Multi-Database Mining[1]

Shichao Zhang[2,3], Xindong Wu[4] and Chengqi Zhang[2]

***Abstract*— Multi-database mining is an important research area because (1) there is an urgent need for analyzing data in different sources, (2) there are essential differences between mono- and multi-database mining, and (3) there are limitations in existing multi-database mining efforts. This paper designs a new multi-database mining process. Some research issues involving mining multi-databases, including database clustering and local pattern analysis, are discussed.**

I. Introduction

THE increasing use of multi-database technology, such as computer communication networks and distributed, federated and homogeneous multi-database systems, has led to the development of many multi-database systems for real-world applications. For decision-making, large organizations need to mine the multiple databases distributed throughout their branches. In particular, as the Web is rapidly becoming an information flood, individuals and organizations can take into account low-cost information and knowledge on the Internet when making decisions. The data of a company is referred to as internal data whereas the data collected from the Internet is referred to as external data. Although external data assists in improving the quality of decisions, it generates a significant challenge: how to efficiently identify quality knowledge from multi-databases [26], [30], [31]. Therefore, large companies may have to confront the multiple data-source problem. Recently, the authors have developed local pattern analysis, a new multi-database mining strategy for discovering some types of potentially useful patterns that cannot be mined with traditional data mining techniques. Local pattern analysis discovers high-performance patterns from multi-databases.

There are two fundamental problems that prevent local pattern analysis from widespread applications. First, the data collected from the Internet is of poor quality that can disguise potentially useful patterns. For example, a stock investor might need to collect information from outside data sources when making an investment decision. If fraudulent information collected on the Internet is directly applied to investment decisions, the investor might lose money. In particular, much work has been built on consistent data. With distributed data mining algorithms it is assumed that the databases do not conflict with each other. However, reality is much more inconsistent, and inconsistency must be resolved before a mining algorithm can be applied. These observations generate a crucial requirement: data preparation.

The second fundamental problem is efficient algorithms for identifying potentially useful patterns in multi-databases. Over the years, there has been a lot of work in distributed data mining. However, traditional multi-database mining still utilizes mono-database mining techniques. That is, all the data from relevant data sources is pooled to amass a huge dataset for discovery. This can destroy useful patterns. For example, a pattern like "80% of the 15 supermarket branches reported that their sales increased 9% when bread and milk were frequently purchased" can often assist in decision-making at a central company level. However, mono-database mining techniques may miss such a pattern in the centralized database. On the other hand, using our local pattern analysis, there can be huge amounts of local patterns. These observations generate a strong requirement for the development of efficient algorithms for identifying useful patterns in multi-databases.

There are other essential differences between mono- and multi-database mining. Both data and patterns in multi-databases present more challenges than those in mono-databases. For example, unlike in mono-databases, data items in multi-databases may have different names, formats and structures in different databases. They may also conflict with each other.

In this paper we present a multi-database mining system through defining a new process for multi-database mining. The rest of this paper is organized as follows. Section II illustrates the role of multi-database mining in real-world applications. Section III describes multi-database mining problems. Section IV analyzes the differences between mono- and multi-database mining by demonstrating the features of data in mono- and multi-databases. Section V recalls the research into multi-database mining. Section VI designs a process for multi-database mining. Section VII discusses the features of our proposed multi-database mining.

II. Multi-Database Mining in Real-World Applications

Business, government and academic sectors have all implemented measures to computerize all, or part of, their daily functions [12]. An interstate (or international) company consists of multiple branches. The National Bank of Australia, for example, has many branches in different locations. Each branch has its own database, and the bank data is widely distributed and thus becomes a multi-database problem (see Fig. 2).

In Fig. 2, the top level is an interstate company (IC). This IC is responsible for the development and decision-making for

[1]This research has been partially supported by the Australian Research Council under grant number DP0343109, the Guangxi Natural Science Funds, and the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number DAAD19-02-1-0178.

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

[3]State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, China

[4]Department of Computer Science, University of Vermont, Burlington, Vermont 05405, USA xwu@cs.uvm.edu

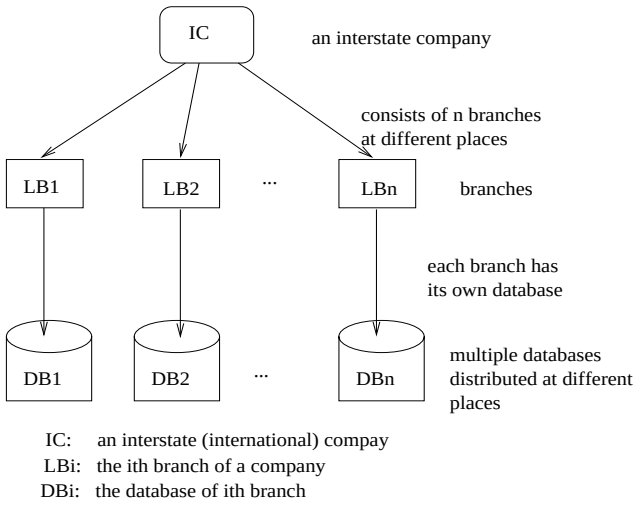

Fig. 2. An interstate company and its branches

the entire company. The middle level consists of n branches LB_1, LB_2, $\cdots$, LB_n. The bottom level consists of n local databases DB_1, DB_2, $\cdots$, DB_n of the n branches.

