Animesh Adhikari*, Lakhmi C. Jain and Bhanu Prasad

A State-of-the-Art Review of Knowledge Discovery in Multiple Databases

DOI 10.1515/jisys-2015-0154
Received December 2, 2015; previously published online May 20, 2016.

Abstract: Knowledge discovery in multiple databases offers many opportunities and challenges. We have given a number of motivating points on knowledge discovery in multiple databases. In view of further studies on this aspect, we highlight some domains that generated numerous problems on multiple related databases. Activities related to data preprocessing in a multi-database mining environment are also discussed. Important techniques of mining multiple databases are outlined. Many interesting patterns that originated out of multi-database environments are highlighted. We shall witness more research outcomes and investigations as the number of multi-database domains is on the rise.

Keywords: Data preparation in multi-database environment, multi-database mining, patterns in multiple databases.

2010 Mathematics Subject Classification: 68T10.

1 Introduction

Knowledge discovery in multiple databases started in the middle of the 1990–1999 decade. Later on, many researchers contributed to this field and made it a prominent area of research in data mining. The investigations on this topic could be attributed to a variety of reasons, and some of them are mentioned as follows: (i) a single computer might take a very long time for mining a large database. In some cases, it may not be possible to complete the mining task within a given time frame. Thus, it might be difficult to mine a big dataset as a single data source using a computer. Szalaya et al. [56] tried to scale out by using multiple inexpensive ‘bricks’, rather than scale up to large, multiprocessor ‘mainframes’ and storage network arrays while handling datasets of order of petabyte. Thus, one may wish to divide and mine the individual component dataset, and then combine the mining results obtained from the component datasets. (ii) Time-based, region-based, or a few database-oriented knowledge discovery could be an important goal of many data analyses. Region-based data analyses of agricultural data are often required for big countries like the United States, China, and India. Also, large organizations operating across different countries and/or regions often seek for region-based or even time-based data analyses. Such analyses could result in regional features of data. The central meteorological department collects data from different states, while the state meteorological departments collect rainfall data from different parts of the state. In a high-dimensional dataset, dimension-specific comparisons could serve as a foundation of strategic decisions adopted. Such region-specific, dimension-specific [21], and time-specific data analyses are frequently done in many applications. (iii) As the requirements of our society are becoming more diversified, many real-world applications would generate multiple databases. Multi-sensors can generate multiple datasets. A large government project could collect data from multiple regions on multiple aspects. Different stock exchanges could generate multiple data series on the same stock

***Corresponding author: Animesh Adhikari,** Department of Computer Science, Parvatibai Chowgule College, Margao, Goa 403 602, India, e-mail: animeshadhikari@yahoo.com
Lakhmi C. Jain: School of Electrical and Information Engineering, University of South Australia, Mawson Lakes Campus, Australia
Bhanu Prasad: Department of Computer and Information Sciences, Florida A&M University, Tallahassee, FL 32307, USA

that is listed in different stock exchanges. More instances of this kind of situation are highlighted later on. (iv) Traditional data mining techniques [16, 32, 51] may not be suitable to mining multiple datasets because traditional data mining requires merging all the databases. This merger might prohibit the return of local patterns (local features) in different datasets. Thus, a local database may be required to be mined locally. (v) Some databases are inherently distributed. For example, when a data mining application requires integrating multi-party data [35], the privacy of data of each party needs to be respected. These data are neither to be amalgamated nor are they desired to be centralized. Also, we need to protect the privacy of sensitive data.

Nowadays, there are several sources available for multiple databases. A few domains, which have generated different problems on multiple related databases, are discussed below.

1.1 Multi-relational Tables

Multi-relational (MR) data mining approaches look for patterns that involve multiple tables (relations) from a relational database [22, 29]. It includes MR association rule discovery, MR decision trees, MR distance-based methods, etc. Domingos [26] has presented a survey on MR data mining, and held discussions on several areas such as WWW, counter-terrorism, virtual marketing, social networks, computational biology, and ubiquitous computing that include relational data mining. Spyropoulou et al. [55] have introduced a new syntax for MR patterns in MR data, and the RMiner algorithm to mining them efficiently.

1.2 Time-based Databases

Time-stamped data are a natural domain for data analyses. Most of the data generated are related to time either directly or indirectly; hence, the data analyses using time-stamped data has become an important area of study. Another important domain is health-care data where most health-care data include a time stamp specifying the temporal representation of patient information. Nigrin and Kohane [49] demonstrated how their previously described data retrieval application, DXtractor, can be used as a database querying application with expressive power close to that of temporal databases and temporal query languages. They used standard SQL and existing time-stamp-based repositories only. Adhikari [12] studied various problems related to multiple time-stamped data.

1.3 Distributed Datasets

Distributed data mining (DDM) algorithms deal with mining multiple databases distributed over different geographical regions. In the last few years, researchers have started addressing problems where the databases are stored at different places that cannot be moved to a central storage area for a variety of reasons. The DDM environment often comes with different distributed sources of computation. Ubiquitous computing [30], sensor networks [65], grid computing [59], and privacy-sensitive multiparty data [35] present examples where centralization of data is either not possible or at least not always desirable.

