Mining Multiple Large Databases

Animesh Adhikari
Department of Computer Science, S. P. Chowgule College, Goa-403602, India
animeshadhikari@yahoo.com

P. R. Rao
Department of Computer Science and Technology, Goa University, Goa-403206, India
pralhaad@rediffmail.com

Jhimli Adhikari
Department of Computer Science, Narayan Zantye College, Goa-403529, India
Jhimli_adhikari@ yahoo.co.in

Abstract

Effective data analysis with multiple databases requires highly accurate patterns. But, local pattern analysis might extract low quality of patterns from multiple databases. Thus, it is necessary to improve mining multiple databases. In this paper, we propose a new technique of mining multiple databases. In this technique, each local database is mined using a traditional data mining technique in a particular order for synthesizing global patterns. The proposed technique improves quality of synthesized global patterns significantly. We conduct experiments on both real and synthetic datasets to judge effectiveness of the proposed technique.

1. Introduction

Many large companies operate from a number of branches located at different geographical regions. Each branch collects data continuously and data get stored locally. Thus, the collection of all branch databases might be large. Many decisions of a multi-branch company are based on data stored over the branches. The challenges involve in making good quality of decisions based on large volume of data distributed over the branches. It creates not only risks but also opportunities. One of the risks might be significant amount investment on hardware and software to deal with multiple large databases. Our objective is to provide good quality of knowledge by minimizing the risks.

The first question comes to our mind whether a traditional data mining technique [3], [6] could provide a good solution in dealing with multiple large databases. To apply a traditional data mining technique we need to amass all the branch databases together. A traditional data mining technique might not provide a good solution due to the following reasons.

- The company might have to invest heavily on hardware and software to deal with a large volume of data.
- A single computer might take unreasonable amount of time to mine a huge amount of data.
- It might be difficult to identify local patterns if a traditional data mining technique is applied on the entire dataset.

Thus, a traditional data mining technique might not be suitable in this situation. So, it is a different problem. Hence, it is required to be dealt with in a different way. In this situation local pattern analysis [13] could be a solution. Under this model of mining multiple databases, each branch requires to mine its database using a traditional data mining technique. Afterwards, each branch is required to forward the pattern base to the central office. Then, the central office would process the pattern bases collected from different branches. Due to the following reasons, the local pattern analysis alone might not be a judicious choice for mining multiple large databases.

- A synthesized global pattern might differ considerably from true global patterns.
- In context of multi-database mining, no pattern estimating technique has been reported so far for estimating a pattern in a local database.

For the purpose of mining multiple databases, one could apply *partition algorithm* (PA) proposed by Savasere et al. [9]. The algorithm was designed to mine a very large database by partitioning. The algorithm works as follows. It scans the database twice. The database is divided into disjoint partitions, where each partition is small enough to fit in memory. In a first scan, the algorithm reads each partition and computes locally frequent itemsets in each partition using apriori algorithm [3]. 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 partition algorithm mines frequent itemsets exactly, it is an

0-7695-3068-0/07 $25.00 © 2007 IEEE
DOI 10.1109/ICIT.2007.42

expensive solution to mining multiple large databases, since each local database is required to scan twice.

The above difficulties motivate us to propose a multi-database mining technique. There are two benefits of the proposed technique. Firstly, it improves significantly the accuracy of mining multiple large databases as compared to local pattern analysis. Secondly, it scans each local database only once.

For mining multiple databases, there are three situations: (i) Each of the local databases is small, so that a single database mining technique (SDMT) could mine the union of all databases. (ii) At least one of the local databases is large, so that a SDMT could mine every local database, but fail to mine the union of all local databases. (iii) At least one of the local databases is very large, so that a SDMT fails to mine each local database. We face challenges to handle the cases (ii) and (iii). The challenges are posed to us due to large size of some local databases.

A multi-database mining technique (MDMT) using local pattern analysis could be viewed as a two-step process M+S, explained as follows.

- Mine each local database using a SDMT by following a model M (step 1)
- Synthesize patterns using an algorithm S (step 2)

We use notation MDMT: M+S to represent above multi-database mining technique. We propose a MDMT that improves the quality of both synthesized patterns and analysis of local patterns. The proposed algorithm could handle the cases (ii) and (iii) reasonably well, and it requires mining each local database only once.