Fig. 2 illustrates the structure of a two-level interstate company. In the real world, the structure of an interstate company is usually more complicated, and each branch may also have multi-level sub-branches.

Many organizations have a pressing need to manipulate all the data from their different branches rapidly and reliably. This need is very difficult to satisfy when the data is stored in many independent databases, and the data is all of importance to an organization. Formulating and implementing queries requires data from more than one database. It requires knowledge of where all the data is stored, mastery of all the necessary interfaces and the ability to correctly combine partial results from individual queries into a single result.

To respond to these demands, researchers and practitioners have intensified efforts on developing appropriate techniques for utilizing and managing multi-database systems. Hence, developing multi-database systems has become an important research area.

Also, the computing environment is becoming increasingly widespread through the use of Internet and other computer communication networks. In this environment, it has become more critical to develop methods for building multi-database systems that combine relevant data from many sources and present the data in a form that is comprehensible for users, and provide tools that facilitate the efficient development and maintenance of information systems in a highly dynamic and distributed environment. One important technique within this environment is the development of multi-database systems. This includes managing and querying data from the collections of heterogeneous databases.

While multi-database technology can support many multi-database applications, it would be useful and necessary to mine these multi-databases to enable efficient utilization of the data. Thus, the development of multi-database mining is both a challenging and critical task.

Some essential differences between mono- and multi-database mining will be demonstrated below. We will show that traditional multi-database mining techniques are inadequate for two-level applications within large organizations such as interstate companies.

III. Multi-Database Mining Problems

An interstate company often consists of multi-level branches. Without loss of generality, this paper simplifies each interstate company as a two-level organization (a central company and multiple branches), as depicted in Fig. 2. Each branch has a database and the database is simplified as a relation or a table for our mining purposes.

Fig. 2 can be used to demonstrate that there are fundamental differences between mono- and multi-database mining. For example, multi-database mining may be restricted by requirements imposed by two-level decisions: the central company's decisions (global applications) and branch decisions (local applications). For global applications and for corporate profitability, central company headquarters are more interested in patterns (rather than the original raw data) that have the support of most of its branches, and those patterns are referred to as high-vote patterns hereafter. In local applications, a branch manager needs to analyze the data to make local decisions.

Two-level applications in an interstate company are depicted in Fig. 3.

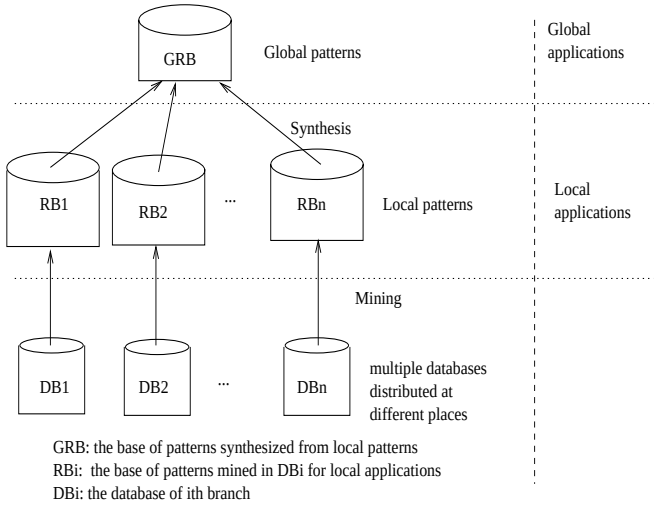

Fig. 3. Two level applications in an interstate company

In Fig. 3, the bottom level consists of n local databases DB_1, DB_2, $\cdots$, DB_n of n branches within an interstate company. The middle level consists of n sets RB_1, RB_2, $\cdots$, RB_n of local patterns discovered from databases DB_1, DB_2, $\cdots$, DB_n, respectively. These local patterns can be used for decision-making within branches (local applications). The top level is a set of global patterns that are synthesized from the n sets RB_1, RB_2, $\cdots$, RB_n. These global patterns are used for the overall company's decision-making (global applications).

One possible way for multi-database mining is to integrate all the data from these databases to amass a huge dataset for discovery by mono-database mining techniques. However, there are important challenges and difficulties involved in applying this method to real-world applications. We will discuss these challenges and difficulties in detail in Section V-B.

In Fig. 3 each database has been mined at each branch for use in local applications. While collecting all data together from different branches might produce a huge database and lose some important patterns for the propose of centralized processing, forwarding the local patterns (rather than the original raw data) to central company headquarters provides a feasible means of dealing with multiple database problems. The patterns forwarded from branches are called *local patterns*.

