Discovering Semantic Relationships Among Object Classes in Database Systems

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

Mei-Ling Shyu
School of Electrical and
Computer Engineering
Purdue University
West Lafayette, IN 47907

Chi-Min Shu
Department of Environmental
Safety Engineering
National Yunlin University of
Science and Technology
Yunlin, Taiwan, R.O.C

Abstract

Providing integrated access to heterogeneous databases in a distributed information-providing environment is challenging for cooperation and interoperability. In addition, the accessing of many data sources has aggravated problems for users of heterogeneous databases because of the heterogeneities among databases. To solve these problems, the discovery of semantic knowledge regarding the object classes present in the databases can be used as the pre-processing procedure for schema integration. This will also assist in increasing the interoperability of the heterogeneous databases. In this paper, we explore a new data mining approach which discovers new semantic relationships of the object classes in different databases. The proposed approach uses logical reasoning and object-oriented techniques to bridge heterogeneity in a large scale heterogeneous database environment. Mechanisms for accomplishing the objective are presented in theoretical terms, along with a running example.

Key words: object-oriented, data mining, databases

1 Introduction

In a distributed information-providing environment, semantically related data might be represented in different database schemas under diverse *database management systems (DBMSs)*. Retrieving information from them is a challenge since incompatibilities exist among the databases. To provide an integrated access to multiple heterogeneous databases, two issues need to be discussed. First, how to discover the semantically related information, i.e., information such as whether two object classes have a superclass, subclass, or equivalence semantic relationship, to support integration. Second, how to perform schema integration to provide a uniform access interface. A number of researchers [1, 2, 4, 6] have investigated the problem of semantic interoperability in a heterogeneous database environment. However, most of them focus only on the second issue.

Advanced data storage technology and database management systems have increased our capabilities to collect and store data of all kinds. However, our ability to interpret and analyze the data is still limited, creating an urgent need to accelerate discovery of information in databases. As pointed out by [3], there is a need and an opportunity for at least a partially automated form of *knowledge discovery in databases (KDD)*, or *data mining* to handle the huge size of real-world database systems. Data mining is the method of discovering useful information such as rules and previously unknown patterns existing between data items embedded in large databases. Because of the rapid growth of databases and data, how to effectively utilize the large amount of accumulated data becomes important.

In a previous study, we proposed an object-oriented split/cluster approach for managing a network of databases [5]. The proposed split/cluster approach is affinity-based and partitions the database according to their degree of affinity. The network of databases is recursively split into a set of clusters and a cluster hierarchy is generated. This paper is an extended work to discover the new semantic relationships among the object classes in each cluster. In this paper, our focus is on discovering and reasoning about the semantic aspects of the object classes for the first issue. A logical reasoning-based knowledge discovery approach is proposed to exploit the new semantic relationships among the object classes in the databases. The proposed approach is applied to each cluster to discover the new relationships. Clearly, discovering the new semantic relationships for object classes across multiple databases will not only help schema integration but also speed up query processing. Toward this end, we explore a new data mining capability that involves mining new semantic relationships among object classes in a network of databases. The proposed approach, supporting data mining, logical reasoning, and object-oriented techniques, allows the analysis of source descriptions for discovering a set of semantically related information and provides support for schema integration.

This paper is organized as follows. In next section, we briefly give the meaning of various terms and expressions used throughout this paper. The proposed logical reasoning based knowledge discovery approach is introduced in Section 3. Section 4 concludes the paper.

2 Glossary

2.1 Object-Oriented Paradigm

The object-oriented paradigm is adopted in our proposed approach. Since things in the world around us have properties of features, we can think of data as an object class with its defining objects (attributes).

Definition 1 : An *object class* in a database d_i is any distinguishable entity which contains two or more objects whose descriptions are available in d_i. It is denoted by C_{ij}, where the index *'i'* indicates the database identification and *'j'* represents the object class identification within the database.

Definition 2 : A class of *objects*, O^k_{ij}, where *'k'* denotes the object identification, associated with an object class C_{ij} are to characterize C_{ij} and to represent the information pertaining to the d_i available to the application queries. The values of *i, j,* and *k* are unique.