1.3.1 Ubiquitous Computing

There is no doubt that ubiquitous computing could be the next wave of computing. We experienced the first wave of computing due to the excessive use of mainframes in both academia and industries. Each mainframe is shared by many people. Now, we are in the personal computing era where a person and a machine face each other uncomfortably across the desktop. Moreover, sometimes, a person needs to spend hours to finish the task. It makes the person tiresome. Next comes ubiquitous computing, or the age of *calm* technology, when technology recedes quietly into the background of our lives. As opposed to the desktop paradigm, in which a single user consciously engages a single device for a specialized purpose, someone using ubiquitous computing engages many computational devices and systems simultaneously, in the course of ordinary activities, and may not necessarily even be aware that they are doing so.

1.4 Spatial Databases

Miller and Han [42] provided a scenario for discovering knowledge in spatial databases. Lazarevic and Obradovic [39] have applied techniques to discover knowledge using centralized and distributed learning from spatial heterogeneous databases. The centralized algorithm consists of a spatial clustering followed by local regression aimed at learning the relationships between the driving attributes and the target variable inside each region identified through clustering. In this distributed learning, similar regions in multiple databases are first discovered by applying a spatial clustering algorithm independently on all the sites, then identifying the corresponding clusters on the participating sites.

1.5 Multiple Stream Data

Sometimes, data streams are collected from multiple sensors. An important characteristic of stream data is that it never ends. Knowledge discovery in multiple data streams is an interesting problem for research [60]. Krempl et al. [36] identified challenges in covering the full cycle of knowledge discovery for data stream mining research, such as protecting data privacy, dealing with legacy systems, handling incomplete and delayed information, analysis of complex data, and evaluation of stream mining algorithms.

As many domains have already emerged, knowledge discovery in multiple related databases becomes an important area of data mining research. With the availability of multiple related databases, and due to the constraints as mentioned above, knowledge discovery in multiple databases is gaining more importance.

The rest of the paper is organized as follows. Section 2 provides various data preparation techniques often required in a multi-database environment. Then, in Section 3, we present a few techniques for mining multiple related databases. Over time, many patterns are reported in a multi-database environment, and they are discussed in Section 4. Some related works are discussed in Section 5, and we conclude the article in Section 6.

2 Data Preparation

There are numerous applications that require collecting and processing of a large quantity of data coming from different sources (see Section 3). Data preparation becomes a common task in most of the data mining applications. It could be a difficult task to identify a universal procedure for analyzing the data because the characteristics of data might vary over the domains, and the objectives of data preparation may vary from one application to another. In order to make the inspection and the processing of information easy and efficient, it is useful to transform the raw data into a processed form. This kind of analysis is usually preliminary, or could be viewed as complementary to the execution of different tasks, such as visual data exploration, data mining and summarization, and knowledge extraction and reasoning. Data preprocessing is an important step for deriving meaningful knowledge inherent in databases. In the following, we discuss different data preparation activities that are often applied to a multi-database environment.

2.1 Preparation of Data Warehouses

Each branch database needs to be preprocessed to make multi-databases suitable for data mining. It could well be that all the data sources are not in the same format. Sometimes, data need to be converted from one type to another. The data need to be processed before any mining task takes place. A few important steps for preparing the data at a branch are aggregation, sampling, dimensionality reduction, feature subset selection, feature creation, discretization, and variable transformation [57]. Relevant data are required to be retained for the purpose of mining. Also, the definitions of data are required to be the same at every data source. The preparation of data warehouse at every branch of the organization could be a significant task [7, 52, 60].

2.2 Temporal Aggregation

In historical databases, temporal aggregation is a process in which the timeline is partitioned and the values of various attributes in the database are accumulated over these partitions [27]. A typical example of temporal aggregation is the monthly accumulation of salary payment. Due to the large variety of temporal data and their distribution over the timeline, efficient algorithms to perform temporal grouping are required. Moon et al. [48] proposed several methods for large-scale temporal aggregation. In this context, the choice of time granularity is an important issue, as the characteristics of temporal patterns are heavily dependent on this parameter. Let us consider an online shop that acquires monthly reports from their web hosts. The web hosts deliver the activity reports at regular intervals. Here, time granularity refers to a month instead of a year. Therefore, for this application, month-wise time-stamped data could be accumulated to form smaller databases. Similarly, for stock market applications, weekly data accumulation may be preferred over monthly data.

2.3 Partitioning the Database

Partitioning a large database becomes an important task in some cases. Consider an organization possessing data over 50 consecutive years. The organization might be interested in mining the knowledge for various activities such as finding the items whose supports are stable over the time [15], and extracting yearly periodic patterns [14]. In order to extract such knowledge, one could divide the given database into a number of yearly databases. Also, for the purpose of mining change [20], one may require partitioning a given database. Thus, a single data source generates multiple databases. In a prediction problem, one requires analyzing the past events that would originate from the previous databases. Given such a problem, it might be strategically necessary to divide a large database into smaller databases. While dividing a large database, one requires selecting a certain time period. Selection of time period is an important decision, and it is dependent on the problem. For many problems, one could divide the database into yearly databases [13–15]. One needs to consider the time granularity as 1 year for certain problems because a season re-appears on a yearly basis and the customers' purchase patterns might vary from season to season.

2.4 Database Thinning

Database thinning refers to the process of discarding the items (in the transactions) that are not relevant to a given context. These items could be treated as outliers. The thinning process makes the size of transactions shorter; thus, the mining process becomes easier. In order to estimate the association between the select items, or, to find the patterns of select items in multiple databases, we may wish to discard other items in the local databases. One could apply database thinning to each local database [3].