However, the number of forwarded local patterns may be so large that browsing the pattern set and finding interesting patterns can be rather difficult for central company headquarters. Therefore, it can be difficult to identify which of the forwarded patterns (including different and identical ones) are really useful at the central company level.

IV. Differences Between Mono- And Multi-Database Mining

The previous sections have indicated that there are essential differences between mono- and multi-database mining. This section illustrates these differences using the features of data and patterns in mono- and multi-databases.

A. Features of Data in Multi-databases

There are many ways to model a given real-world object (and its relationships with other objects) in, for example, an interstate company, depending on how the model will be used [12]. Because local databases are developed independently with differing local requirements, a multi-database system is likely to have many different models, or representations, for similar objects. Formally, a multi-database system is a federation of autonomous, and possibly heterogeneous, database systems used to support global applications and concurrent accesses to data stored in multiple databases [12].

We now illustrate data features in multi-databases.

1) *Name differences*. Local databases may have different conventions for the naming of objects, leading to problems with synonyms and homonyms.

A synonym means that the same data item has a different name in different databases. The global system must recognize the semantic equivalence of the items and map the differing local names to a single global name. A homonym means that different data items have the same name in different databases. The global system must recognize the semantic difference between items and map the common names to different global names.

2) *Format differences*. Many analysis or visualization tools require that data be in particular formats within branches. Format differences include differences in data type, domain, scale, precision, and item combinations.

An example is when a part number is defined as an integer in one database and as an alpha-numeric string in another. Sometimes data items are broken into separate components in one database while the combination is recorded as a single quantity in another.

Multi-database systems typically resolve format differences by defining transformation functions between local and global representations. Some functions may consist of simple numeric calculations such as converting square feet to acres. Others may require tables of conversion values or algorithmic transformations. A problem in this area is that the local-to-global transformation (required if updates are supported) may be very complex.

3) *Structural differences*. Depending on how an object is used in a database system, it may be structured differently in different local databases.

A data item may have a single value in one database and multiple values in another. An object may be represented as a single relation in one location or as multiple relations in another. The same item may be a data value in one location, an attribute in another, and a relation in a third. So the data often has discrepancies in structure and content that must be cleaned.

4) *Conflicting data*. Databases that model the same real-world object may have conflicts within the actual data values recorded.

One system may lack some information due to incomplete updates, system errors, or insufficient demands to maintain such data. A more serious problem arises when two databases record the same data item but assign it different values. The values may differ because of an error, or because of valid differences in the underlying semantics.

5) *Distributed data*. In most organizations, data is stored in various formats, in various storage media, and with various computers.

Therefore, data is created, retrieved, updated and deleted using various access mechanisms.

6) *Data sharing*. A major advantage of multi-database systems is the means by which branch data and sources can be shared.

In an interstate company, each of its branches has individual functions, data and sources. These branches can interact and share their data when they cannot solve problems that are beyond their individual capabilities.

7) *Data for two-level applications*. Comprehensive organizations have two-level decisions: central company's decisions (global applications) and branch decisions (local applications).

The above features demonstrate that data in multi-databases is very different from data in mono-databases.

B. Features of Patterns in Multi-databases

Generally, patterns in multi-databases can be divided into (1) local patterns, (2) high-vote patterns, (3) exceptional patterns, and (4) suggested patterns.

1) *Local patterns.* In an interstate company, local branches need to consider the original raw data in their databases so they can identify local patterns for local decisions.

Each branch of an interstate company has certain individual functions. The branch must design its own plan and policy for development and competition. It therefore needs to analyze data only in their local databases to identify local patterns. Each branch can then share these patterns with other branches. More importantly, they can forward their local patterns to the central company when global decisions need to be made.

2) *High-vote patterns.* These are patterns that are supported/voted for by most branches. They reflect common characteristics among branches and are generally used to make global decisions.

When an interstate company makes a global decision, the central company headquarters are usually interested in local patterns rather than original raw data. Using local patterns, they can learn what is supported by their branches. High-vote patterns are helpful in making decisions for the central company.

3) *Exceptional patterns.* These are patterns that are strongly supported/voted for by only a few branches. They reflect the individuality of branches and are generally used to create special policies specifically for those branches.

Although high-vote patterns are useful in reaching decisions for an interstate company, the headquarters are also interested in viewing the exceptional patterns used for making special decisions at only a few of the branches. Exceptional patterns may also be useful in predicting/testing the sales of new products.

4) *Suggested patterns*. These are patterns that have less votes than the minimal vote (written as $minvote$) but are very close to $minvote$.

The minimal vote is given by the user or a domain expert. It means that if a local pattern has votes equal to, or greater than, $minvote$, the local pattern becomes a global pattern, and is known as a high-vote pattern. Under the threshold $minvote$, there may be some local patterns that have less votes than $minvote$ but are very close to it. We call these patterns suggested patterns and they are sometimes useful in global decisions.

It is important to note that *local patterns also inherit the features of data in multi-databases*.

