Association-rule Based Information Source Selection[+]

Hui Yang[1], Minjie Zhang[1], and Zhongzhi Shi[2]

[1] School of IT and Computer Science, University of Wollongong, Wollongong, Australia
{hy92, minjie}@uow.edu.au
[2]Institute of Computing Technology, Chinese Academy of Science, Beijing, P.R.China
shizz@ics.ict.ac.cn

Abstract: The proliferation of information sources available on the Wide World Web has resulted in a need for database selection tools to locate the potential useful information sources with respect to the user's information need. Current database selection tools always treat each database independently, ignoring the implicit, useful associations between distributed databases. To overcome this shortcoming, in this paper, we introduce a data-mining approach to assist the process of database selection by extracting potential interesting association rules between web databases from a collection of previous selection results. With a topic hierarchy, we exploit intraclass and interclass associations between distributed databases, and use the discovered knowledge on distributed databases to refine the original selection results. We present experimental results to demonstrate that this technique is useful in improving the effectiveness of database selection.

1 Introduction

With the explosive growth of information sources available on the Wide World Web, the web has become an enormous, distributed, and heterogeneous information space. To effectively and efficiently find interesting information from the huge amount of resource existing on the web, one of the important steps is to firstly select a subset of distributed collections which are most likely to contain relevant documents regarding the user query before extracting useful information in individual information sources. As a result, **information source selection** (or **resource discovery**) problem is becoming an increasingly important research issue in the distributed information retrieval (DIR) area [4].

Nowadays, several database selection approaches have been introduced to help select the most relevant information sources from the Wide World Web and have received encouraging achievements [2, 3]. Unfortunately, these methods always treat each database independently, ignoring the implicit, useful associations between distributed databases. But the discovery and analysis of the useful information about the relations between the databases will be beneficial to the performance improvement of database selection.

Data mining, an important part of knowledge discovery, is concerned with the process of extracting implicit, previously unknown, and potentially useful information from given data. Algorithms for data mining focus on the discovery of relevant and interesting patterns within large amounts of data. To the best of our knowledge, very little work has been done on the mining of association rules between distributed databases used for database selection in distributed

[+] The research was supported by URC Small Grand-227381019 of Wollongong University

information retrieval (DIR). The work in this paper could be viewed as a step towards combining data mining techniques with the problem of database selection.

Given a collection of previous database-selection results, a data-mining algorithm is developed to discover knowledge on the databases by extracting potential relations between distributed databases with the use of a topic hierarchy. The discovered knowledge provides assistance in refine the relatively rough original results that are obtained from the database selection tools. Here, what is needed to emphasize is that the aim of this paper is not intended to propose an alternative database selection approach, but provide a subsidiary means to refine the final results on the basis of original selection results with the discovered associations between the databases so as to improve the effectiveness of database selection. Therefore, this association-rule approach can be regarded as a step towards the post-processing of database selection. The contributions of this paper are summarized as follows:

(1) A new methodology for the problem of database selection is proposed from the viewpoint of data mining.
(2) In consideration of the diversity of topic contents of distributed web databases, a topic-based association-rule mining process is accomplished by a twofold approach: first, to generate associations between the databases within the same topic class (i.e., intraclass); and then to deduce the association rules between relevant topics (i.e., interclass) such as parent-child classes and sibling classes in the hierarchical structure.

The remainder of this paper is organized as follows: in the next section, we firstly introduce some background knowledge of our current work. In Section 3, the details of discovering association rules between distributed databases using hierarchical topics are given. In the same section, we also discuss how association rule technique can be applied to the database-selection process. Experimental setup and experimental methodologies are given in Section 4. In Section 5, the performance study of the proposed approach is performed on the Reuters-21578 data set, and the results of the experiments are analyzed. Finally, Section 6 concludes the paper with a brief discussion of future work.

2 Background Knowledge of A Topic-based Database Selection

This paper is, in fact, the extension of our previous work on database selection. In [6], we proposed a topic-based database selection approach. We firstly partition multiple, distributed web databases based on their subject contents into a structured hierarchy of topics using a Bayesian network learning algorithm. Given a user query, the task of database selection is decomposed into two distinct stages: First, at the category-specific search stage, the system identifies one or more specific topic domains that best match the user's information need. Second, at the term-specific search stage, the selection system computes the likelihood of the databases associated with the relevant topics chosen at the first stage, and selects the potential most appropriate databases based on the ranking scores of the likelihood.