In order to illustrate the way our approach works, the following example is used. Consider a heterogeneous database environment with six databases. For simplicity, only a part of the objects is shown here. In d_1, it provides two distinguishable object classes – *resident* and *employee*, which are denoted by C_{11} and C_{12}, respectively. Moreover, suppose three objects, *name, age,* and *address*, are to characterize the object class *resident*.

Example:

$d_1 = \{resident, employee\} = \{C_{11}, C_{12}\}$;

$C_{11} \Rightarrow \{name, age, address\} \Rightarrow \{O^1_{11}, O^2_{11}, O^3_{11}\}$;

$d_2 = \{emp, faculty, dept\} = \{C_{21}, C_{22}, C_{23}\}$;

$d_3 = \{faculty, prof, secretary, engineer\}$
$=\{C_{31}, C_{32}, C_{33}, C_{34}\}$;

$d_4 = \{professor, class, student, grade, teaching\_assist\}$
$=\{C_{41}, C_{42}, C_{43}, C_{44}, C_{45}\}$;

$d_5 = \{student, RA, TA, div\} = \{C_{51}, C_{52}, C_{53}, C_{54}\}$;

$d_6 = \{course, room, department\} = \{C_{61}, C_{62}, C_{63}\}$.

2.2 Semantic Relationships

Definition 3: $CR(C_{ij}, C_{mn})$, an *object class relationship*, 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.

Given C_{ij} and C_{mn}, the following relationships among them are considered:

- superclass relation:

$$\mathbf{P} = \begin{cases} 1 & \text{if } C_{ij} \text{ is a superclass of } C_{mn} \\ 0 & \text{otherwise} \end{cases}$$

- subclass relation:

$$\mathbf{B} = \begin{cases} 1 & \text{if } C_{ij} \text{ is a subclass of } C_{mn} \\ 0 & \text{otherwise} \end{cases}$$

- equivalence relation:

$$\mathbf{E} = \begin{cases} 1 & \text{if } C_{ij} \text{ is equivalent to } C_{mn} \\ 0 & \text{otherwise} \end{cases}$$

In a single database, the object class equivalence relation cannot exist between two different object classes since a database schema should be non-redundant. Hence, the element E in the triplet (P,B,E) is always 0 within one database. In addition, for the purpose of the derivation of the relationship between two component database schemas, if two object classes are equivalent, let (P,B,E)=(1,1,1).

The above semantic relationships among the object classes either in a database or in different databases can be captured through the *object class relationship matrix* and the *object class relationship inversion function.*

Definition 4: R_{im}, an *object class relationship matrix*, represents the relationships between database d_i and d_j in the way that every *(j, n)th* element in R_{im} is the value $CR(C_{ij}, C_{mn})$.

Definition 5: $\mathrm{g}(C_{mn}, C_{ij})$ is the *object class relationship inversion function* such that

$CR(C_{mn}, C_{ij}) = \mathrm{g}(C_{mn}, C_{ij}) = (B_1, P_1, E_1)$
if $CR(C_{ij}, C_{mn})$=(P_1, B_1, E_1).

2.3 Prior Information

The following three relation sets are provided as *a priori* for the proposed approach in the current stage. However, algorithms for constructing these three sets are under investigation. Tables 1 to 3 list the three relation sets in the forms of the triplets for the above example.

1. Equivalence set (S_{eq}) which contains those pairs of object classes that are equivalent.

   If the pairs of object classes in S_{eq} have different names, then they are synonymous. Therefore, naming conflicts (synonyms) are captured in S_{eq}.

2. Relation set (RS_1) which contains the object class relationships within each database.

3. Relation set (RS_2) which contains the object class relationships for object classes C_{ij} in d_i and C_{m1} in d_m for i<m.