The above differences in data and patterns in multi-database systems demonstrate that multi-database mining differs from mono-database mining. This invites the exploration of efficient mining techniques for identifying novel patterns in multi-databases such that patterns can serve two-level applications in large organizations.

V. Related Work

A. Existing Research Efforts on Multi-Database Mining

If an interstate company is a comprehensive organization where its databases belong to different types of businesses and have different meta-data structures, the databases would have to be classified before the data is mined. For example, if a company like Coles-Myer has 25 branches including 5 supermarkets for food, 7 supermarkets for clothing, and 13 supermarkets for general commodities, these databases would first have to be classified into three clusters according to their business types before they are mined. Therefore, a key problem in multi-database mining is how to effectively classify multi-databases.

To mine multi-databases, the first method (mono-database mining technique) is to put all the data together from multiple databases to create a huge mono-dataset. There are various problems with this approach and we will discuss them in Section V-B.

In order to confront the size of datasets, Liu, Lu and Yao have proposed an alternative multi-database mining technique that selects relevant databases and searches only the set of all relevant databases [15]. 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 thus 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 huge database upon which existing data mining techniques or tools are applied. The approach is effective in reducing search costs for a given application.

Identifying relevant databases in [15] is referred to as database selection. In real-world applications, database selection needs, however, multiple times to identify relevant databases to meet different applications. In particular, the users may need to mine their multi-databases without specifying any application, and in this case, the database selection approach does not work. The database selection approach is application-dependent.

While data mining techniques have been successfully used in many diverse applications, multi-database mining has only been recently recognized as an important research topic in the data mining community. Yao and Liu have 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 [28].

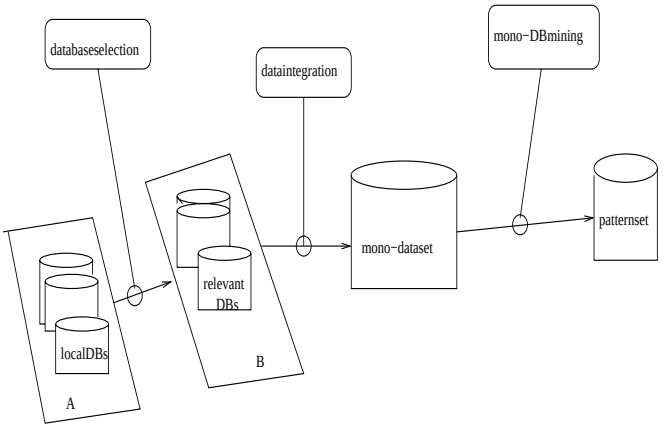

Fig. 4. The traditional process of multi-database mining

Based on [15], [28], Fig. 4 illustrates the functions used in existing multi-database mining. We call this process the traditional process. Area 'A' contains n sets of local databases

in an interstate company, where 'localDBs' stand for a set of local databases. 'databaseselection' is a procedure of the application-dependent database classification that identifies databases most relevant to an application. Area 'B' contains all databases that are relevant to an application. 'dataintegration' is a procedure that integrates all data in the relevant databases into a single dataset, called a 'mono-dataset'. Meanwhile, 'mono-DBmining' is a procedure that uses mono-database mining techniques to mine the integrated mono-dataset. 'patternset' is a set of the discovered patterns in the mono-dataset integration.

Zhong *et al.* have proposed a way of mining peculiarity patterns from multiple statistical and transaction databases based on previous work [31]. A peculiarity pattern is discovered from the peculiar data by searching the relevance among the peculiar data. Roughly speaking, a data item 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 looks like an exception pattern from the viewpoint of describing a relatively small number of objects, the peculiarity pattern represents a well-known fact with common sense, which is a feature of the general pattern.

A related research effort is distributed data mining (DDM) that deals with different possibilities of data distribution. A famous effort is hierarchical meta-learning [18] which has a similar goal of efficiently processing large amounts of data. Meta-learning starts with a distributed database or a set of data subsets of an original database, concurrently runs a learning algorithm (or different learning algorithms) on each of the subsets, and combines the predictions from classifiers learned from these subsets by recursively learning 'combiner' and 'arbiter' models in a bottom-up tree manner [18]. The focus of meta-learning is to combine the predictions of learned models from the partitioned data subsets in a parallel and distributed environment.

Other related research projects are now briefly reviewed. Wu and Zhang have advocated an approach for identifying patterns in multi-database by weighting [26]. Ribeiro, Kaufman and Kerschberg have 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 [20]. Wrobel has extended the concept of foreign keys to include foreign links, since multi-database mining also involves accessing non-key attributes [25]. Aronis *et al.* introduced a system called WoRLD that uses spreading activation to enable inductive learning from multiple tables in multiple databases spread across the network [4]. Kargupta *et al.* have built a collective mining technique for distributed data [14], [13]. Grossman *et al.* have built a system, known as Papyrus, for distributed data mining [9], [22]. Existing parallel mining techniques can also be used to deal with multi-databases [5], [7], [18], [19], [21].