The rest of the paper is organized as follows. We discuss related work and define the problem in section 2. We propose a model for mining multiple databases in section 3. We define error of an experiment in section 4. In section 5, we provide experimental results.

2. Problem definition

Consider a multi-branch company that operates from n branches. Let D_i be the database corresponding to the i-th branch, for i = 1, 2, …, n. Let D be the union of all branch databases. Before presenting proposed model, we shall first study work related to this issue.

2.1. Related work

In the context of step 1 of a MDMT using local pattern analysis, Zhang et al. [11] have proposed algorithm *IdentifyExPattern* (IEP) for identifying global exceptional patterns in multi-databases. Every local database is mined separately at random order (RO) using a SDMT for synthesizing global exceptional patterns.

In the context of step 2 of a MDMT using local pattern analysis, Zhang et al. [11] have proposed a technique for synthesizing global patterns. In algorithm IEP, a pattern in a local database is assumed as zero, if it does not get reported. Let $supp_a(p, DB)$ and $supp_s(p, DB)$ be the actual (i.e, apriori) support and synthesized support of pattern p in database DB. Support of pattern p in D has been synthesized as follows.

$$supp_s(p, D) = \frac{1}{num(p)} \sum_{i=1}^{num(p)} \frac{supp_a(p, D_i) - \alpha}{1 - \alpha} \quad (1)$$

where, num(p) is the number of databases that report p at a given minimum support level (α).

An algorithm *Association-Rule-Synthesis* (ARS) for synthesizing association rules in multiple real databases has been proposed in [1]. For real databases, the trend of the customers' behaviour exhibited in one database is usually present in other databases. In particular, a frequent itemset in one database is usually present in some transactions of other databases even if it does not get extracted. The estimation procedure captures such trend and estimates the support of a missing association rule. Without any loss of generality, let the itemset X be extracted from first m databases, for $1 \le m \le n$. Then trend of X in first m databases could be expressed as follows.

$$trend^{1,m}(X \mid \alpha) = \frac{1}{\sum_{i=1}^{m} |D_i|} \times \sum_{i=1}^{m} [supp_a(X, D_i) \times |D_i|] \quad (2)$$

One could use the trend of X in first m databases for synthesizing support of X in D. Now we estimate support of X in each of the remaining databases by $\alpha \times trend^{1,m}(X \mid \alpha)$, for j = k + 1, k + 2, …, n. Then the synthesized support of X could be computed as follows.

$$supp_s(X, D) = \frac{trend^{1,m}(X \mid \alpha)}{\sum_{i=1}^{n} |D_i|} \times \left[(1 - \alpha) \times \sum_{i=1}^{m} |D_i| + \alpha \times \sum_{i=1}^{n} |D_i| \right] \quad (3)$$

In synthesizing high-frequent association rule, Wu and Zhang [10] have proposed an algorithm *RuleSynthesizing* (RS) for synthesizing high-frequent association rule in multiple databases. Based on the association rules in different databases, the authors have estimated weights of different databases. Let w_i be the weight of i-th database, for i = 1, 2, …, n. Without any loss of generality, let the association rule r be extracted from first m databases, for $1 \le m \le n$. $supp_a(r, D_i)$ has been assumed as 0, for i = m + 1, m + 2, …, n. Then support of r in D has been synthesized as follows.

$$supp_s(r,D) = w_1 \times supp_a(r,D_1) + \ldots + w_m \times supp_a(r,D_m) \quad (4)$$

In the context of mining multiple databases, Zhang [12], Zhang et al. [14] studied various strategies for mining multiple databases. Existing parallel mining techniques [2], [4] could also be used to deal with multiple large databases.

2.2. Our approach

We need to process local databases as they may not be at the right state for the mining task. Various data preparation techniques [8] like data cleaning, data transformation, data integration, and data reduction are applied to branch databases. We get processed database corresponding to (original) local database. Then we keep all the data that are relevant to data mining applications. Using a relevance analysis, we could detect outlier data [7] and are kept in a separate storage. After removing outlier data from a processed database we get desired data warehouse and the data in a data warehouse become ready for the mining task. Let W_i be the data warehouse corresponding to the i-th branch, for i = 1, 2, …, n. Then the local patterns for the i-th branch are extracted from W_i, for i = 1, 2, …, n. We mine each data warehouse using a SDMT. In figure 1, we propose a new model of mining multiple databases.

