Mining Database Semantic Relationships

Mei-Ling Shyu
Department of Electrical and
Computer Engineering
University of Miami
Coral Gables, FL 33124-0640

Shu-Ching Chen
School of Computer Science
Florida International University
Florida, FL 33199

Abstract

With the explosive growth of the size of databases and the amount of data in them, the discovery of semantic relationships in databases has become a solution to facilitate the integration of those seemingly disparate and autonomous components since it is difficult to detect semantic heterogeneity among databases. An earlier paper provided an approach which uses logical reasoning and object-oriented techniques to discover new semantic relationships of the object classes in multiple databases. However, several prior information are required in the logical reasoning-based knowledge discovery approach. In this paper, an extension of the data mining approach for the discovery of the semantic relationships is proposed. The extended data mining approach includes the original logical reasoning approach with an addition of an association rule mining algorithm.
Key words: Data mining, association rule, databases, semantic relationships.

1 Introduction

Today, many databases exist, especially in a distributed information-providing environment that consists of a network of heterogeneous databases. Unavoidably, users now need shared access across these multiple autonomous databases. Related information important to a global application or request may exist in multiple and incompatible local databases. In addition, semantically related data might be represented in different database schemas under different *database management systems (DBMSs)*. That is, semantically similar pieces of information may have very different names and different data structures in separate local databases [2].

A number of researchers [1, 4, 5, 6] have investigated the problem of semantic interoperability in a heterogeneous database environment. It is very difficult to find semantic heterogeneity since the schemas of the databases do not provide enough semantics to interpret data consistently. For this purpose, the *data mining* technique can be used. In a previous study, we proposed a logical reasoning-based approach that is based on the object-oriented paradigm to discover the semantic relationships in the databases [3]. However, in the logical reasoning-based knowledge discovery approach, the equivalence relation set needs to be provided as *a priori*. This paper is an extension for the knowledge discovery process. In this paper, an extended data mining approach that incorporates an association rule mining algorithm in the original logical reasoning-based approach to form one single framework is proposed. The association rule mining algorithm can exploit the set of quasi-equivalent object classes in the databases as shown in [7]. The discovered quasi-equivalent object class pairs are to approximate the structurally equivalent relationships in the databases. Hence, the equivalence relation set does not need to be provided as a prior information. An empirical study is also given to illustrate how the proposed data mining approach works.

This paper is organized as follows. In next section, the proposed approach consisting of an overview of the logical reasoning-based knowledge discovery approach, the association rule mining algorithm, and an empirical study is introduced. Section 3 concludes the paper.

2 The Proposed Approach

2.1 Overview of the Logical Reasoning-Based Approach

In this paper, C_{ij} is an *object class* in a database d_i, where the index *'i'* indicates the database identification and *'j'* represents the object class identification within the database. O_{ij}^{k} is an object associated with

the object class C_{ij}, where *'k'* denotes the object identification.

- An *object class relationship* $CR(C_{ij}, C_{mn})$ represents the superclass, subclass, and equivalence semantic relationships of two object classes C_{ij} and C_{mn}. Its value is captured through a triplet (P, B, E) where P, B, and E indicate the *suPerclass, suBclass*, and *Equivalence* relations between C_{ij} and C_{mn}, respectively. Each entry with value **1** indicates that the two object classes have the corresponding semantic relationship.

- An *object class relationship matrix* R_{im} is generated to represent the relationships between databases d_i and d_j in the way that every *(j, n)th* element in R_{im} is the value $CR(C_{ij}, C_{mn})$.

- $\mathbf{g}(C_{mn}, C_{ij})$ is the *object class relationship inversion function* such that
  $CR(C_{mn}, C_{ij}) = \mathbf{g}(C_{mn}, C_{ij}) = (B_1, P_1, E_1)$
  if $CR(C_{ij}, C_{mn})=(P_1, B_1, E_1)$.

- $\mathbf{h}(C_{ij}, C_{mn})$ is the *logical reasoning function* which derives the new semantic relationships between two object classes C_{ij} and C_{mn} from different databases, where i<m and n>1.

  $$\mathbf{h}(C_{ij}, C_{mn}) = CR(C_{ij}, C_{m1}) \diamond CR(C_{m1}, C_{mn}),$$

  where $\diamond$ is the logical operator $\wedge$ and is applied to each element in the triplet.