The above efforts have provided good insights into multi-database mining. However, they are inadequate for identifying two new types of patterns: high-vote patterns and exceptional patterns, which reflect the distributions of local patterns.

B. Limitations of Mono-Database Mining for Dealing with Multiple Databases

Despite there being several methods of multi-database mining, most of them are still closely modeled on techniques for mono-database mining. This leads to a number of serious concerns and problems.

1) Due to the difficulty of data preparation, most work on multi-database mining has been built on quality data, and it is assumed that the data in different data sources is nicely distributed and contains consistent and correct values. However, existing data preparation focuses on single databases [29]. Because there are essential differences between multi- and mono-databases, there is a significant need of preparing the data in multi-databases.
2) Putting all the data from relevant databases into a single database can destroy some important information that reflects the distribution of patterns. These patterns may be more important than the patterns present in the single database in terms of global decision-making by a centralized company. Hence, existing techniques for multi-databases mining are inadequate.

We have provided an example in this regard in the introduction. In some cases, each branch of an interstate company, large or small, has equal power of voting for patterns involved in global decisions. For global applications, it is natural for the central company headquarters to be interested in the patterns voted for by most of the branches or exceptional patterns. It is therefore inadequate in multi-database mining to utilize existing techniques used for mono-databases mining.

3) Collecting all data from multi-databases can amass a huge database for centralized processing.

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. The French Teletel system has 1500 separate databases [12].

Parallel mining is sometimes unnecessary as there are many techniques such as sampling and parallel algorithms, for dealing with large databases.

A better approach is to first classify the multiple databases. The data from a class of databases can then be put into a single database for knowledge discovery utilizing existing techniques.

4) Forwarding all rules mined in branches to a central company. The number of forwarded rules may be so large that browsing the rule set and finding interesting rules from it can be a difficult task. In particular, it is more difficult to identify which of the forwarded rules are genuinely useful.

One strategy may be to reuse all the promising rules discovered in branches because the local databases have been mined for local applications. However, to reuse the local rules and select from them, a method must be developed to (1) determine valid rules for the overall organization from the

amassed database, and (2) reduce the size of the candidate rules from multi-databases. The following problems arise: (a) any rule from a database has the potential to contribute in the construction of a valid rule for the overall organization, and (b) the number of promising rules from multi-databases can be very large before it is determined which ones are of interest.

5) 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 environment is of extreme importance. Most multi-database designers take privacy very seriously, and allow for some protection facilities. For resource sharing in real-world applications, sharing patterns is a feasible way. This is because (1) certain data, such as commercial data, is secret for competition reasons; (2) reanalyzing data is costly; and (3) inexperienced decision-makers don't know how to confront huge amounts of data. The branches of an interstate company must search their databases for local applications. Hence, forwarding the patterns (rather than the original raw data) to the centralized company headquarters presents a feasible way to deal with multi-database problems.

Even though all of the above limitations might not be applicable to some organizations, efficient techniques, such as sampling and parallel and distributed mining algorithms, are needed to deal with the amassed mono-databases. However, sampling models depend heavily on the transactions of a given database being randomly appended to the database in order to hold the binomial distribution. Consequently, mining association rules upon paralleling (MARP), which employ hardware technology such as parallel machines to implement concurrent data mining algorithms, are a popular choice [2], [5], [8], [16], [17], [21]. Existing MARP efforts endeavor to scale up data mining algorithms by changing existing sequential techniques into parallel versions. These algorithms are effective and efficient, and have played an important role in mining very large databases. However, in addition to the above five limitations, MARP has two more limitations when performing data mining with different data sources.

6) MARP does not make use of local rules at branches; nor does it generate these local rules. In real-world applications, these local rules are useful for the local data sources, and would need to be generated in the first instance.
7) Parallel data mining algorithms require more computing resources (such as massive parallel machines) and additional software to distribute components of parallel algorithms among different processors of parallel machines. Most importantly, it is not always possible to apply MARP to existing data mining algorithms. Some data mining algorithms are sequential in nature, and can not make use of parallel hardware.

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

VI. MDM: A New Process for Multi-Database Mining

As previously explained, there are three factors that illustrate the importance of multi-database mining: (1) there are many multi-databases already serving large organizations; (2) there are essential differences between mono- and multi-database mining; and (3) there are limitations in existing multi-database mining techniques. For these reasons, we have designed a high-performance prototype system for multi-database mining (MDM). Below we introduce our MDM design through defining a new process of multi-database mining and describing its functions.

A. Three Steps in MDM

There are various existing data mining algorithms that can be used to discover local patterns in local databases [1], [11], [23]. These include the paralleling algorithms mentioned above [18], [21]. Our MDM process focuses on local pattern analysis as follows.

> Given n databases within a large organization, MDM performs three steps: (i) searching for a good classification of these databases; (ii) identifying two types of new patterns from local patterns: high-vote patterns and exceptional patterns; and (iii) synthesizing local patterns by weighting.