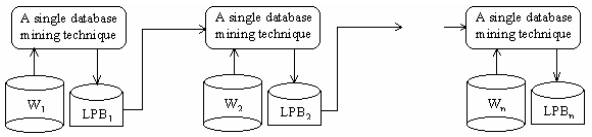

Figure 1. Pipelined feedback model (PFM) of mining multiple databases

In PFM, W_1 is mined using a SDMT and 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 like, minimum support and minimum confidence. Apart from these patterns, some new patterns that satisfy user defined 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 user defined threshold values of interestingness measures, for i = 2, 3, …, n. Due to this nature of mining each data warehouse, the technique is called a feedback model. Thus, $|LPB_{i-1}| \leq |LPB_i|$, for i = 2, 3, …, n. There are n! arrangements of pipelining for n databases. All the arrangements of data warehouses would 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 a maximal. Let $size(W_i)$ be the size of W_i (in bytes), for i = 1, 2, …, 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 $size(W_i) \geq size(W_{i-1})$, for i = 2, 3, …, n. For the purpose of increasing number of local patterns, W_i precedes W_{i-1} in the pipelined arrangement of mining data warehouses if $size(W_i) \geq size(W_{i-1})$, for i = 2, 3, …, n. Finally, we analyze the patterns in LPB_1, LPB_2, …, and LPB_n for synthesizing global patterns, or analyzing local patterns.

For synthesizing global patterns in D we discuss here a simple pattern synthesizing (SPS) algorithm. Without any loss of generality, let the itemset X be extracted from first m databases, for $1 \leq m \leq n$. Then synthesized support of X in D could be obtained as follows.

$$supp_s(X, D) = \frac{1}{\sum_{i=1}^{n} |D_i|} \times \sum_{i=1}^{m} [supp_a(X, D_i) \times |D_i|] \quad (5)$$

3. Mining multiple databases

In this section, we propose a new algorithm for mining multiple databases. The algorithm is based on the pipelined feedback model discussed in section 2.

Algorithm 1. Mine multiple data warehouses using pipelined feedback model.

procedure *PipelinedFeedbackModel* (W_1, W_2, …, W_n)
Input: W_1, W_2, …, W_n
Output: local pattern bases
01: **for** i = 1 to n **do**
02: **if** W_i does not fit in memory **then**
03: partition W_i into W_{i1}, W_{i2}, …, and Wip_i;
04: **else** $W_{i1} = W_i$;
05: **end if**
06: **end for**
07: sort data warehouses on size in non-increasing order and the data warehouses are renamed as DW_1, DW_2, …, DW_N, where $N = \sum_{i=1}^{n} p_i$;
08: **let** $LPB_0 = \phi$;
09: **for** i = 1 to N **do**
10: mine DW_i using a SDMT with input LPB_{i-1};
11: **end for**
12: return LPB_1, LPB_2, …, LPB_N;
end procedure

In above algorithm, the usage of LPB_{i-1} during mining DW_i has been explained in section 2.2. Once a pattern is extracted from a data warehouse, then it gets extracted from the remaining data warehouses. Thus, the algorithm *PipelinedFeedbackModel* improves

synthesized patterns and analysis of local patterns significantly.

4. Error of an experiment

To evaluate MDMT: PFM+SPS, we need to measure the amount of error of the experiments. An experiment mines frequent itemsets in local databases using PFM, and then synthesizes global patterns using SPS algorithm. We need to find how the global synthesized support differs from the exact support of an itemset.

Let $LPB_n = \{X_1, X_2, \ldots, X_m\}$. There are several ways one could define error of an experiment. We have defined following two types of error of an experiment.