2.5 Ordering of Databases

In the context of mining multiple large databases, the order of mining each local database seems to be an important issue. Mining multiple large databases could be thought of as a two-step process: mining each local database using a single database mining technique (SDMT) by applying an ordering method or a model, then synthesizing the patterns (derived from that mining) using an algorithm. By applying the above strategy, we proposed the pipelined feedback technique (PFT) [11] for mining multiple large databases.

2.6 Selection of Databases

An approximate form of knowledge resulting from a large number of databases would be adequate for many decision support applications. In this sense, the selection of databases might be important in many decision support applications as it reduces the cost of searching for necessary information. Their selection could be based on the inherent knowledge residing in the databases. For that purpose, we need to mine each local database. Then, we process the local patterns in different databases for the purpose of selecting the relevant databases. Based on the local patterns, one could cluster the local databases for processing relevant queries [10, 61].

2.7 Integration of Databases

In the context of a multi-database environment, different databases are required to be integrated with the data mining system. We need to respect the rights associated with each data set, as we deal with company's own data, third-party data, and customers' data. Lu [41] discussed the problem of seamless integration of data mining with DBMS and applications from three directions. These days, most of the database management systems have extended their functionality in data analysis. Such capability should be fully explored to develop DBMS-aware data mining algorithms. Another type of integration includes algorithm selection and parameter setting. Reducing or eliminating mining parameters as much as possible, and developing automatic or semi-automatic mining algorithm selection techniques will greatly increase the application friendliness of data mining systems. Lastly, standardizing the interface among databases, data mining algorithms, and applications can also facilitate the integration to certain extent.

2.8 Other Data Preparation Activities

Primitive data preprocessing techniques such as data cleaning, data transformation, data reduction, and data summarization make data fit for the mining tasks [31].

3 Mining Multiple Databases

Let us discuss some major techniques that have been reported in the last two decades dealing with multiple big databases. There are two types of techniques for mining multiple databases: (i) techniques that employ mining each database in a distributed manner, and combining the results (patterns) afterwards; (ii) techniques that employ mining each database separately, not necessarily in a distributed way. The first type of techniques originated because of the inability to move data into one place because of privacy or some other reasons. However, the latter category of techniques appeared because of reasons such as retaining local features (patterns) of data, requirement for approximate results, and inability to handle multiple databases at a time. Some techniques/approaches are discussed to deal with multiple large databases.

3.1 Local Pattern Analysis

In the context of a multi-database environment, the patterns in databases could be classified into local patterns, global patterns, and patterns that are neither local nor global. Patterns in a local (branch) database are termed as *local patterns*. Local patterns can be used for local data analysis and decision making problems [10, 57]. *Global patterns* are based on all the databases that are under consideration. They are useful for global data analyses [7, 58]. One could mine the global patterns by mining each local database, then by analyzing all the local patterns in different databases. This technique is simply called *local pattern analysis*. Zhang et al. [64] designed local pattern analysis for the purpose of addressing various problems related to multiple large databases.

Let us assume that there are n branches in a multi-branch company. Also, let D_i be the database corresponding to the i-th branch, $i = 1, 2, \ldots, n$. The essence of mining the multiple databases using local pattern analysis could be explained using Figure 1 (see Chapter 1 of Ref. [8] for more details).

Here, LPB_i denotes the local pattern base for D_i, $i = 1, 2, \ldots, n$. Multi-database mining using local pattern analysis could be considered as an approximate method for mining multiple large databases.

An extended model of local pattern analysis was reported by Adhikari and Rao [7]. It suggested an idea of mining the patterns in multiple databases using a systematic approach. In order to enhance the quality of knowledge, it might be required to synthesize a huge amount of patterns in different local databases. There is need to store the patterns efficiently. Thus, IS coding [10] and ACP coding [9] for storing the local frequent itemsets and local association rules, respectively, are proposed.

3.2 Pipelined Feedback Technique

Consider a multi-branch organization with n branch databases. Let W_i be the data warehouse corresponding to the i-th branch, $i = 1, 2, \ldots, n$. Then, the local patterns for the i-th branch are extracted from W_i, $i = 1, 2, \ldots, n$. We mine each data warehouse using an SDMT. PFT is elaborated using Figure 2 (see Chapter 6 of Ref. [4] for more details).

In PFT, W_1 is mined using an SDMT and the local pattern base LPB_1 is extracted. While mining W_2, all the patterns in LPB_1 are extracted irrespective of their values of interestingness measures, such as minimum support and minimum confidence. Apart from these patterns, some new patterns that satisfy the user-defined

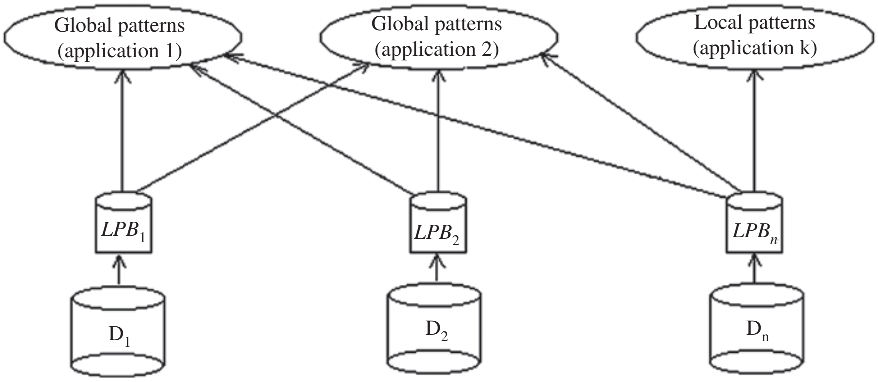
Global patterns (application 1)
Global patterns (application 2)
Local patterns (application k)
LPB_1
LPB_2
LPB_n
D_1
D_2
D_n

Figure 1: Mining Patterns in Multiple Databases Using Local Pattern Analysis.

