A MECHANISM TO DISCOVER AND INTEGRATE ASSOCIATION RULES FROM MULTIPLE SOURCES

Nuntawut Kaoungku, Kittisak Kerdprasop and Nittaya Kerdprasop
School of Computer Engineering
Suranaree University of Technology
111 University Avenue, Nakhon Ratchasima 30000, Thailand

ABSTRACT

The aim of this paper is to study the problem of distributed association mining and to propose a novel method for discovering and integrating association rules from multiple sources. Currently, with the advancement of computer and internet technologies, data are stored in several places such as the medical information gathered by numerous branches and various departments of the hospital. This situation has made association rule mining a difficult task because traditional mining method has been designed for centralized analysis. Therefore, this paper intends to solve the distributed association rule mining problem by proposing a method for analyzing pieces of information at the local sites and then integrate the induced association rules via the reasoning mechanism in its natural language form. Experimental results confirm the efficacy of our proposed method.

KEYWORDS: Distributed data mining, Association rule analysis, Logical inference, Natural language

1. Introduction

With the current advancement of ubiquitous computing, electronic data are enormously generated and stored at the local sites of most organizations. Due to the tremendous amount of data and information that are constantly increasing every day, current technology of intelligent data analysis such as data mining is facing scalability trouble of its learning algorithm.

Data mining is a technique for data analysis that has been designed to extract knowledge from the database content being gathered and stored at a single source. Stored data in their raw format can be analyzed base on the statistics and machine learning

techniques to find the trend, common data characteristics, association or affiliation among data attributes, and so on. But with the present characteristic of the data that are large in their size and distributive in their location, data mining methodology has to be adjusted and extended to deal with such problem.

Many researches have proposed the concept of distributed data mining [1], which is a commonsense solution to extract knowledge from data that are stored in several places to get a single knowledge base. This concept can also be applied to the problem of the inability to extract knowledge from the single data source, but the amount of data is too large to be processed in a centralized manner. This decentralized concept can be applied to many data mining tasks such as classification, clustering, association analysis, and others. In this research, we focus on the association mining task.

Association rule mining is the popular data mining algorithm to discover the relationship of associated items in the database. The discovered association can be of great beneficial to the organization, but the discovering process is computationally expensive in terms of computing resources such as CPU time and memory. A decentralized solution to this problem is to divide the data into many processes during the association rule mining. We sketch the idea of distributed versus centralized data mining and graphically shown in Figure 1. Our research work presented in this paper is in the mainstream of distributed data mining. The main focus of our research is on the knowledge consolidation part. We propose to perform an association mining separately at each local site. The learned knowledge in a format of association rules from each site is then integrated through the natural language reasoning mechanism. The missing knowledge can also be inferred through the ontology technology.

The contributions of this paper are as follows:

- With the proposed mechanism, association rule mining can be performed over distributed and big data.
- We introduce the transformation of general association rules to the controlled natural language association rules representing knowledge that are suitable for building ontologies.
- The proposed mechanism can be used to infer new association rules from ontologies. The inferred rules are used to fill the missing association.

