarity oriented multi-database mining. In: *Principles of Data Mining and Knowledge Discovery*, 1999: 136-146.