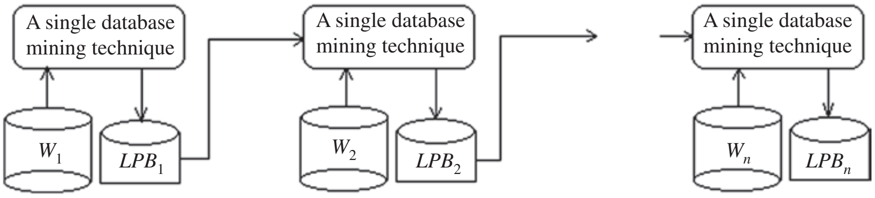
A single database mining technique
A single database mining technique
A single database mining technique
W_1
LPB_1
W_2
LPB_2
W_n
LPB_n

Figure 2: PFT of Mining Multiple Databases.

threshold values of interestingness measures are also extracted. In general, while mining W_i, all the patterns in W_{i-1} are mined irrespective of their values of interestingness measures, and some new patterns that satisfy the user-defined threshold values of interestingness measures, $i = 2, 3, \ldots, n$, are extracted. Due to this nature of mining each data warehouse, PFT is called a feedback technique. Thus, $|LPB_{i-1}| \leq |LPB_i|$, for $i = 2, 3, \ldots, n$. There are n arrangements of pipelining for n databases. All the arrangements of data warehouses might not produce the same mining result. If the number of local patterns increases, we get more accurate global patterns and a better analysis of local patterns. An arrangement of data warehouses would produce near optimal result if $|LPB_n|$ is maximal. Let size(W_i) be the size of W_i (in bytes), $i = 1, 2, \ldots, n$. We shall follow the following rule of thumb regarding the arrangements of data warehouses for the purpose of mining. The number of patterns in W_i is greater than or equal to the number of patterns in W_{i-1} if size(W_i) ≥ size(W_{i-1}), $i = 2, 3, \ldots, n$. For the purpose of increasing the number of local patterns, W_i precedes W_{i-1} in the pipelined arrangement of mining data warehouses if size(W_i) ≥ size(W_{i-1}), $i = 2, 3, \ldots, n$. Finally, we analyze the patterns in LPB_1, LPB_2, …, and LPB_n for synthesizing the global patterns or for analyzing the local patterns.

Let W be the collection of data from all the warehouses. For synthesizing the global patterns in W, we discuss here a simple pattern synthesizing formula. Without any loss of generality, let the itemset X be extracted from the first m databases, for $1 \leq m \leq n$. Then, the synthesized support of X in W could be obtained as follows:

$$\mathrm{supp}_s(X, W) = \frac{1}{\sum_{i=1}^{n} |W_i|} \times \sum_{i=1}^{m} [\,\mathrm{supp}_a(X, W_i) \times |W_i|]. \tag{1}$$

3.3 Distributed Approaches

Applications with distributed data are very common. Data analysis using distributed data is a challenging issue. A wireless ad hoc network can have multiple sensors that collect data continuously. Moving such data to a central location may not be a desirable objective. Thus, computation must be done in a distributed manner. An earth observatory system may consist of multiple satellites, and they send the data to their control centers. A distributed data analysis might be a critical approach to gain knowledge inherent in these datasets. Park and Kargupta [50] presented different issues on DDM algorithms. Tsoumakas [58] discussed some classical distributed problems such as distributed classification and regression, distributed association rule mining, and distributed clustering.

3.4 Other Techniques

Khiat et al. [33] applied the maximum entropy method to query selectivity estimation, and it consists of the following steps:

(i) Construct a graph clique tree structure to gather the common distributions.
(ii) Apply an iterative scaling algorithm as a convergence of maximum entropy solution for reducing the complexity estimation.
(iii) Perform the bucket elimination technique for each clique tree cluster for accelerating the estimation of common factors in the scale.

Mining multiple databases could be done using *partition algorithm* proposed by Savasere et al. [54]. Each database may require to be partitioned because a local database could be large. Thus, the original partition algorithm may need to be modified. The algorithm was designed for mining a very large database by partitioning it. The algorithm was stated as follows. It scans the database twice. The database is divided into disjoint partitions, where each partition is small enough to fit in the memory. In the first scan, the algorithm reads each partition and computes the locally frequent itemsets in each partition using the Apriori algorithm [16]. In the second scan, the algorithm counts the supports of all locally frequent itemsets toward the complete database. In this case, each local database could be considered as a partition. Though this partition algorithm mines frequent itemsets in a database exactly, it might be an expensive solution to mine multiple big databases because each database is required to be scanned twice. During the time of the second scanning, all the local patterns obtained during the first scan are analyzed. It remains an expensive technique for mining multiple big databases.