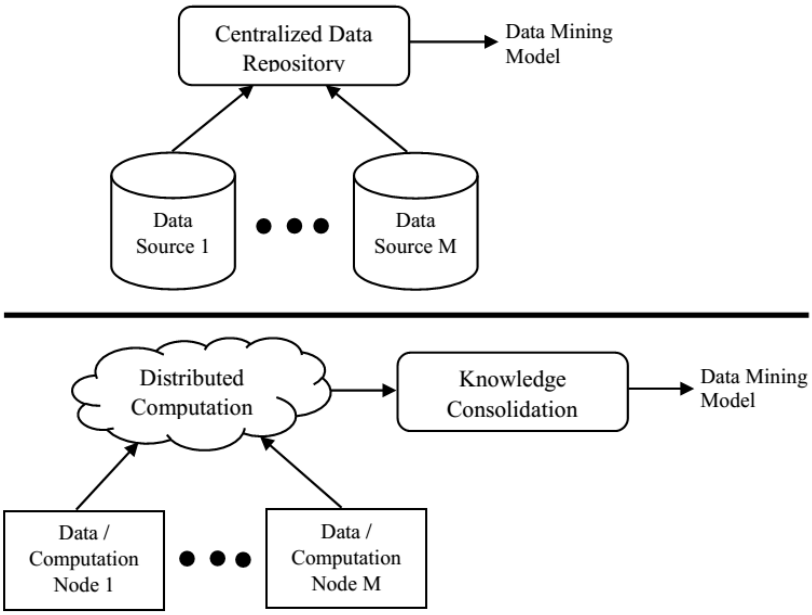

Figure 1 The centralized data mining (above) versus distributed data mining (below)

2. Related Work.

Agrawal et al. proposed Apriori algorithm in 1993 [2] for the association rule mining task. The main idea is a find frequently occurred patterns from the database, which are item values that often co-occurring in the transaction records. These frequent item patterns are extracted and represented as antecedent-consequent rules to be used in the association analysis task. Due to the fact that current data are very large, Apriori algorithm thus requires much time for the search of frequent patterns. The search process for candidate frequent itemset has a good structure suitable for parallel association rule mining. Therefore, many research teams [3 - 5] have studied and proposed the parallel strategies for association rule mining algorithm.

In 1996, Agrawal et al. proposed algorithm for parallel association rule mining based on count distribution and data distribution [6, 7] to be applied in the frequent-pattern mining phase. This algorithm divides data into blocks and then sends the block to each processor for the computation of frequent itemsets. But this algorithm consumes a large amount of memory. An Amount of processors and communication also increases in proportion to the number of data distributions.

Cheung et al. also proposed in 1996 the fast distributed mining (FDM) algorithm [8, 9] based on the classical association rule mining. FDM reduces the number of candidate itemsets by pruning at each site the itemsets that are not locally frequent. This research team proposed the extension of FDM in 2002 and called the extended one as fast parallel mining (FPM) algorithm [10]. FPM can reduce the amount of communication and speed-up the overall association rule mining process.

Yu et al. proposed in 2010 the load-balanced distributed parallel mining algorithm [11]. They proposed that to perform frequent pattern mining on each processor is redundant. It causes the processor not fully functional. Therefore, they suggested the distributed management in each processor to speed-up the association rule mining.

From the related work, it can be seen that there exist many techniques for solving the distributed association rule mining problem. However, these paradigms still require a large amount of memory and communication. We thus propose an economical mechanism to discover and integrate association rules from multiple sources.

3. Preliminaries.

3.1 Association Rule Mining.

Association rule mining is the search for the relationship of the event or frequent pattern that has the potential to be applied in the analysis or predicting the future events. This relationship is in the form *IF condition Then result* [2]. To limit the search space, the discovered relationship has to satisfy these two criteria:

- Support. It is the frequency of the occurring event and can be computed as the probability that two events (A and B) occur in the same transaction (equation 1). The minimum threshold of support value is normally specified by users.

$$Support(A \rightarrow B) = P(A \wedge B) \tag{1}$$

- Confidence. It is the proportion of frequency of co-occurring events (A and B) to the frequency of the antecedent event (A). The computation is in equation 2. The minimum confidence is the threshold used to screen only interesting relationships.

$$Confidence(A \rightarrow B) = \frac{Support(A \rightarrow B)}{Support(A)} \quad (2)$$

3.2 Attempto Controlled English.

Attempto Controlled English (ACE) is a controlled natural language, which is written in the form of English sentences to serve as knowledge representation language. ACE has been designed as the writing language in the form that both human and machine can understand [12]. The examples of ACE simple sentences are as the following:

Every pets is an animal.
Dog is a pet.
Cat is a pet.

This research uses ACE as a tool to transform general association rules to the format of natural language for the purpose of checking inconsistency of the induced relationships obtained from distributed sites. We check knowledge consistency with the FaCT++ reasoner that is available as a plug-in in the Protégé editor.