The major technical challenge in MDM is to serve the two-level applications in large organizations, such as interstate companies. MDM is depicted in Fig. 5.

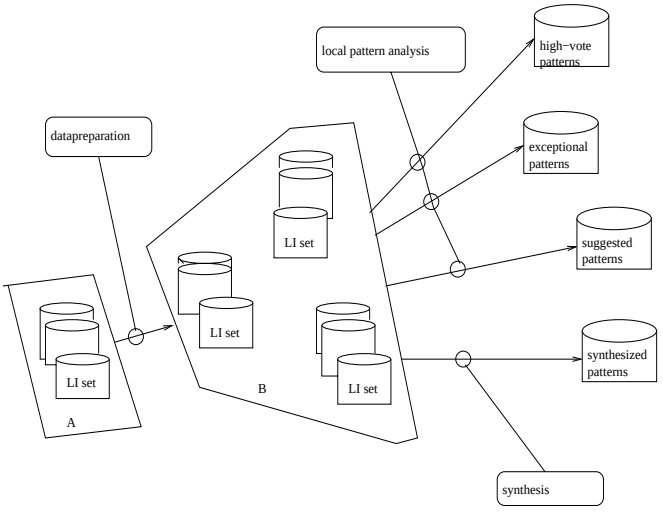

Fig. 5. The MDM process

In Fig. 5, area 'A' contains n sets of local patterns of an interstate company, where 'LIset' stands for a local pattern set; and 'datapreparation' is a procedure of application-independent database classification. After classifying the multi-databases, the local pattern sets are divided into several groups in area 'B'. For each group of local pattern sets, we use procedure 'localpatternanalysis' to search for patterns, such as high-vote

patterns, exceptional patterns, and suggested patterns. Procedure 'synthesis' is used to aggregate the local patterns in each group.

B. Research Issues in the MDM Process

In Fig. 5, three procedures, 'datapreparing', 'localpatternanalysis', and 'synthesis' are needed, as well as other procedures, to unify names of items and remove noise. Although the problem of unifying names of items and removing noise is also faced by multi-database systems [12], our MDM process focuses on issues raised from the three procedures in Fig. 5.

1) Data preparation can be more time consuming, and can present more challenges than mono-database mining. The importance of data preparation can be illustrated by the following observations: (1) real-world data is impure; (2) high-performance mining systems require quality data; and (3) quality data yields concentrative patterns. Therefore, the development of data preparation technologies and methodologies is both a challenging and critical task.

There are four key problems in data preparation: (i) developing techniques for cleaning data, (ii) constructing a logical system for identifying quality knowledge from different data sources, (iii) constructing a logical system for resolving inconsistency in different data sources, and (iv) designing application-independent database clustering.

(a) Developing techniques for cleaning data. Data cleaning techniques have been widely studied and applied in pattern recognition, machine learning, data mining and Web intelligence. For multi-database mining, distributed data cleaning presents more challenges than traditional data cleaning for single databases. For example, data may conflict within multi-databases. We need the following techniques to generate quality data for multi-database mining.

- Recover incomplete data: filling missing values, or expelling ambiguity;
- Purify data: consistency of data names and data formats, correcting errors, or removing outliers (unusual or exceptional values); and
- Resolve data conflicts: using domain knowledge or expert decisions to settle discrepancy.

(b) Constructing a logical system to identify quality knowledge from different data sources. As we argued previously, sharing knowledge (rather than the original raw data) presents a feasible way to deal with different data source problems [26]. Accordingly, assume that a data source is taken as a knowledge base[1]; a company (or a branch of the company) is viewed as a data source; and a rule has two possible values in a data source: true (if the data source supports the rule) or false (otherwise).

In the Web environment, the database from a company and information from different websites (called external data sources) can be treated as different data sources. External data sources may be subject to noise, and therefore, if a data source (a company or a branch) wants to form its own knowledge for data mining applications, the data source needs the ability of refining external knowledge. To do so, we advocate a logical system for identifying quality knowledge that focuses on the following epistemic properties.

- Veridicality. Knowledge is true.
- Introspection. A data source is aware of what it supports and of what it does not support.
- Consistency. A data source's knowledge is non-contradictory.

(c) Constructing a logical system for resolving inconsistency in different data sources. Traditional (positive) association rules can only identify companionate correlations among items. It is desirable in decision-making to catch the mutually-exclusive correlations among items that are referred to as negative associations. Therefore, we have developed a new method for identifying both positive and negative association rules in databases [27]. Negative association rules can increase the quality of decisions. However, in a multi-database environment, negative association rules can cause inconsistency within databases.

(d) Designing application-independent database clustering. To perform an effective application-independent database classification, we will have to (1) construct measurements for database relevance, (2) construct measurements of good classifications, and (3) design effective algorithms for application-independent database classification.

2) To provide effective multi-database mining strategies for identifying new patterns, we will develop four techniques for searching for new patterns from local patterns, that is, (a) design a local pattern analysis strategy; (b) identify high-vote patterns; (c) find exceptional patterns; and (d) synthesize local patterns by weighting.