Wu and Zhang [62] have proposed the *RuleSynthesizing* algorithm for synthesizing high-frequency association rules in multiple databases. Using this technique, every local database is mined separately at *random order* using an SDMT for synthesizing high-frequent association rules. A pattern in a local database is assumed as zero if it does not get reported. Based on the association rules in different databases, the authors have estimated the weights of different databases. Let w_i be the weight of the i-th database, $i = 1, 2, \ldots, n$. Without any loss of generality, let the association rule r be extracted from the first m databases, $1 \leq m \leq n$. $supp_a(r, D_i)$ has been assumed as 0, $i = m+1, m+2, \ldots, n$. Here, D_i represents the i-th database, $i = 1, 2, \ldots, n$. Then, the support of r in D has been synthesized as follows:

$$supp_s(r, D) = w_1 \times supp_a(r, D_1) + \ldots + w_m \times supp_a(r, D_m), \tag{2}$$

where D represents the union of local databases. The algorithm *RuleSynthesizing* is an indirect approach for synthesizing the association rules in multiple databases. The authors used the same weights for local databases while synthesizing confidence of r in D. It requires further elaboration whether the same weights are justifiable to conditional probabilities while synthesizing the confidence of a rule.

Big data [18] is a broad term used for datasets that are so large or complex that traditional data processing applications are inadequate to process them. One could assume here that a big database cannot be mined using a traditional data mining technique. A technique similar to PFT, presented in Section 3.2, could be designed for mining multiple big databases, and we wish to address this issue in our future work.

4 Patterns in Multiple Related Databases

When all the databases are allowed to be put together, and if it is possible to mine the combined database, then there is no difference between multi-database mining and mono-database mining. Due to the large size of a local database, multi-database mining becomes more challenging. Often, one needs to apply an approximate method of mining multiple large databases by making use of local patterns. As a result, we may encounter extreme types of patterns such as high-frequency association rule [62], heavy association rules [7], and exceptional global patterns [2] while mining multiple databases using local pattern analysis.

As the transaction identifiers (ids) in the database tables are unique and would not usually be frequent, mining frequent patterns with transaction ids showing the records they occurred in will provide an efficient way to mine frequent patterns in many types of databases, including multiple tables and distributed databases. Ezeife and Zhang [28] have proposed a set of algorithms, called TidFPs for mining frequent patterns, with their transaction ids in a single transaction database, in multiple tables, and a distributed database.

Zhu and Wu [73] have proposed DRAMA, a framework for discovering relational patterns across multiple databases. More specifically, given a series of data collections, the authors tried to discover patterns coming from different databases with the patterns' relationships satisfying the user-specified constraints. The method sought to build a hybrid frequent pattern tree (HFP-tree) from multiple databases, and mined patterns from the HFP-tree, by integrating the users' constraints into the pattern mining process.

Lan et al. [38] have proposed new kinds of patterns called rare utility itemsets, which consider not only the individual profits and quantities but also the common existing periods and branches of items in a multi-database environment. They have also proposed a two-phase algorithm, TP-RUI-MD, to discover rare utility itemsets.

Kum et al. [37] have proposed ApproxMAP to mine approximate sequential patterns, called consensus patterns, from large sequence databases in two steps. First, the sequences are organized into similarity groups, called clusters. Then, consensus patterns are mined directly from each cluster through multiple alignments.

A multi-domain sequential pattern is a sequence of events whose occurrence time is within a pre-defined time window. Given a set of sequence databases across multiple domains, Peng and Liao [51] have aimed at mining multi-domain sequential patterns. They have proposed the algorithm Naive, in which multiple sequence databases are joined as one sequence database for utilizing traditional sequential pattern mining algorithms (e.g. PrefixSpan). Due to the nature of join operations, the algorithm Naive incurs substantial computing overhead. Later, the authors have proposed improved algorithms without having any join operations for mining multi-domain sequential patterns.

Zhong et al. [72] proposed peculiarity rules as a new class of rules that can be discovered from a relatively low number of peculiar data by searching the relevance among that peculiar data. The authors illustrated that such peculiarity rules represent a typically unexpected and interesting regularity hidden in the databases.

Yan et al. [63] have introduced a new paradigm called ratio rule. Ratio rules are aimed at capturing the quantitative association knowledge. The authors have extended this framework to mine ratio rules from distributed and dynamic data sources. The authors have proposed an integrated method to mine ratio rules from distributed and changing data sources, by first mining the ratio rules from each data source separately through a novel, robust, and adaptive one-pass algorithm, then integrating the rules of each data source in a simple probabilistic model.

Zhang et al. [69] have proposed a non-linear method, named KEMGP (kernel estimation for mining global patterns), which adopts kernel estimation to synthesize the global patterns using local patterns. The authors also adopted a method to divide all the data in different databases, according to the attribute dimensionality, to reduce the total space complexity.

Global exceptional patterns are prominent patterns that are exhibited by a few local databases. Thus, it describes the interesting individuality of a few branches. Therefore, it is interesting to identify such patterns. Adhikari [2] has introduced two types of exceptional patterns in multiple databases, and a strategy for identifying one type of global exceptional patterns in multiple databases.

Principal component analysis is frequently used for constructing a reduced representation of data. This method often reduces the dimensionality of the original data by a large factor and constructs the features that capture the maximally varying directions in the data. Kargupta et al. [34] have proposed a technique of computing the collective principal component analysis from heterogeneous sites.

Many data analyses are based on the influence of some items on other items. In Adhikari et al. (Chapter 10 of Ref. [6]), the notion of the overall influence of a set of items on another set of items is presented. Using this notion, two algorithms for analyzing the influence involving specific items in a database are designed.