3.3 Logical Inference.

Logical inference refers to the action or process of inference on knowledge [13] that has not been stated clearly or not directly specified. Logical inference can be divided into two types: inductive inference and deductive inference. The logical inference is applied in many fields such as the expert system and semantic Web. The statements to be reasoned are written as follows:

$$H1 \wedge H2 \wedge \ldots \wedge H_n \rightarrow C$$

where

$H1, H2, \ldots, H_n$ are hypothesis,
C is a conclusion.

The example of simple logical inference statements can be listed as the following:

P → *Q: All men are mortal.*

Q → *R: Socrates is a man.*

P → *R: Therefore, Socrates is mortal.*

FaCT++ is a reasoner (or reasoning mechanism) that has developed from the FaCT algorithm using C++ language. This reasoner is based on the description logics, which is a powerful first-order language to be used for checking the inconsistency of ontology [14]. An example of the reasoning process is as follows.

1. If X is a man then X is a human.

2. If X is a John then X is a man.

3. If X is a John then X is not a human

When only the first two sentences have been applied to the FaCT++ reasoner, the reasoner can entail the new knowledge as "John is a human." But if all three sentences are given as input to the FaCT++ reasoned, the inconsistency in the sentence 3 will be detected because it contradicts to the sentences 1 and 2.

4. Proposed Method and Algorithm.

Mining association rules from big data or distributed data is a difficult task because the rules combined from distributed processors may be inconsistent. Moreover, some association rules may be missing when compared to association rules that are mind from centralized association rule mining process. Missing rules may happen from the situation that distributed data having different distribution from the centralized data and thus make them unsatisfying the minimum support and confidence requirements. Therefore, this research proposes a mechanism to discover missing rules and to solve inconsistency problem from the integration of association rules from multiple sources. This concept can be easily applied to the traditional association rule mining algorithm and its efficiency in discovering a complete set of association rules is very close to the traditional centralized association rule mining.

Figure 2 shows conceptual framework of this research. In step 1, the data that are stored in different places can be independently mined for association rules. In step 2, we combine rules that occur in every site. Step 3 is the pre-processing step for inconsistency detection. In this step, association rules have to be transformed to controlled natural language (CNL) with Attempto Controlled English (ACE). Step 4 is the check for inconsistency with the FaCT++ reasoner of Protégé editor. Some association from the FaCT++ reasoner can be used to fill missing rules in step 2.