1. Average Error (AE)

$$AE(D, \alpha) = \frac{1}{m} \Sigma_{i=1}^{m} \left| supp_a(X_i, D) - supp_s(X_i, D) \right| \quad (6)$$

2. Maximum Error (ME)

$$ME(D, \alpha) = maximum \left\{ \left| supp_a(X_i, D) - supp_s(X_i, D) \right|, i = 1, 2, \ldots, m \right\} \quad (7)$$

$supp_a(X_i, D)$ is obtained by mining D using a traditional data mining technique, for i = 1, 2, …, m. $supp_s(X_i, D)$ is obtained by SPS, for i = 1, 2, …, m.

5. Experiments

We have carried out several experiments to study the effectiveness of our approach. All the experiments have been implemented on a 2.8 GHz Pentium D dual core processor with 512 MB of memory using visual C++ (version 6.0) software. We present experimental results using synthetic dataset *T10I4D100K* (T) [5] and two real datasets *retail* (R) [5] and *BMS-Web-Wiew-1* (B) [5]. We present some characteristics of these datasets in table 1.

Let NT, AFI, ALT, and NI denote the number of transactions, average frequency of an item, average length of a transaction, and number of items in a database respectively. Each of the above datasets is divided into 10 databases for the purpose of carrying out experiments. The databases obtained from T, R and B are named as T_i, R_i, and B_i respectively, for i = 0, 1, …, 9. The databases T_i, R_i, and B_i are called input databases (DBs), for i = 0, 1, …, 9. Some characteristics of these input databases are presented in the table 2. In table 3, we present some outputs to show that the proposed technique improves significantly the mining results. We have performed experiments using other MDMTs on these datasets for the purpose of comparing with MDMT: PFM+SPS.

Table 1. Dataset characteristics

Dataset	N T	ALT	AFI	NI
T	1,00,000	11.10228	1276.12413	870
R	88,162	11.30575	99.67380	10000
B	1,49,639	2.00000	155.71176	1922

In the figures 1, 2, and 3, we show average errors against different αs. From figures 1, 2, and 3, we could conclude that AE normally increases as α increases. The number of databases reporting a pattern decreases as α increases. Thus, the AE of synthesizing patterns normally increases as α increases.

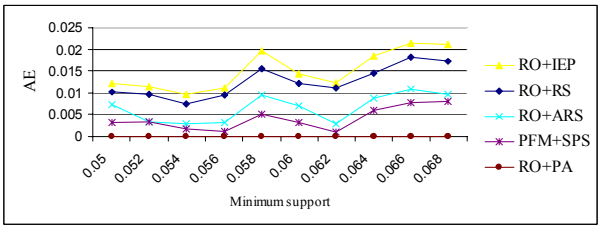

Figure 2. AE vs. α for experiments using dataset T

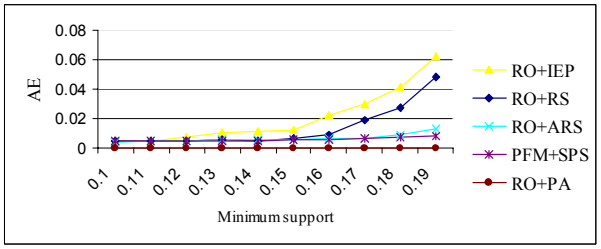

Figure 3. AE vs. α for experiments using dataset R

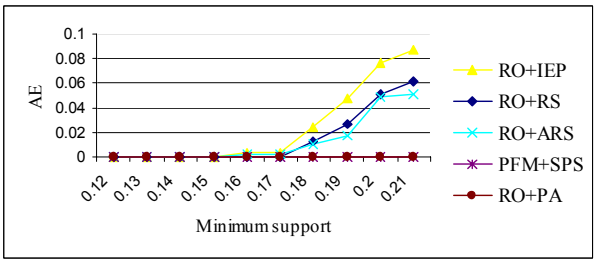

Figure 4. AE vs. α for experiments using dataset B

6. Conclusion

In this paper, we propose a new technique for mining multiple databases. It improves significantly the accuracy of mining multiple databases as compared to existing techniques that scan each database only once. The proposed technique could also be used for mining a large database by dividing it into sub-databases. MDMT: PFM+SPS is effective and promising.