5 Related Studies

The concept of learning or discovering knowledge from multiple databases started almost two decades ago. Zhong and Ohsuga [67] generated multiple databases from a given database for the purpose of discovering concept clusters. Ribeiro et al. proposed AQ [53], an inductive learning program, with capabilities for incremental learning and constructive induction with respect to their prototype knowledge discovery system, called INLEN. When the primary key in one database appears as a field in another database, it is possible to discover knowledge linking those two databases without having to actually combine their data. AQ is an inductive learning program with capabilities for incremental learning and constructive induction. Aronis et al. introduced WoRLD [17], a system that uses spreading activation to enable inductive learning from multiple tables in multiple databases spread across the network. Liu et al. [40] has proposed a relevance measure in order to select relevant databases for multi-database mining. An algorithm, RelavantDB, is proposed for selecting relevant databases. This is how knowledge discovery or learning from multiple databases took place at the initial research phase of this topic.

Zhang worked on this topic extensively, and proposed local pattern analysis [64], and submitted his thesis in 2002 [68]. Zhang et al. have published a book on mining knowledge from multiple databases [70] and organized a special issue on local pattern data analysis [71].

In the decade of 2000–2009, Dzeroski et al. organized a series of workshops on MR data mining in conjunction with the Association for Computing Machinery's Knowledge Discovery and Data Mining (ACM KDD) conference [19, 23–25]. In the last decade, many applications as well as patterns have been reported at a faster rate. Therefore, an entire section (see Section 4) is devoted on this aspect.

Zhu et al. proposed a solution to bridge the local and global analysis by identifying and eliminating mislabeled data items from large or distributed datasets through local analysis and global incorporation [66]. Zhu et al. organized workshops on mining multiple information sources in conjunction with the ACM KDD conference and the Institute of Electrical and Electronics Engineers' International Conference on Data Mining [43–47].

Adhikari et al. [1, 5, 6, 8] worked on various issues related to multiple databases. The authors proposed a technique for clustering the databases based on inherent knowledge [10], enhancing the quality of knowledge from multiple databases by proposing ACP coding [9] and IS coding [10], and extending the model of local pattern analysis for the purpose of synthesizing association rules in multiple databases [7].

6 Conclusions

With the advent of network technologies, a large multi-branch company stores its transactions centrally rather than locally. It is a major shift in data storage policy of organizations. Naturally, the techniques reported for mining multiple big databases may find these situations unsuitable. Are the multi-database mining techniques really needed? In the age of big data, data analysis and pattern recognition in multiple databases may indirectly come as a means of handling big data. Knowledge discovery in multiple databases can serve us in many ways. Here are some applications.

(i) Miming a big database approximately by diving it into a number of manageable databases.
(ii) Dividing a big database into some region- or time-specific databases for region- or time-specific analyses.
(iii) Comparing and contrasting similar datasets.
(iv) Mining multiple large databases can serve as a basis of mining a large database incrementally. An increment dataset(s) could be a single file, or multiple files as a form of yearly database(s), region-specific database(s), or some other databases. These increment databases could be mined separately, and the knowledge combined with the initial databases to obtain the final result.
(v) Characterization of centralized data originated from multiple sources as a multi-sourced data.

With the development of science and technology as well as the requirements of modern society, the sources of multiple large databases will be on the rise. We shall be witnessing more domains and applications in the coming days.

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive comments on an earlier version of this paper.

Bibliography

[1] A. Adhikari, Knowledge discovery in databases with an emphasis on multiple large databases, Goa University, 2009. Link to thesis: http://shodhganga.inflibnet.ac.in/handle/10603/12532.

[2] A. Adhikari, Synthesizing global exceptional patterns in different data sources, *J. Intell. Syst.* **21** (2012), 293–323.

[3] A. Adhikari and J. Adhikari, Mining patterns of select items in different data sources, in: *Advances in Knowledge Discovery in Databases*, pp. 233–253, Springer, Switzerland, 2015.

[4] A. Adhikari and J. Adhikari, *Advances in knowledge discovery in databases*, Springer, Berlin, 2015.

[5] A. Adhikari and J. Adhikari, Mining patterns of different related databases, in: *Advances in Knowledge Discovery in Databases*, pp. 83–95, Springer, Switzerland, 2015.

[6] A. Adhikari, J. Adhikari and W. Pedrycz, Measuring influence of an item in time-stamped databases, in: *Data Analysis and Pattern Recognition in Multiple Databases*, Springer, Berlin, pp. 209–228, 2014.

[7] A. Adhikari and P. R. Rao, Synthesizing heavy association rules from different real data sources, *Pattern Recognit. Lett.* **29** (2008), 59–71.

[8] A. Adhikari, P. Ramachandrarao and W. Pedrycz, *Developing multi-database mining applications*, pp. 1–13, Springer, Berlin, 2010.

[9] A. Adhikari and P. R. Rao, Enhancing quality of knowledge synthesized from multi-database mining, *Pattern Recognit. Lett.* **28** (2007), 2312–2324.

[10] A. Adhikari and P. R. Rao, Efficient clustering of databases induced by local patterns, *Decis. Support Syst.* **44** (2008), 925–943.

[11] A. Adhikari, P. Ramachandrarao, B. Prasad and J. Adhikari, Mining multiple large data sources, *Int. Arab. J. Inf. Technol.* **7** (2010), 241–249.

[12] J. Adhikari, *Mining and Analysis of Time-stamped Databases*, PhD thesis, Goa University, 2014, Link to thesis: http://www.zantyecollege.ac.in/libraries/view/Dr.-Mrs.-Jhimli-Adhikari/35.