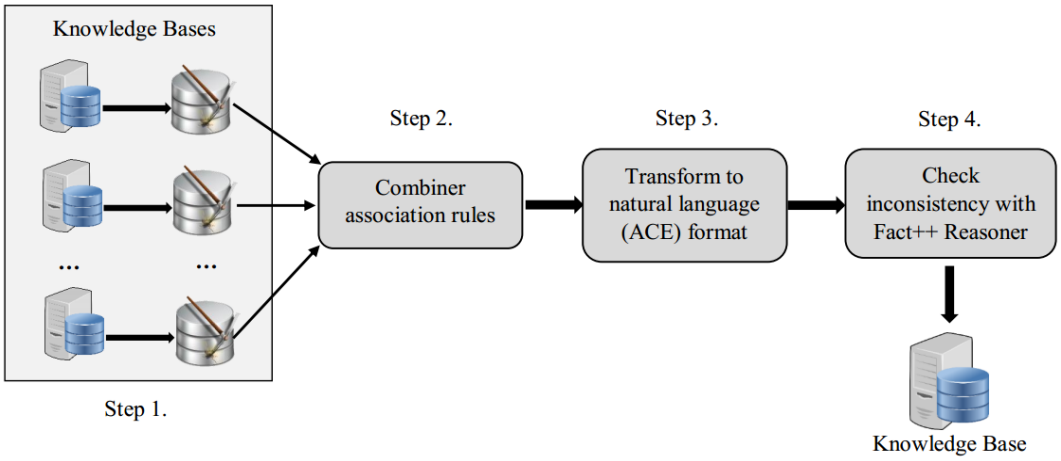

Figure 2 Conceptual framework of the association rule discovery and integration

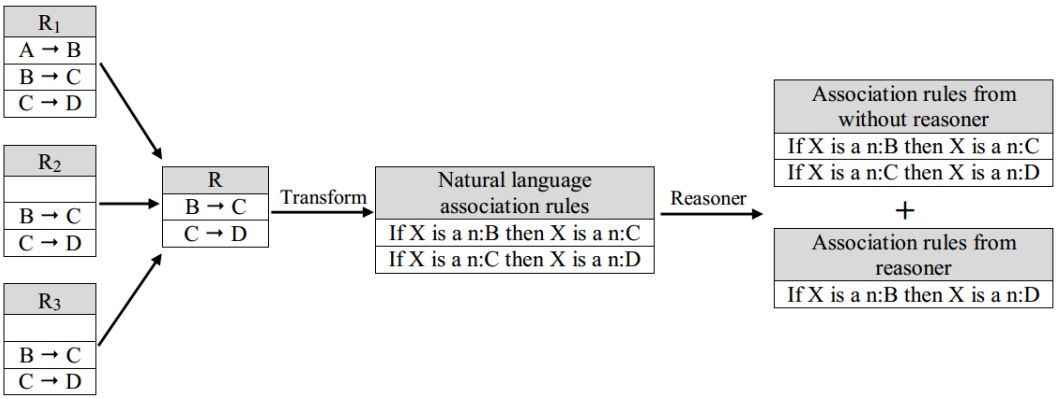

Figure 3 Running example of association integration

Figure 3 shows the running example in the reasoning and rule integration approach. Suppose there are 3 sets of association rules (R1, R2, and R3) that are independently

discovered at their local sites. After joining the rules that are commonly occurred in every site, we obtain the final two rules: “B->C” and “C->D”. After that transform the rules “B->C” and “C->D” to natural language, the result is:

“if X is a n:B then X is a n:C”
“if X is a n:C then X is a n:D”

The prefix “n:” is used to specify part of speech as a noun. Finally, these two rules in the natural language format have been applied to Fact++ reasoned to check for inconsistency and also to infer new rule. These rules are consistent. From the inference mechanism, we obtain new association rule, which is:

“if X is a n:B then X is a n:D”

The algorithm for rule discovery and integration is given in Figure 4. Our algorithm consists of three phases. The first phase is from line 1 to 5. This phase is for extracting common rules from all sources: $C = R_1 \cap R_2 \cap ... \cap R_i$ with $i = 1,2,3,...,n$.

Phase 2 is from line 6 to 16. This phase is the transformation from the general association rules normally represented as IF-THEN rules to be rules in the natural language format. The transformation is based on the simple technique to find and replace strings or characters. For instance, general association rule ‘{A=B} => {B=C}’ will be transformed into natural language as ‘if X is a n: A_equal_B then X is a n: B_equal_C’.

Phase 3 is from line 17 to 25. Line 18 is for the ontology building using ACE tool in the Protégé editor. Lines 19-24 check inconsistency and infer new association rules by using reasoner (FaCT++ reasoner). If there exist inconsistent rules, they will be removed. Finally, the algorithm returns the new association rules from reasoner and the possibly inferred rules that can be used to impute the missing association rules in the first phase.

Algorithm 1 Discover and Integrate Association Rules from Multiple Sources

//Input: $\{R_1, R_2, \ldots, R_N\}$, association rules in all nodes.

//Output: I, Inconsistency of Association Rules as true or false.