(a) *Designing a local pattern analysis strategy*. Using traditional multi-database mining techniques, we can identify patterns, such as frequent itemsets, association patterns and classification patterns, by analyzing all the data in a database cluster. However, as mentioned in the introduction, these techniques can lose useful patterns. Therefore, analyzing local patterns is very important for mining novel and useful patterns in multi-databases.

On the other hand, for a large company, the number of local patterns may, however, be so large that browsing the pattern set and finding interesting patterns from it can be a difficult task for the company headquarters. In particular, it is harder to identify which of the local patterns are genuinely useful. Therefore, analyzing local patterns is also a difficult task.

In a multi-database environment, a pattern has attributes such as the name of the pattern, the rate voted for by branches, and supports (and confidences for a rule) in branches that vote for the pattern. In other words, a pattern is a super-point of the form

$$P(name, vote, vsupp, vconf).$$

In our system, we have designed a local pattern analysis strategy in [29] by using the techniques in [30]. The key problem to be solved is how to analyze the diverse projections of patterns in a multi-dimension space consisting of local patterns within a company.

[1]If a data source contains only data, we can transform it into knowledge by existing mining techniques.

(b) *Identifying high-vote patterns*. Within a company, each branch, large or small, has a power to vote for patterns for global decision-making. Some patterns can receive votes from most of the branches. These patterns are referred to as high-vote patterns. These patterns may be far more important in terms of global decision-making within the company.

Because traditional mining techniques cannot identify high-vote patterns, these patterns are regarded as novel patterns in multi-databases. In our system, we have designed a mining strategy for identifying high-vote patterns of interest based on a local pattern analysis. The key problem to be solved in this mining strategy is how to post-analyze high-vote patterns.

(c) *Finding exceptional patterns*. Like high-vote patterns, exceptional patterns are also regarded as novel patterns in multi-databases. But an exceptional pattern receives votes from only a few branches. While high-vote patterns are useful when a company is making global decisions, headquarters are also interested in viewing exceptional patterns when special decisions are made at only a few of the branches, perhaps for predicting the sales of a new product. Exceptional patterns can capture the individuality of branches. Therefore, these patterns are also very important.

(d) *Synthesizing patterns by weighting*. Although each branch has a power to vote for patterns for making global decisions, branches may be different in importance to their company. For example, if the sale of branch A is 4 times of that of branch B in a company, branch A is more important than branch B in the company. The decisions of the company can be reasonably partial to high-sale branches. Also, local patterns may have different supports in different branches. We will need a new strategy for synthesizing local patterns based on an efficient model for synthesizing patterns from local patterns by weighting [26].

VII. Features of the MDM Process

The MDM process in Section VI provides a new way for building multi-database mining systems. The main features of this process are as follows.

1) New mining techniques and methodologies can significantly increase the ability of multi-database mining systems.

Previous techniques in multi-databases mining were developed to search for patterns using existing mono-database mining. Although data in multi-databases can be merged into a single dataset, such merging can lead to many issues such as tremendous amounts of data, the destruction of data distributions, and the infiltration of uninteresting attributes. In particular, some concepts, such as regularity, causal relationships and patterns cannot be discovered if we simply search a single dataset, since the knowledge is essentially hidden within the multi-databases [31]. It is a difficult task to effectively exploit the potential ability of mining systems and it is one of the issues essential to achieve the objective of designing effective mining strategies.

Our multi-database mining strategy is to identify two types of patterns, high-vote patterns and exceptional patterns, from analyzing local patterns. Because previous techniques search patterns in the same way as in existing mono-database mining, they cannot discover high-vote patterns and exceptional patterns in multi-databases. Therefore, the high-vote and exceptional patterns are regarded as novel patterns. In particular, the discovery of these patterns can capture certain distributions of local patterns and assist global decision-making within a large company.

2) New mining techniques and methodologies can significantly improve the performance of multi-database mining systems.

As we argued previously, an interstate company must confront two-level decisions: the company's decisions (global applications) and the branches' decisions (local applications). For global applications, the company headquarters must tackle huge amounts of data and local patterns. Therefore, the development of high-performance systems for mining multi-databases is very important.

The local pattern analysis strategies can deliver two direct benefits: greatly reduce search costs by reusing local patterns, and offer more useful information for global applications.

For efficient multi-database mining, a key problem is how to analyze the data in the databases so that useful patterns can be found to support various applications. We have mentioned two new strategies in dealing with this difficult problem. The first strategy is to design an efficient and effective application-independent database classification. The second strategy is to develop a local pattern analysis for identifying novel and useful patterns.

VIII. Conclusion

As pointed out in [31], most of the KDD methods that have been developed are on the single universal relation level. Although theoretically, any multi-relational database can be transformed into a single universal relation, practically this can lead to many issues such as universal relations of unmanageable sizes, infiltration of uninteresting attributes, loss of useful relation names, unnecessary join operations, and inconvenience for distributed processing. In particular, some concepts, regularity, causal relationships, and rules cannot be discovered if we just search a single database since the knowledge hides in multiply databases basically.