- $\mathbf{TRS_{P_k}}$ is the *total object class relation set* that lists all the semantic relationships of the object classes in the cluster P_k. Here, it is assumed that the network of databases is partitioned into a set of clusters with each cluster (P_k) consisting of those object classes in its member databases.

Initially, $\mathbf{TRS_{P_k}}$ is defined as the following:

$$\mathbf{TRS_{P_k}} = S_{eq} \bigcup RS_1 \bigcup RS_2.$$

Here, S_{eq} is the equivalence set, RS_1 consists of the object class relationships within each database, and RS_2 contains the object class relationships for object classes C_{ij} in d_i and C_{m1} in d_m for i<m. Hence, the new semantic relationships in two databases can be inferred and put in the updated $\mathbf{TRS_{P_k}}$ set. Originally, the equivalence relation set S_{eq} needs to be provided as an input (i.e., prior information) for this logical reasoning-based approach. In order to allow the knowledge discovery process more complete, an association rule mining algorithm that exploits the quasi-equivalence relationships for the object classes in the databases is developed. This association rule mining algorithm is incorporated into the logical reasoning-based knowledge discovery process to make the process of mining database semantic relationships a single framework. The quasi-equivalence relationships are used to approximate the structurally equivalence relationships in the databases.

2.2 The Association Rule Mining Algorithm

Let m and n be object classes, Nmo the total number of object classes, $aff_{m,n}$ the relative affinity relation between m and n, q the total number of queries, $access_k$ the access frequency of query k per time period, and $use_{m,k}$ the usage pattern with value 1 if object class m is accessed by query k and value 0 otherwise. The affinity-based support, confidence, and interest factors are defined as follows.

$$aff_{m,n} = \sum_{k=1}^{q} use_{m,k} \times use_{n,k} \times access_k \quad (1)$$

$$support(m) = \frac{\sum_{k=1}^{q} use_{m,k} \times access_k}{\sum_{k=1}^{q} access_k} \quad (2)$$

$$support(m \rightarrow n) = \frac{aff_{m,n}}{\sum_{k=1}^{q} access_k} \quad (3)$$

$$confidence(m \rightarrow n) = \frac{support(m \rightarrow n)}{support(m)} \quad (4)$$

$$interest(m \rightarrow n) = \frac{support(m \rightarrow n)}{support(m) support(n)} \quad (5)$$

The association mining process consists of two phases.

$\star$ **Steps** for Phase I:

1. For all the 1-itemsets, compute $support(m)$ (Equation 2).

2. For all the 2-itemsets,

   - Compute $aff_{m,n}$ (Equation 1).
   - Compute $support(m \rightarrow n)$ (Equation 3).
   - Compute $confidence(m \rightarrow n)$ (Equation 4).
   - Compute $interest(m \rightarrow n)$ (Equation 5).

3. Initialize
   candidate-pool $= \emptyset, cria1 = 20\%, cria2 = 50\%$, and $iter = 1$.

4. For $m = 1$ to Nmo,

(a) If $iter = 1$ then find the maximal interest value I_m from $interest(m \rightarrow n)$ where an object class n is in a different database since the equivalence relationship can occur only when two object classes are from different databases.

(b) Set interest threshold $IntTd = cria1 \times iter \times I_m$.

(c) For those object classes n's,
if $iter = 1$ then candidate-pool = candidate-pool $\bigcup \{(m,n)\}$ when $interest(m \rightarrow n) \geq IntTd$
else (m,n) is removed from candidate-pool when $interest(m \rightarrow n) < IntTd$.

5. Check the interesting constraint:
if $(m,n) \in$ candidate-pool and $(n,m) \notin$ candidate-pool, then (m,n) is removed from candidate-pool.

6. Check the refinement constraint:
if the number of object class m (which has zero or one (m,n) in candidate-pool) $\geq cria2 \times Nmo$, then goto Phase II else set $iter = iter + 1$ and goto step 4.

⋆ **Steps** for Phase II:

1. Set the confidence threshold $Conf = 99\%$.

2. For each pair (m,n) in candidate-pool,
if $confidence(m \rightarrow n) < Conf$ and $confidence(n \rightarrow m) < Conf$, then (m,n) is removed from candidate-pool.