Since our topic-based database-selection approach is based on a topic hierarchy, we consider integrating the topic hierarchy with the discovery of associations between distributed databases. The main reason for using the topic hierarchy has two aspects: one is *the efficiency of data mining*. For a large collection of previous database-selection results, the use of a topic hierarchy can decompose the data mining task into a set of smaller subtasks, each of which only corresponds to the mining of a focused topic domain in the hierarchical tree, therefore making the accomplishment of the data-mining work more effective and efficient; he other is *the search of relevant association rules*. Given a user query, once the database-selection result is returned, the association rules associated with the specific topics that the user is interested in will be directly used for the refinement of the original selection result. As a result, the expense and time of the search for relevant association rules in the association space will be much reduced.

In our work, we utilize a topic hierarchy to assist in the discovery of association rules between databases. Figure 1 shows an example of a simple topic hierarchy. Let *C* be a classification hierarchy on the topics, which organizes the relationships between the topics in a tree form. Obviously, the relationships between different topics appearing in the structured hierarchy can be classified into three major types of relationships: *parent-child*, *ancestor-descendant*, and *sibling*. For example, in Figure 1, topic "*software*" is the *parent* of topic "*database*", which is semantically more general and broader than topic "*database*". Similarly, topic "*computers*" is the *ancestor* of topic "*database*". Topic "*programming*" is one of the children of topic "*software*", which is defined as a *sibling* to topic "*database*". For each topic in a hierarchical structure, the system stores the knowledge of relationships between this topic and other topics in the hierarchy including parent-child, ancestor-descendent, and sibling relations.

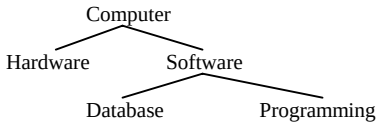

Fig.1 A simple example of the topic hierarchical structure

3 The Discovery of Association Rules between Web Databases Using a Topic Hierarchy

3.1 Association Rules

Formally, as defined in [1], let $I = \{i_1, i_2, \cdots, i_m\}$ be a set of binary attributes called items. Let *D* be a collection of transactions, where each transaction $T \in D$ is a set of items such that $T \subseteq I$ and it is given with a unique identification *TID*. Given an itemset $X \subseteq I$, a transaction *T* is said to contain *X* if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X, Y \subseteq I$ and $X \cap Y = \varnothing$; and *X* is called the *antecedent* of the rule and *Y* is called the *consequence* of the rule. The association rule $X \overset{s,c}{\Rightarrow} Y$ holds in the transaction set *D* with *support s* if *s*% of transactions in *D* contain *X* and *Y*, and *confidence c* if *c*% of transactions in *D* that contain *X* also contain *Y*:

Given a minimum support threshold called *minsup* and a minimum confidence threshold called *minconf*, the problem of discovering association rules from the transaction set *D* is to generate all association rules that have support and confidence greater than the specified minimum support *minsup*, and minimum confidence *minconf*, respectively. These rules are called *strong rules*.

In general, the problem of association rule mining can be decomposed into two subproblems: first, find all the itemsets that have support above the specific minimum support, *minsup*. These itemsets are called *large itemsets* or *frequent itemsets*; second, generate the association rules from the above large itemsets that have found. Given a large itemset with size *k*, a *k*-itemset ζ, $\zeta = i_1 i_2 \cdots i_k$, $k \geq 2$, the antecedent of the rule will be a subset *X* of ζ such that *X* has *j* ($1 \leq j < k$) items, and the consequence *Y* will be the itemset $\zeta - X$, if support (ζ) /support (*X*) >*minconf*, then the rule $X \Rightarrow Y$ is a valid rule.

3.2 A Formal Model of Association Transactions for Database Selection

In the context of distributed information retrieval (DIR), we need to adapt data mining techniques to database selection. The first issue to deal with is to develop a formal model of association transactions for database selection.