[13] J. Adhikari, P. R. Rao and W. Pedrycz, Mining icebergs in time-stamped databases, in: *Proceedings of Indian International Conferences on Artificial Intelligence*, pp. 639–658, IICAI, USA, 2011.

[14] J. Adhikari and P. R. Rao, Identifying calendar-based periodic patterns, in: *Emerging Paradigms in Machine Learning*, S. Ramanna, L. Jain and R. J. Howlett (Eds.), pp. 329–357, Springer, Berlin, 2011.

[15] J. Adhikari, P. R. Rao and A. Adhikari, Clustering items in different data sources induced by stability, *Int. Arab. J. Inf. Technol.* **6** (2009), 394–402.

[16] R. Agrawal and R. Srikant, Fast algorithms for mining association rules. in: *Proceedings of International Conference on Very Large Data Bases*, pp. 487–499, VLDB, Santiago, 1994.

[17] J. Aronis, V. Kolluri, F. Provost and B. Buchanan, The WoRLD: knowledge discovery from multiple distributed databases, in: *Proceedings of the Tenth International Florida AI Research Symposium*, pp. 337–341, FLAIRS, Florida, 1997.

[18] Big Data. https://en.wikipedia.org/wiki/Big_data.

[19] H. Blockeel and S. Dzeroski, Multi-relational data mining 2005: workshop report, *SIGKDD Explor.* **7** (2005), 126–128.

[20] M. Böttcher, F. Hoppner and M. Spiliopoulou, On exploiting the power of time in data mining, *SIGKDD Explor.* **10** (2008), 3–11.

[21] D. L. Donoho, High-dimensional data analysis: the curses and blessings of dimensionality, in: *Proceedings of AMS Conference on Math Challenges of the 21st Century*, American Mathematical Society, Los Angeles, 2000.

[22] S. Dzeroski, Multi-relational data mining: an introduction, *SIGKDD Explor.* **5** (2003), 1–16.

[23] S. Dzeroski and H. Blockeel, Multi-relational data mining 2004: workshop report, *SIGKDD Explor.* **6** (2004), 140–141.

[24] S. Dzeroski and L. D. Raedt, Multi-relational data mining: a workshop report, *SIGKDD Explor.* **4** (2002), 122–124.

[25] S. Dzeroski, L. D. Raedt and S. Wrobel, Multi-relational data mining 2003: workshop report, *SIGKDD Explor.* **5** (2003), 200–202.

[26] P. M. Domingos, Prospects and challenges for multi-relational data mining, *SIGKDD Explor.* **5** (2003), 80–83.

[27] M. Dumas, M. C. Fauvet and P. C. Scholl, Handling temporal grouping and pattern-matching queries in a temporal object model, in *Proceedings of CIKM*, pp. 424–431, ACM, New York, 1998.

[28] C. I. Ezeife and D. Zhang, TidFP: mining frequent patterns in different databases with transaction ID, in: *Proceedings of DaWaK*, pp. 125–137, 2009.

[29] P. A. Flach, Multi-relational data mining: a perspective, in *Proceedings of EPIA*, pp. 3–4, EPIA, Porto, 2001.

[30] A. Greenfield, *Everyware: the dawning age of ubiquitous computing*, 1st ed., New Riders Publishing, San Francisco, 2006.

[31] J. Han, M. Kamber and J. Pei, *Data mining: concepts and techniques*, 3rd ed., Morgan Kaufmann, Burlington, MA, 2011.

[32] J. Han, J. Pei and Y. Yiwen, Mining frequent patterns without candidate generation, in: *Proceedings of ACM SIGMOD Conference on Management of Data*, pp. 1–12, ACM, New York, 2000.

[33] S. Khiat, H. Belbachir and R. S. Ahmed, Probabilistic models for local patterns analysis, *JIPS* **10** (2014), 145–161.

[34] H. Kargupta, W. Huang, S. Krishnamurthy, B. Park and S. Wang, Collective PCA from distributed and heterogeneous data, in: *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases*, pp. 452–457, PKDD, Lyon, 2000.

[35] H. Kargupta, K. Liu and J. Ryan, Privacy sensitive distributed data mining from multi-party data, in: *Proceedings of Intelligence and Security Informatics*, pp. 336–342, Springer, Berlin, 2003.

[36] G. Krempl, I. Zliobaite, D. Brzezinski, E. Hüllermeier, M. Last, V. Lemaire, T. Noack, A. Shaker, S. Sievi, M. Spiliopoulou and J. Stefanowski, Open challenges for data stream mining research, *SIGKDD Explor.* **16** (2014), 1–10.

[37] H.-C. Kum, H. C. Chang and W. Wang, Sequential pattern mining in multi-databases via multiple alignment, *Data Mining Knowl. Discov.* **12** (2006), 151–180.

[38] G.-C. Lan, T.-P. Hong and V. S. Tseng, A novel algorithm for mining rare-utility itemsets in a multi-database environment, in: *Proceedings of the 26th Workshop on Combinatorial Mathematics and Computation Theory*, pp. 293–302, CMCT, Taiwan, 2007.

[39] A. Lazarevic and Z. Obradovic, Knowledge discovery in multiple spatial databases, *Neural Comput. Appl.* **10** (2002), 339–350.