Table 2. Input database characteristics

DB	NT	ALT	AFI	NI	DB	NT	ALT	AFI	NI
T_0	10000	11.05500	127.65588	866	T_5	10000	11.13910	128.62702	866
T_1	10000	11.13330	128.41176	867	T_6	10000	11.10780	128.56250	864
T_2	10000	11.06700	127.64705	867	T_7	10000	11.09840	128.45376	864
T_3	10000	11.12260	128.43649	866	T_8	10000	11.08150	128.55568	862
T_4	10000	11.13670	128.74797	865	T_9	10000	11.08140	128.10867	865
R_0	9000	11.24389	12.070014	8384	R_5	9000	10.85578	16.70977	5847
R_1	9000	11.20922	12.265410	8225	R_6	9000	11.20011	17.41552	5788
R_2	9000	11.33667	14.596567	6990	R_7	9000	11.15511	17.34554	5788
R_3	9000	11.48978	16.662585	6206	R_8	9000	11.99711	18.69032	5777
R_4	9000	10.95678	16.039525	6148	R_9	7162	11.69199	15.34787	5456
B_0	14000	2.00000	14.94130	1874	B_5	14000	2.00000	280.00000	100
B_1	14000	2.00000	280.00000	100	B_6	14000	2.00000	280.00000	100
B_2	14000	2.00000	280.00000	100	B_7	14000	2.00000	280.00000	100
B_3	14000	2.00000	280.00000	100	B_8	14000	2.00000	280.00000	100
B_4	14000	2.00000	280.00000	100	B_9	23639	2.00000	472.78000	100

Table 3. Error of the experiments at given α

Dataset	*T10I4D100K*		*retail*		*BMS-Web-Wiew-1*	
α	0.05		0.11		0.19	
Error type	AE	ME	AE	ME	AE	ME
MDMT: RO+IEP	0.01218	0.03730	0.00516	0.05825	0.04823	0.14490
MDMT: RO+RS	0.01017	0.03612	0.00502	0.05755	0.02319	0.13490
MDMT: RO+ARS	0.00719	0.03599	0.00491	0.05730	0.02102	0.10514
MDMT: PFM+SPS	0.00321	0.03583	0.00484	0.05725	0	0
MDMT: RO+PA	0	0	0	0	0	0

7. Acknowledgement

The first author would like to thank State Government of Goa, India for having sponsored him on faculty improvement program with leave to take up full time research.

8. References

[1] A. Adhikari, and P. R. Rao, "Synthesizing heavy association rules from different real data sources", Under revision towards publication in *Pattern Recognition Letters* since 2005.

[2] R. Agrawal, and J. Shafer, 1999, "Parallel mining of association rules", *IEEE Transactions on Knowledge and Data Engneering*, 8(6), 1999, pp. 962-969.

[3] R. Agrawal, and R. Srikant, "Fast algorithms for mining association rules", *Proceedings of VLDB*, 1994, pp. 487-499.

[4] J. Chattratichat et al., "Large scale data mining: Challenges, and responses", *Proceedings of KDD*, 1997, pp. 143-146.

[5] Frequent Itemset Mining Dataset Repository, http://fimi.cs.helsinki.fi/data/.

[6] J. Han, J. Pei, and Y. Yiwen, "Mining frequent patterns without candidate generation", *Proceedings of SIGMOD*, 2000, pp. 1-12.

[7] M. Last, and A. Kandel, "Automated detection of outliers in real-world data", *Proceedings of the Second International Conference on Intelligent Technologies*, 2001, pp. 292-301.

[8] D. Pyle, *Data Preparation for Data Mining*, Morgan Kufmann, San Francisco, 1999.

[9] A. Savasere, E. Omiecinski, and S. Navathe, "An efficient algorithm for mining association rules in large databases", *Proceedings of VLBD*, 1995, pp. 432-443.

[10] X. Wu, and S. Zhang, "Synthesizing High-Frequency Rules from Different Data Sources", *IEEE Transactions on Knowledge and Data Engineering*, 14(2), 2003, pp. 353-367.

[11] C. Zhang, M. Liu, W. Nie, and S. Zhang, "Identifying global exceptional patterns in multi-database mining", *IEEE Computational Intelligence Bulletin*, 3(1), 2004, pp. 19-24.

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

[13] S. Zhang, X. Wu, and C. Zhang, "Multi-database mining", *IEEE Computational Intelligence Bulletin*, 2(1), 2003, pp. 5-13.

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