Definition 1: A ***query*** transaction *T* in a *topic-based* DIR system is a 4-tuple *T*=<*Q*, *C*, *S*, *D*> where

- *Q* is a user query that can be denoted as $Q = \{q_1, q_2, \cdots, q_N\}$, where q_i ($1 \le i \le N$) is a query term (word) occurring in the query *Q*.
- *C* is a set of appropriate topics with respect to the query, which can be denoted as $C = \{c_1, c_2, \cdots, c_M\}$, where c_i ($1 \le i \le M$) is one topic in the topic hierarchy.
- *S* is a selected database set in the process of database selection. The database set *S* can be described as $S = \{s_1, s_2, \cdots, s_K\}$, where s_j ($1 \le j \le K$) is a web database which is likely to contain relevant information for the user query.
- *D* is a set of retrieved documents that come from the selected database set *S* and satisfy the user query. Document set *D* can be defined as $D = \{d_{11}, \cdots, d_{ij}, \cdots, d_{LK}\}$, where *L* is the total number of retrieved documents; and *K* is the number of the databases in the database set *S*; d_{ij} ($1 \le i \le L$, $1 \le j \le K$) represents the *i*th web document which comes from database s_j in the database set *S*.

With the database set *S* and the topic categories *C* in the query transaction, we construct a topic-based database-selection transaction that represents a database-selection result. Unfortunately, this type of database-selection transactions focuses on the binary attribute of the database items, which is only concerned with whether a database appears in a transaction or not, but does not take the relevance degree of a database to the user query into account. For example, given a user query, the DIR system returns a query result of 7 relevant web documents. Among them, 5 documents come from database s_1, and 2 documents comes from database s_2. The database-selection transaction can only reflect the fact that the databases s_1 and s_2 are selected as the relevant databases to the query, which leads to the loss of important information about different relevance degrees of individual databases to the user query. To let the database-selection transactions express the information about the relevance degree of databases to the query, with fuzzy set theory, we extend the traditional association rule by assigning a weight to each database item in the transaction, to indicate the relevance (importance) degree of such a database.

Definition 2: A topic-based ***database-selection*** transaction τ is a 2-tuple τ=<*C*, *S*> where *C* is the same as Definition 1; and *S* is a set of weighted databases searched by the DIR system, which can be described as $S = \{< s_1, w_1 >, < s_2, w_2 >, \cdots, < s_K, w_K >\}$, where a pair $< s_j, w_j >$ is called a ***weighted database item*** s_j^w ($1 \le j \le K$), and s_j is a *database* item and w_j is a weight associated with *database* item s_j.

Obviously, a topic-based *database-selection* transaction *T* is the combination of topic items and weighted database items. A simple example of a database selection transaction is show as follows:

$$\text{Transaction } T_1\text{: } T_1 = \{< c_1, c_2 >, < s_1, w_1 >, < s_2, w_2 >\}.$$

Here, we use fuzzy set concept to express the relevance (importance) degree of each database in database set *S* to the user query. A fuzzy set is defined as a collection of elements with the associated membership value between 0 (complete exclusion) and 1 (complete inclusion).

The membership value represents the degree of membership of an element in a given set [5]. A fuzzy set *A* in the database set *S* is defined as a set of ordered pairs:

$$A = \{(s_j, u_A(s_j)) \mid s_j \in S\} \quad (1)$$

where $u_A(s_j)$ is called the membership function. The membership function maps each database s_j in database set *S* to a membership grade between 0 and 1. The membership function $u_A(s_j)$ can be described as

$$w_j = u_A(s_j) = \frac{\sum_i d_{ij}}{\sum_i \sum_t d_{it}} \quad (2)$$

where d_{it} ($1 \le i \le L$, $1 \le t \le K$) represents the *i*th retrieved document which appears in database s_t in the database set *S* (recall Definition 1). $\sum_i d_{ij}$ denotes the number of the documents retrieved from database s_j. w_j, the weight associated with database s_j, is assigned by the membership function $u_A(s_j)$, and $\sum_j w_j = 1$ ($1 \le j \le K$).