E, **Association rules entailed from reasoner**.

```
1.  Create C as empty list;
2.  Create CACE as empty list;
3.  for i =1 ← to N do
4.      C = intersection (Ri)
5.  end for
6.  Create D as dictionary = {      'A{' :'if X is a n:'
7.                          '=>' : 'then X is a',
8.                          '=' : '_equal_',
9.                          ',' : ' and X is a n:',
10.                         '{' : 'n:',
11.                         '}' : '',
12. }
13. for i =1 ← to length(C) do
14.     RN = multiple_replace(D, Ci);
15.     add RN to CACE;
16. end for
17. I = true
18. Ontology = ACE_views (CACE)
19. while I = true{
20.     (I, E) = Reasoner (Ontology)
21.     if I = true
22.         Ontology = Remove_rules_inconsistent (CACE)
23.     end if
24. end while
25. Return (I, E)
```

Figure 4 Algorithm to discover and integrate association rules from multiple sources

5. Experimental Results.

5.1 Real Application to Simple Data.

The proposed mechanism to discover and integrate association rules from multiple sources has experimented with real data from the UCI machine learning repository. The UCI data are Breast-Cancer with 286 records and 9 attributes. To simulate the distributive scenario, we divide this dataset into 3 subsets, that is, Data1, Data2, and Data3. Each data subset has been mined for association rules using the minimum support value ranging from 0.1, 0.2, 0.3, 0.4, 0.5, to 0.6. We compare the performance of mining from multiple sources against the centralized one based on the number of rules. We also compare the rule integration with and without reasoning mechanism.

Table 1 shows the comparative results of centralized association rule mining and association rule mining on multiple sources using small dataset (Breast-Cancer data). From the table, we notice that the rule discovery and integration without reasoning mechanism discover a much smaller set of association rules than those obtained from the centralized method. When applying reasoning mechanism, the improvement in terms of number of the association rules is noticeable. We can observe from this result that our proposed method gives a satisfying result when minimum support has been set at 0.4 or higher.

Table 1 Comparative results of centralized association rule mining versus association rule mining on multiple sources using Breast-Cancer data.

Support	# Association rules from centralized method	# Association rules from multiple sources without reasoner	# Association rules from multiple sources with reasoner	% Improvement by reasoner
0.1	883	372	380	2.15%
0.2	203	119	126	5.88%
0.3	62	30	35	16.67%
0.4	24	19	24	26.32%
0.5	24	10	15	50.00%
0.6	9	6	9	50.00%

Table 2 Comparative results of centralized association rule mining versus association rule mining on multiple sources using big data.

Support	# Association rules from centralized method	# Association rules from multiple sources without reasoner	# Association rules from multiple sources with reasoner	% Improvement by reasoner
1.0	-	-	-	-
0.9	16,031	15,468	15,498	0.19%
0.8	1,560,469	1,510,653	-	-

5.2 Real Application to Big Data.

The big data taken from the UCI are US Census Data (1990) with 2,458,285 records and 68 attributes. We divided this big data into 10 data subsets and set the minimum support to be 0.8, 0.9, and 1.0.

Table 2 shows the comparative results of centralized association rule mining and our method of association rule mining on multiple sources. From the table, we can see that the number of association rules from reasoner with minimum support 0.8 is null because ACE has a limited number of rules to represent knowledge. But the number of association rules from multiple sources without reasoned is close to the number of centralized association rules.

6. Conclusions.

Data are extremely easy to be generated from the current electronic machines and be kept in various kinds of storage devices as modern technology are being developed continuously. When data are big and scatter to several places, it becomes a difficult task to do data analysis such as association rule mining. Current research cannot handle this extreme situation. We thus propose in this paper a mechanism to integrate association rules that have been analyzed in distribute places. We design a framework to induce association rules at local sites. Then transfer the learned knowledge to the central one for the integration and reasoning process. The proposed reasoning mechanism is for rule inconsistency detection. With the power of first-order logic, this mechanism can also infer new rules from

the existing ones. The new rules can be used as an imputation for the missing ones during the distributed learning of association rules.

From the experimental results, we observe the power of reasoning mechanism that it can improve the discovery of association rules that are induced from small pieces of information. The proposed method tends to favor distributes association rule mining at a high minimum support value. This tendency is suitable for the association rule learning from big data. We, therefore, plan to extend our method toward the association rule learning from big data.

Acknowledgement

The first author has been funded by the scholarship from Suranaree University of Technology. This research has been funded by grants from the National Research Council of Thailand and Suranaree University of Technology through the funding of Knowledge Engineering Research Unit (headed by the second author) and Data Engineering Research Unit (headed by the third author).