This paper has shown that the problem of multi-database mining is challenging and pressing. In particular, due to essential differences between mono- and multi-databases, we have defined a new process of multi-database mining for our system.

References

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

[2] R. Agrawal, J. Shafer: Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engneering*, 8(6) (1996): 962-969.

[3] J. Albert, Theoretical Foundations of Schema Restructuring in Heterogeneous Multidatabase Systems. In: *Proceedings of CIKM*, 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] J. Chattratichat, etc., Large scale data mining: challenges and responses. In: *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 1997: 143-146.

[6] P. Chan, An Extensible Meta-Learning Approach for Scalable and Accurate Inductive Learning. *PhD Dissertation*, Dept of Computer Science, Columbia University, New York, 1996.
[7] D. Cheung, J. Han, V. Ng and C. Wong, Maintenance of discovered association rules in large databases: an incremental updating technique. In: *Proceedings of International Conference on Data Engineering*, 1996: 106-114.
[8] D. Cheung, V. Ng, A. Fu and Y. Fu, Efficient Mining of Association Rules in Distributed Databases, *IEEE Trans. on Knowledge and Data Engg.*, 8(1996), 6: 911–922.
[9] R. Grossman, S. Bailey, A. Ramu, B. Malhi and A. Turinsky, The preliminary design of Papyrus: A system for high performance, distributed data mining over clusters. In: *Advances in Distributed and Parallel Knowledge Discovery*, AAAI Press/The MIT Press, 2000: 259-275.
[10] E. Han, G. Karypis and V. Kumar, Scalable Parallel Data Mining for association rules. In: *Proceedings of ACM SIGMOD*, 1997: 277-288.
[11] J. Han, J. Pei, and Y. Yin, Mining Frequent Patterns without Candidate Generation, *Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data SIGMOD'00)*, Dallas, TX, May 2000.
[12] A. Hurson, M. Bright, and S. Pakzad, *Multidatabase systems: an advanced solution for global information sharing.* IEEE Computer Society Press, 1994.
[13] H. Kargupta, W. Huang, K. Sivakumar, and E. Johnson, Distributed clustering using collective principal component analysis. *Knowledge and Information Systems*, 3(4) (2001): 422-448.
[14] H. Kargupta, W. Huang, K. Sivakumar, B. Park, and S. Wang, Collective Principal Component Analysis from Distributed, Heterogeneous Data. In: Principles of Data Mining and Knowledge Discovery, 2000: 452-457.
[15] 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.
[16] J. Park, M. Chen, P. Yu: Efficient Parallel and Data Mining for Association Rules. In: *Proceedings of CIKM*, 1995: 31-36.
[17] S. Parthasarathy, M. J. Zaki, W. Li, Memory placement techniques for parallel association mining. *Proceedings of International Conference on Knowledge Discovery and Data Mining* 1998: 304-308.
[18] A. Prodromidis, S. Stolfo. Pruning meta-classifiers in a distributed data mining system. In: *Proc. of the First National Conference on New Information Technologies*, 1998: 151-160.
[19] 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.
[20] J. Ribeiro, K. Kaufman, and L. Kerschberg, Knowledge discovery from multiple databases. In: *Proceedings of KDD95.* 1995: 240-245.
[21] T. Shintani and M. Kitsuregawa, Parallel mining algorithms for generalized association patterns with classification hierarchy. In: Proc. of ACM SIGMOD, 1998: 25-36.
[22] K. Turinsky and R. Grossman, A framework for finding distributed data mining strategies that are intermediate between centeralized strategies and in-place strategies. In: *Proceedings of Workshop on Distributed and Parallel Knowledge Discovery at KDD-2000*, 2000: 1-7.
[23] G. Webb, Efficient search for association rules. In: *Proceedings of ACM SIGKDD*, 2000: 99-107.
[24] D.H. Wolpert, Stacked Generalization. *Neural Networks*, 5(1992): 241–259.
[25] 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.
[26] X. Wu and S. 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.
[27] X. Wu, C. Zhang and S. Zhang, Mining Both Positive and Negative Association Rules. In: *Proceedings of 19th International Conference on Machine Learning*, Sydney, Australia, July 2002: 658-665.
[28] J. Yao and H. Liu, Searching Multiple Databases for Interesting Complexes. In: *Proc. of PAKDD*, 1997: 198-210.
[29] S. Zhang, Knowledge discovery in multi-databases by analyzing local instances. PhD Thesis, *Deakin University*, 2001.
[30] C. Zhang and S. Zhang, *Association Rules Mining: Models and Algorithms*. Springer-Verlag Publishers in Lecture Notes on Computer Science, Volume 2307, p. 243, 2002.
[31] N. Zhong, Y. Yao, and S. Ohsuga, Peculiarity oriented multi-database mining. In: *Proceedings of PKDD*, 1999: 136-146.