3.3 The Discovery of Fuzzy Association Rule with A Topic Hierarchy

In this subsection, we will first give the definition of fuzzy association rule (FAR). Then we will discuss the issues and problems in the mining of intraclass association rules and interclass association rules, respectively.

3.3.1 Fuzzy Association Rule

We use the term *weighted database itemset* to represent a set of weighted database items with set membership value [0,1] in the database-selection transactions.

Definition 3: A ***weighted database k-itemset*** δ in a transaction is a set of weighted database items, $\delta = \{s_1^w, s_2^w, \cdots, s_k^w\}$, where s_i^w ($1 \le i \le k$) is a *weighted database item* (recall Definition 2).

Definition 4: *Item*() is a database function which extracts the database set from a *weighted database itemset* δ.

For example, given a *weighted database k-itemset* δ, $Item(\delta) = \{s_1, s_2, \cdots, s_K\}$, where s_i ($1 \le i \le k$) is a *database* item in the itemset δ.

Definition 5: Given a set of transactions *T*, an interesting ***fuzzy association rule*** (FAR) is defined as an implication of the form $X \overset{s,c,r}{\Rightarrow} Y$, where *X* and *Y* are two *weighted database itemsets*, and $item(X) \cap item(Y) = \varnothing$. We said that the fuzzy association rule holds in the transaction set *T* with *support* s if *s*% of transactions in *T* contain *item*(*X*) and *item*(*Y*), *confidence c* if *c*% of transactions in *T* that contain *item*(*X*) also contain *item*(*Y*), and *relevance* $r \in [0,1]$ if the weight of each item in the itemsets, *item*(*X*) and *item*(*Y*), is greater than the relevance threshold *r*.

Here, the *relevance* concept is introduced to develop effective pruning techniques to identify potentially important database items for the fuzzy association rule mining. To efficiently discover the interesting rules, we push *relevance* constraint in the candidate itemset generating phase of the association rule mining algorithm in order to only retain the suitable candidate

itemsets which have the database items with higher weight in the transactions, hence discarding those trivial ones with low weight. This pruning saves both the memory for storing large itemsets and mining efforts. Intuitively, *relevance* parameter can be viewed as an indicator of the required relevance (importance) degree of each item in the *large weighted database itemsets* to a specific topic.

In sum, given a transaction set *T*, our objective is to discover a set of fuzzy association rules which have *support*, *confidence* and *relevance* satisfying the specific minimums, *minsup*, *minconf* and *minrele*.

3.3.2 The Discovery of Intraclass Association Rules

As previously mentioned, the connections among the databases in the context of a topic hierarchy can be grouped into two major types of association rules: one is *intraclass association rules* within the same topic class, the other is *interclass association rules* between relevant topic classes. Now, we first will discuss how to mine intraclass association rules between the databases on a specific topic. Here, we are only interested in a subset of transactions which are labeled with the specific topic considered.

Definition 6: An interesting ***intraclass association rule*** is described as $X \overset{s,c,r}{\Rightarrow} Y \mid C = c_i$, where c_i is the specific topic considered; and the parameters *X*, *Y*, *s*, *c*, *r* are the same as Definition 5.

We present an Aprior-like algorithm to perform the generation of an intraclass association rule. The three major mining steps are described as follows:

(1) Generate all *large database itemsets* which have *support* greater than the specific minimum support *minsup*. For a database itemset ζ, if in the transaction set, the fraction of transactions containing the itemset ζ is greater than *minsup s*, we call ζ a *large database itemset*.

(2) For each of the above large database itemsets, the weight w_i of each database item s_i in a *large database itemset* ζ is calculated by first summing the weights of item s_i in all the transactions containing the itemset δ, and then dividing it by the total number of the transactions containing the itemset ζ, which is defined as

$$w_i = \frac{\text{Sum of the weights of item } s_i \text{ in all the transactions containing the itemset } \delta}{\text{the total number of all the transactions containing the itemset } \delta} \quad (3)$$

If the weights of all the database items in the itemset ζ are all greater than specified minimum relevance *minrele r*, the itemset ζ is called a *large weighted database itemset*.