References

[1] G. Tsoumakas and I. Vlahavas. (2009). “Distributed Data Mining”. **Encyclopedia of Data Warehousing and Mining.** : 157-164.

[2] R. Agawal, T. Imilinski, and A. Swami. (1993). “Mining association rules between sets of items in large databases”. **ACM SIGMOD Record.** Vol.22 (2): 207-216.

[3] S. Einakian and M. Ghanbari. (2006). “Parallel implementation of association rules in data mining”. **In Proceedings of the 38th southeastern symposium on system theory.** : 21-26.

[4] S. Parthasarathy, M. J. Zaki, M. Ogihara, and W. Li. (2001). “Parallel data mining for association rules on shared-memory systems”. **Knowledge and Information Systems.** Vol.3 (1): 1-29.

[5] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. (1997). “Parallel algorithms for discovery of association rules”. **Data Mining and Knowledge Discovery.** Vol.1 (4): 343-373.

[6] R. Agrawal and R. Srikant. (1994). “Fast algorithms for mining association rules”. **In**

Proceedings of the 20th international conference on very large databases. : 487-499.

[7] R. Agrawal and J. C. Shafer. (1996). “Parallel mining of association rules”. **IEEE Transactions on Knowledge and Data Engineering.** Vol.8 (6): 962-969.

[8] D. W. Cheung, J. Han, V. T. Ng, A. W. Fu, and Y. Fu. (1996). “A fast distributed algorithm for mining association rules”. **In The fourth international conference on parallel and distributed information systems.** : 31-42.

[9] D. W. Cheung, V. T. Ng, and A. W. Fu. (1996). “Efficient mining of association rules in distributed databases”. **IEEE Transactions on Knowledge and Data Engineering.** Vol.8 (6): 911-922.

[10] D. W. Cheung, S. D. Lee, and Y. Xiao. (2002). “Effect of data skewness and workload balance in parallel data mining”. **IEEE Transactions on Knowledge and Data Engineering.** Vol.14 (3): 498-514.

[11] K. M. Yu, J. Zhou, T. P. Hong, and J. L. Zhou. (2010). “A load-balanced distributed parallel mining algorithm”. **Expert Systems with Applications.** Vol.37 (3): 2459-2464.

[12] N. E. Fuchs, and K. Kaljurand. (2006). “Attempto Controlled English: Language” **Tools and Applications.** Lecture/Presentation.

[13] A. Fuhrmann. (1998). “Nonmonotonic Logic”. **Routledge Encyclopedia of Philosophy**. Vol.7: 30-35.

[14] D. Tsarkov and I. Horrocks. (2006). “FaCT++ description logic reasoned: System description. Automated reasoning”. **In Springer Berlin Heidelberg**. : 292-297.

Nuntawut Kaoungku is currently a doctoral student with the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2012, and master degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2013 and doctoral degree in Computer Engineering from Suranaree University of Technology, Thailand, in 2014. His current research includes semantic web and association.

Kittisak Kerdprasop is an associate professor and chair of the School of Computer Engineering, Suranaree University of Technology, Thailand. He received his bachelor degree in Mathematics from Srinakarinwirot University, Thailand, in 1986, master degree in Computer Science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in Computer Science from Nova Southeastern University, U.S.A., in 1999. His current research includes Data mining, Artificial Intelligence, Functional and Logic Programming Languages, Computational Statistics.

Nittaya Kerdprasop is an associate professor at the School of Computer Engineering, Suranaree University of Technology, Thailand. She received her bachelor degree in Radiation Techniques from Mahidol University, Thailand, in 1985, master degree in Computer Science from the Prince of Songkla University, Thailand, in 1991 and doctoral degree in Computer Science from Nova Southeastern University, U.S.A, in 1999. She is a member of ACM and IEEE Computer Society. Her research of interest includes Knowledge Discovery in Databases, Artificial Intelligence, Logic Programming, and Intelligent Databases.