3. Check if further conditions need to be imposed to remove some unreasonable situations.

2.3 Empirical Study

An empirical study is used to illustrate the incorporation of the association rule mining algorithm in the logical reasoning approach for the discovery of database semantic relationships. Assume there are ten queries that are run on six databases. The access frequencies for the queries are **25, 100, 30, 70, 45, 35, 40, 60, 10**, and **10**, respectively. Each database has a set of *object classes* and each object class has a set of *attributes*. Let the object class be numbered from **1** to **21** and the object classes in the same database have consecutive numbers. Table 1 shows the usage patterns of object classes versus the set of queries. The entity with value 1 indicates that the query accessed the corresponding object class.

Table 1: Query usage patterns

usage	query									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1 ($C_{1,1}$)	0	0	0	0	0	0	0	0	1	1
2 ($C_{1,2}$)	1	1	0	0	1	0	1	0	0	0
3 ($C_{2,1}$)	1	1	1	0	1	0	1	0	0	0
4 ($C_{2,2}$)	1	0	1	1	1	1	0	0	0	0
5 ($C_{2,3}$)	0	0	1	1	0	0	1	1	0	0
6 ($C_{3,1}$)	1	0	1	1	1	1	0	0	0	0
7 ($C_{3,2}$)	0	1	1	1	0	0	1	1	0	0
8 ($C_{3,3}$)	0	0	0	0	1	0	1	0	0	0
9 ($C_{3,4}$)	1	0	0	0	0	0	0	0	0	1
10 ($C_{4,1}$)	0	1	1	1	0	0	1	1	0	0
11 ($C_{4,2}$)	1	0	0	1	0	1	0	0	0	0
12 ($C_{4,3}$)	0	1	0	1	1	0	1	1	0	0
13 ($C_{4,4}$)	0	0	0	0	1	0	0	0	1	0
14 ($C_{4,5}$)	1	0	1	1	0	1	1	0	0	0
15 ($C_{5,1}$)	0	1	0	1	1	0	1	1	0	0
16 ($C_{5,2}$)	0	1	0	0	1	0	0	0	0	1
17 ($C_{5,3}$)	1	0	1	1	0	1	1	0	0	0
18 ($C_{5,4}$)	1	0	1	1	0	1	1	1	0	0
19 ($C_{6,1}$)	1	0	1	1	0	1	0	0	0	0
20 ($C_{6,2}$)	0	0	0	0	0	0	0	1	1	0
21 ($C_{6,3}$)	1	0	1	1	0	0	1	1	0	0

To find the quasi-equivalence relationships for the object classes, the above information is applied to the association rule mining algorithm. First, the values for $cria1$, $cria2$, and $Conf$ need to be predefined. Set $cria1 = 0.2$, $cria2 = 0.5$, and $Conf = 99\%$. Then, the association rule mining algorithm is executed based on these values. Four iterations were executed in Phase I. For each iteration, the interest threshold, interest constraint, and refinement constraint were checked. The interest threshold value increases in proportional to the number of iterations. In the first iteration, there were **265** object class pairs satisfying the minimal interest threshold checking and **260** pairs remaining in the candidate-pool after the interest constraint checking. The refinement constraint checking failed so it goes to next iteration for Phase I. In the second iteration, **200** pairs satisfied the minimal interest threshold checking and **44** pairs were removed from the candidate-pool after the the interest constraint checking. The refinement constraint checking failed, too. In the third iteration, there were **142** object class pairs satisfying the minimal interest threshold checking and the candidate-pool shrank to **90** pairs after the interest constraint checking. Again, the refinement constraint checking failed. Then, it went to the fourth iteration that **200** pairs satisfied the minimal interest threshold checking and **44** pairs were removed from the candidate-pool after the the interest

constraint checking. Now, the refinement constraint was satisfied so that Phase I stopped.