(3) Once all the *large weight database itemsets* are found, the potentially interesting association rules can be derived from the large itemsets in a straightforward manner. For each *large weight database itemset*, all association rules that have greater than the specified minimum confidence *miniconf* will be derived. For example, for a *large weighted database itemset* ζ, and any *X* ($X \subset \zeta$), if *support* (*item*(ζ)) / *support* (*item* (ζ) –*item* (*X*)) > *minconf*, the rule $X \Rightarrow (\zeta - X)$ will be derived.

It is important to note that for each intraclass association rule, it in fact contains two types of information: one is the information on the coourence between the databases, and the other is the information on different relevance degree of individual databases to the specific topic considered. For example, there is an intraclass association rule, that is, *Rule A*: $\{< s_1, 0.4>, < s_2, 0.2>\} \Rightarrow \{< s_3, 0.1>\} | C =$ "*software*", which indicates that for topic domain "*software*", if the databases s_1, s_2 are chosen by a database-selection tool, then it is likely that database s_3 will also be selected; on the other hand, it implies that the content of database s_1 is more relevant to

topic "*software*" than that of the databases s_2 and s_3, since its potential relevance weight is 0.4, the biggest one among the three databases.

Intraclass association rules can be used to improve the performance of database selection. Consider such a scenario that assumes that a user is searching the information of topic "*software*" on the Internet. The *original* database-selection result by a database-selection tool is the databases s_1 and s_2 which are considered to contain the documents of interest. With *Rule A*, we can add database s_3 into the extended search space, because since the databases s_1 and s_2 have been chosen, and according to *Rule A*, database s_3 will be selected as a potentially useful database with respect to topic "*software*". At the same time, among these three databases, we will rank database s_1 ahead of the databases s_2 and s_3 in the *final* result since database s_1 is more important than other two databases according to *Rule A*.

3.3.3 The Discovery of Interclass Association Rules

As described earlier, a *database-selection* transaction is probably labeled with multiple topics. It is necessary to identify the correlations among the databases in the context of the closely-related topics. In order to simplify the explanation, our work will be introduced based on the assumption that there are a pair of related topics in the topic hierarchy, which will be easily extended to any number of related topics in the hierarchy.

Now we firstly introduce the notion of *overlap* factor. The *overlap* factor is the ratio of the transactions containing both topics c_i, c_j to the transactions that topic c_i or topic c_j appears in, which can be presented as

$$o_{c_i c_j} = \frac{transaction(c_i) \cap transaction(c_j)}{transaction(c_i) \cup transaction(c_j)} \quad (4)$$

It is obvious that the *overlap* factor is an indicator of the correlation degree of topics c_i and c_j. When $o_{c_i c_j}$ is greater than the specified overlap threshold *minover*, we treat the topics c_i and c_j as a "strong" correlated topic pair. Here, we try to discover some potentially interesting associations between "strong" correlated topic pairs.

Definition 7: An interesting ***interclass association rule*** is described as

$$X \overset{s,c,r}{\Rightarrow} Y \mid C =< c_i, c_j >, \text{ and } o_{c_i c_j} > overlap\_threshold$$

where the relationship of the topic pair $< c_i, c_i >$ is either parent-child or siblings, and topic c_i and topic c_j are "strong" correlated. The parameters *X*, *Y*, *s*, *c*, *r* are the same as Definition 5.

Once the "strong" correlated topic pairs are determined, the algorithm of mining association rules in each "strong" correlated topic pair will be the same as the one for the mining of intra-class association rules (recall Subsection 3.3.2).

Interclass association rules can be used to improve the performance of database selection. For example, in some cases, the user may be interested in the information of one more topics such as two specific siblings with "strong" correlation. In this case, the interclass association rules about these two siblings can be used either to expand the database search space or to help determine the final database ranking order of the selection result.

4 Experimental Design

As described previously, the goal of our work is considered as a step of the post-processing of database selection, which perfects the relative-rough original database-selection results from the database selection tool by using the potentially useful associations among the databases. Therefore, the objective of our experiments is to compare the selection performance of the refined results obtained by the association-rule approach with that of the original results. We conducted a series of experiments on 20 databases that consist of documents from the Reuters-21578 text dataset (Http://www.research.att.com/~lewis/ ~reuters21578. html) - a well-known text categorization dataset for database selection. Each database contains documents of several topic classes.