[40] H. Liu, H. Lu and J. Yao, Toward multi-database mining: identifying relevant databases, *IEEE Trans. Knowl. Data Eng.* **13** (2001), 541–553.

[41] H. Lu, Seamless integration of data mining with DBMS and applications, in: *Proceedings of PAKDD*, pp. 3, PAKDD, Hong Kong, 2001.

[42] H. J. Miller and J. Han, Eds., *Geographic data mining and knowledge discovery*, 2nd ed., CRC Press, Boca Raton, FL, 2009.

[43] Mining Multiple Information Sources, 2007. Available at: citeseerx.ist.psu.edu

[44] Mining Multiple Information Sources, 2008. Available at: citeseerx.ist.psu.edu.

[45] Mining Multiple Information Sources, 2009. Available at: http://www.cse.fau.edu/~xqzhu/.

[46] Mining Multiple Information Sources, 2010. Available at: http://www.cse.fau.edu/~xqzhu.

[47] Mining Multiple Information Sources, 2011. Available at: http://www.cse.fau.edu/~xqzhu.

[48] B. Moon, I. F. V. Lopez and V. Immanuel, Efficient algorithms for large-scale temporal aggregation, *IEEE Trans. Knowl. Data Eng.* **15** (2003), 744–759.

[49] D. J. Nigrin and I.S. Kohane, Temporal expressiveness in querying a timestamp-based clinical database, *J. Am. Med. Inform. Assoc.* **7** (2000), 152–163.

[50] B. H. Park and H. Kargupta, Distributed data mining: algorithms, systems, and applications, in: *Data Mining Handbook*, pp. 341–358, Lawrence Erlbaum Associates, Denmark, 2002.

[51] W.-C. Peng and Z.-X. Liao, Mining sequential patterns across multiple sequence databases, *Data Knowl. Eng.* **68** (2009), 1014–1033.

[52] D. Pyle, *Data preparation for data mining*, Morgan Kaufmann, San Francisco, 1999.

[53] J. S. Ribeiro, K. A. Kaufman and L. Kerschberg, Knowledge discovery from multiple databases, in: *Proceedings of KDD*, pp. 240–245, AAAI, California, 1995.

[54] A. Savasere, E. Omiecinski and S. Navathe, An efficient algorithm for mining association rules in large databases, in: *Proceedings of the 21st International Conference on Very Large Data Bases*, pp. 432–443, VLDB, USA, 1995.

[55] E. Spyropoulou, T. D. Bie and M. Boley, Interesting pattern mining in multi-relational data, *Data Mining Knowl. Discov.* **28** (2014), 808–849.

[56] A. S. Szalaya , J. Grayb and J. Vandenberga, Petabyte scale data mining: dream or reality?, Technical Report MSR-TR-2002-84, Johns Hopkins University, 2002.

[57] P.-N. Tan, V. Kumar and M. Steinbach, *Introduction to Data Mining*, Pearson Education, London, 2006.

[58] G. Tsoumakas, Distributed data mining, *Encyclopedia of Data Warehousing and Mining*, pp. 709–715, IGI Global, Pennsylvania, 2009.

[59] B. Wilkinson, *Grid Computing: Techniques and Applications*, CRC Press, Boca Raton, FL, 2009.

[60] W. Wu and L. Gruenwald, Research issues in mining multiple data streams, in: *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining*, pp. 56–60, ACM, New York, 2010.

[61] X. Wu, C. Zhang and S. Zhang, Database classification for multi-database mining, *Inf. Syst.* **30** (2005), 71–88.

[62] X. Wu and S. Zhang, Synthesizing high-frequency rules from different data sources, *IEEE Trans. Knowl. Data Eng.* **14** (2003), 353–367.

[63] J. Yan, N. Liu, Q. Yang, B. Zhang, Q. Cheng and Z. Chen, Mining adaptive ratio rules from distributed data sources, *Data Mining Knowl. Discov.* **12** (2006), 249–273.

[64] S. Zhang, X. Wu and C. Zhang, Multi-database mining, *IEEE Comput. Intell. Bull.* **2** (2003), 5–13.

[65] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, Morgan Kaufmann, San Francisco, 2004.

[66] X. Zhu, X. Wu and Q. Chen, Bridging local and global data cleansing: identifying class noise in large, distributed data datasets, *Data Min. Knowl. Discov.* **12** (2006), 275–308.

[67] N. Zhong and S. Ohsuga, Discovering concept clusters by decomposing databases, *Data Knowl. Eng.* **12** (1994), 223–244.

[68] S. Zhang, *Knowledge discovery in multi-databases by analyzing local instances*, Ph.D. thesis, Deakin University, 2002.

[69] S. Zhang, X. You, Z. Jin and X. Wu, Mining globally interesting patterns from multiple databases using kernel estimation, *Exp. Syst. Appl.* **36** (2009), 10863–10869.

[70] S. Zhang, C. Zhang and X. Wu, *Knowledge discovery in multiple databases*, Springer, Berlin, 2004.

[71] S. Zhang and M. J. Zaki, Mining multiple data sources: local pattern analysis, *Data Min. Knowl. Discov.* **12** (2006), 121–125.

[72] N. Zhong, Y. Yao and M. Ohshima, Peculiarity oriented multidatabase mining, *IEEE Trans. Knowl. Data Eng.* **15** (2003), 952–960.

[73] X. Zhu and X. Wu, Discovering relational patterns across multiple databases, in: *Proceedings of ICDE*, pp. 726–735, IEEE, USA, 2007.