In Phase II, the confidence threshold $Conf$ is set to be 99%. There were **18** pairs left in the candidate-pool after the confidence constraint checking. That is, **12** more pairs were removed from the candidate-pool in the confidence constraint checking. Finally, it was checked whether some unreasonable situations need to be removed. In this example, there is no unreasonable situation in the candidate-pool. Hence, the mining algorithm stopped and the final number of object class pairs that have the quasi-equivalence relationships is **18**. Since the quasi-equivalence relationship (m, n) is the same as the quasi-equivalence relationship (n, m), there are **9** quasi-equivalent object class pairs when the order of the object classes is not considered. In fact, these nine pairs match with the structurally equivalent object classes known to be existing in the databases. This result shows that the association rule mining algorithm exploits the equivalent object class pairs correctly.

After the quasi-equivalence relationships are discovered, the relationship derivation algorithm can be executed. For example, the semantic relationships between C_{11} and C_{22} and between C_{12} and C_{22} can be discovered in the following manner.

$$\begin{aligned}\mathbf{h(C_{11}, C_{22})} &= CR(C_{11}, C_{21}) \diamond CR(C_{21}, C_{22}) \\ &= (1,0,0) \diamond (1,0,0) = (1,0,0). \\ \mathbf{h(C_{12}, C_{22})} &= CR(C_{12}, C_{21}) \diamond CR(C_{21}, C_{22}) \\ &= (1,1,1) \diamond (1,0,0) = (1,0,0).\end{aligned}$$

Here, $CR(C_{12}, C_{21}) = (1,1,1)$ (i.e., C_{12} and C_{21} are equivalent) is obtained from the result of the association rule mining algorithm. In addition, the object class relationship matrix for every pair of databases can be generated by the logical reasoning-based approach. For example, the object class relationship matrix R_{12} for databases d_1 and d_2 is shown as follows.

$$R_{12} = \begin{array}{c} \\ C_{11} \\ C_{12} \end{array} \begin{array}{c} \begin{array}{ccc} C_{21} & C_{22} & C_{23} \end{array} \\ \begin{pmatrix} (1,0,0) & (1,0,0) & (0,0,0) \\ (1,1,1) & (1,0,0) & (0,0,0) \end{pmatrix} \end{array}$$

All the semantic relationships in these two databases are captured in this matrix. Since the association rule mining algorithm is incorporated in the original logical reasoning-based approach, the new data mining process becomes a single framework for discovering the database semantic relationships.

3 Conclusions

In this paper, a new data mining approach that extends the original logical reasoning-based approach with the incorporation of an association rule mining algorithm is presented. A new set of affinity-based measures to augment the standard measures of support, confidence, and interest is defined in the association rule mining algorithm. The result of the empirical study shows that the association rule mining algorithm discovers the set of quasi-equivalent object class pairs that matches the structural equivalent object class pairs in the databases correctly. Once the semantic relationships are discovered, the information can be used to resolve the heterogeneity and hence to speed up schema integration.

References

[1] S. Bergamaschi, S. Castano, and M. Vincini, "Semantic integration of semistructured and structured data sources," *SIGMOD Record*, vol. 28, no. 1, pp. 54-59, March 1999.

[2] M.W. Bright, A.R. Hurson, and S. Pakzad, "Automated resolution of semantic heterogeneity in multidatabases," *ACM Transactions on Database Systems*, vol. 19, no. 2, June 1994.

[3] S.-C. Chen, M.-L. Shyu and Chi-Min Shu, "Discovering semantic relationships among object classes in database systems," *ISCA 1st International Conference On Information Reuse And Integration (IRI-99)*, pp. 100-103, November, 1999.

[4] S.E. Lander and V.R. Lesser, "Sharing metainformation to guide cooperative search among heterogeneous reusable agents," *IEEE Transactions on knowledge and Data Engineering*, vol. 9, no. 2, pp. 193-208, March/April 1997.

[5] M. Roantree, J. Murphy, and W. Hasselbring, "The OASIS multidatabase prototype," *SIGMOD Record*, vol. 28, no. 1, pp. 97-103, March 1999.

[6] A. Tomasic, L. Raschid, and P. Valduriez, "Scaling access to heterogeneous data sources with DISCO," *IEEE Transactions on knowledge and Data Engineering*, vol. 10, no. 5, pp. 808-823, September/October 1998.

[7] M.-L. Shyu, S.-C. Chen, and R. L. Kashyap, "Discovering quasi-equivalence relationships from database systems," *ACM Eighth International Conference on Information and Knowledge Management (CIKM'99)*, pp. 102-108, November, 1999.