In this paper, we use the mean-squared root error metric, which is the variation of the well-known Mean Squared Error (MSE) [2]. The mean-squared root error of the collection ranking for a single query is calculated as:

$$Error = \frac{1}{|C|} \cdot \sqrt{\sum_{i \in C} (O_i - R_i)^2} \qquad \textbf{(5)}$$

where: (1) O_i is the position of database S_i in the optimal relevance-based ranking O_Q given a query Q. The optimal ranking O_Q is produced based on the following two criteria: (a) the number of relevant topics in the databases. If database S_i has more classes than database S_j, then S_i is ranked ahead of S_j. That is, $Rank(S_i, S_j) = \{S_i, S_j\}$. (b) the number of relevant documents in the databases. If database S_i has more documents associated with relevant classes than database S_j, then S_i is ranked ahead of S_j. That is, $Rank(S_i, S_j) = \{S_i, S_j\}$. (2) R_i is the position of database S_i in the selection ranking result which is based on the likelihood scores of databases. The database with the largest value of likelihood is ranked 1, the database with second largest value is ranked 2, and so on; (3) C is the set of collections being ranked.

5 Performance Study

5.1 Analysis of Execution Time and The Number of Association Rules

This subsection discusses the effects of the variety of minimum support threshold on the execution time and on the number of association rules generated at different topic levels in the hierarchy. We vary the values of the minimum support threshold in wide range in order to observe all possible differences in the mining. In this manner, we can more clearly determine the effect of the support parameter on the execution time and the size of association rules.

Figure 2-3 show the running time and the number of association rules with respect to the minimum support threshold. It is observed that the smaller the minimum support threshold, the larger the number of the discovered association rules and the more time it takes to generate the rules. The reason for this is that when the minimum support threshold was set to be very small, the size of the candidate itemsets became large. As a result, more association rules would be generated from the candidate itemsets. However, our association-rule mining algorithm requires all the candidate itemsets to be in memory during the mining process, which leads to

most of the available memory space is occupied by the candidate itemsets and consequently less memory is used for the generation of association rules.

It is also easily noted that the effects of various minimum support thresholds on the execution time and the number of association rules vary at different topic levels in the hierarchy. The higher the topic level, the fewer the number of association rules generated and the less the execution time should be taken. This is understandable that since the total number of the query transactions at the high level is much more than that of lower levels, the support threshold at the high level should be very smaller. Hence, we had to flexibly define the support thresholds at different topic levels in order to capture the interesting associations as many as possible.

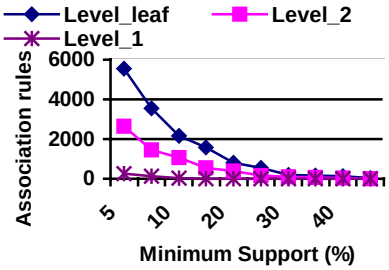

Fig. 2. The effect of different support thresholds on the number of association rules

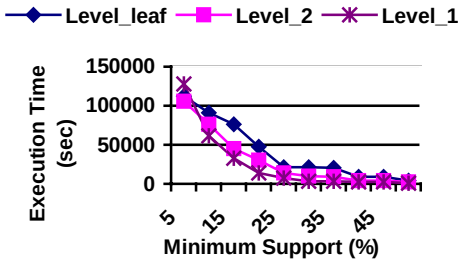

Fig. 3. The effect of different support thresholds on execution time

5.2 Comparison of Selection Performance

Comparing the original selection results by the database selection tool, we examine the selection performance of the refined results obtained by the association-rule (AR) approach with different minimum support thresholds. In Figure 4, we find that the selection performance of the refined results strongly outperforms that of the original ones in the Reuters_21578 dataset. This should not be surprising, because the AR approach provides a much better opportunity to distinguish the relevant databases with the use of the discovered associations between the databases. From Figure 4, it clearly shows that with the AR approach, the mean-squared root error of the refined results is significantly reduced by 24.9% on average against that of the original results. This suggests that potential interesting association rules between the databases should be one of the key factors that affect the selection accuracy.

It is also interesting to note that the selection-performance differentiation in the variety of support thresholds is related to the number of association rules used for selection. Noted that here we mainly examine the effect of associations between the topics at the leaf level on database-selection performance, since the topics at the leaf level include the majority of the topics in the hierarchy. As shown in Figure 5, the selection accuracy increased as the minimum support threshold decreased. It means that the more association rules were used, the larger the chance became to discovery the useful correlations between the databases. However, we can also see that the AR approach with sup_0.1 slightly outperforms that the AR approach with sup_0.2, but the AR approach with sup_0.1 counts the total of about 4,000 association rules and the AR approach with sup_0.2 only counts about 1,500 association rules. The possible reason for this may be because although the AR approach with sup_0.2 has fewer association rules, it still contains most of the potential useful association rules that are large enough to enable significant improvement on database selection performance. It implies that when the collection of query transactions becomes huge, it is possible to choose the larger minimum support threshold with consideration of the trade-off between the memory space occupied and the number of association rules used.

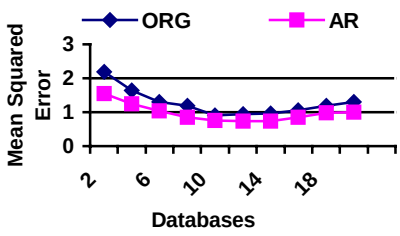

Fig. 4. The comparison of the refined selection results by the association-rule approach (*minsup*=0.2) with the original selection results

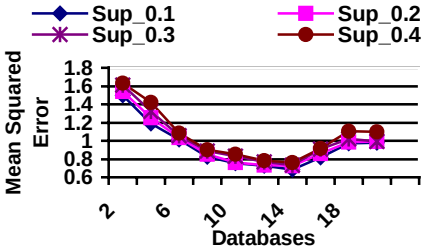

Fig. 5. Selection performance of the association rule approach with different support thresholds

6 Conclusion and Future Works

Information retrieval researchers have developed some sophisticated database-selection tools to locate most relevant databases on the web for the users' information needs. However, they always overlook the potentially useful correlations between the databases at the process of database selection. This paper introduces the use of association rules for the problem of database selection. With the assistance of data mining tools, we extract patterns or associations between distributed databases from a collection of previous selection results, and the discovered knowledge on the databases is in turn used to refine the results from the database selection tools so as to further improve the accuracy of database selection. An association-rule mining approach is proposed to generate intraclass and interclass associations between the databases with the use of a topic hierarchy. We tested the effectiveness of our algorithm on the Reuters-21578 dataset and the experimental results are promising and show some potential in future study on database selection.

However, we view this work as a first step, with a number of interesting problems remaining open and subjected to further research. For example, we are investigating ways to develop more effective discovery algorithms. It appears possible to find other mining algorithms that could perform faster or better the discovery of association rules. Second, the interclass associations described in this paper only involve adjacent topics such as parent-child classes and sibling classes in the hierarchy. Therefore, to discover associations between the child classes with different parent classes is another issue worth exploration. Finding such rules needs future work.

References

[1] Agrawal, R., Imielinski, T., and Swami, A.: Mining Association Rles between Sets of Items in Large Databases. Proceedings of the 1993 Acm Sigmod International Conference on Management of Data, (1993) 26-28.

[2] Callan, J. P., Lu, Z., and Croft, W. B.: Searching Distributed Collections with Inference Networks. Proceedings of the 19th Annual International Acm Sigir Conference on Research and Development in Information Retrieval, (1995) 21-29.

[3] Gravano, L., Garcia-Molina, H., and Tomasic, A.: Gloss: Text-Source Discovery over the Internet. ACM Transactions on Database Systems,Vol. 24 (2). (1999) 229-264.

[4] Hawking, D., and Thistlewaite, P.: Methods for Information Server Selection. ACM Transaction on Information System,Vol. 17 (1). (1999) 40-76.

[5] Kantardzic, M.: Data Mining-Concepts, Models, Methods, and Algorithms, New Jork: IEEE Press (2002).

[6] Yang, H., and Zhang, M.: A Language Modeling Approach to Search Distributed Text Databases. The Proceedings of 16th Australian Joint Conference on Artificial Intelligence, Perth, Australia, (2003) 196-207.