Table 1: S_{eq} (Object class equivalence relationships)

	(P,B,E)
CR(C_{12},C_{21})	(1,1,1)
CR(C_{22},C_{31})	(1,1,1)
CR(C_{23},C_{54})	(1,1,1)
CR(C_{23},C_{63})	(1,1,1)
CR(C_{32},C_{41})	(1,1,1)
CR(C_{42},C_{61})	(1,1,1)
CR(C_{43},C_{51})	(1,1,1)
CR(C_{45},C_{53})	(1,1,1)
CR(C_{54},C_{63})	(1,1,1)

Table 2: Partial RS_1 (object class relationships)

		(P,B,E)
d_1	CR(C_{11},C_{12})	(1,0,0)
d_2	CR(C_{21},C_{22})	(1,0,0)
	CR(C_{21},C_{23})	(0,0,0)
	CR(C_{22},C_{23})	(0,0,0)
d_3	CR(C_{31},C_{32})	(1,0,0)
	CR(C_{31},C_{33})	(1,0,0)
	CR(C_{31},C_{34})	(1,0,0)
	CR(C_{32},C_{33})	(0,0,0)
	CR(C_{32},C_{34})	(0,0,0)
	CR(C_{33},C_{34})	(0,0,0)

Table 3: Partial RS_2 (object class relationships)

(P,B,E)	C_{21}	C_{31}	C_{41}	C_{51}	C_{61}
C_{11}	(1,0,0)	(1,0,0)	(1,0,0)	(0,0,0)	(0,0,0)
C_{12}	(1,1,1)	(1,0,0)	(1,0,0)	(0,0,0)	(0,0,0)
C_{21}		(1,0,0)	(1,0,0)	(0,0,0)	(0,0,0)
C_{22}		(1,1,1)	(1,0,0)	(0,0,0)	(0,0,0)
C_{23}		(0,0,0)	(0,0,0)	(0,0,0)	(0,0,0)

Moreover, all the semantic relationships among the object classes in a database can be derived directly from the prior information and the *object class relationship inversion function.* Let the object class relationship matrix R_{ii} for d_i be constructed as follows:

$$R_{ii} = \bigcup_{C_{ij},C_{ik} \in d_i} CR(C_{ij}, C_{ik})$$

3 Inference of New Semantic Relationships in Databases

While all the semantic relationships among the object classes in a database can be derived directly, the derivation of the semantic relationships of the object classes in a cluster requires a new mechanism to perform the task. For this purpose, a logical reasoning-based mechanism is proposed for the inference of new semantic relationships in two databases. A new set of semantic relationships among object classes in two different databases is derived by applying the proposed *logical reasoning function* and kept in a *total object class relationship set* TRS_{P_k} for the cluster P_k. A cluster P_k contains those object class present in its member databases, and $\mathbf{TRS_{P_k}}$ lists all the semantic relationships of those object classes in P_k.

Definition 6 : Let $\mathbf{TRS_{P_k}}$ be the *total object class relation set* for the cluster P_k.

Definition 7: $\mathbf{h}(C_{ij}, C_{mn})$ 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.

For example, the semantic relationships between C_{11} and C_{22} can be discovered in the following manner.

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

Table 4: *Relationship derivation algorithm*

```
INPUT :
(1) $S_{eq}$ relation set    (2) $P_k$
(3) Initially, $\mathbf{TRS_{P_k}} = S_{eq} \bigcup RS_1 \bigcup RS_2$
OUTPUT :
(1) Updated $TRS_{P_k}$ set
METHOD :
For any two object classes $C_{ij}$ and $C_{mn}$ in the
databases in $P_k$, where i<m and n>1 {
  if (($C_{ij}, C_{mn}$) $\notin TRS_{P_k}$) {
    if ($\exists$ $C_{pq}$ satisfying ($C_{ij}, C_{pq}$) $\in S_{eq}$) {
      For every $C_{pq}$ {
        if (($C_{pq}, C_{mn}$) $\in TRS_{P_k}$)
          $CR(C_{ij}, C_{mn}) = CR(C_{pq}, C_{mn})$;
        else if (p>m || q>n)
          $CR(C_{ij}, C_{mn}) = \mathbf{g}(C_{pq}, C_{mn})$;
      }
    }
    else $CR(C_{ij}, C_{mn}) = \mathbf{h}(C_{ij}, C_{mn})$;
    $TRS_{P_k} = TRS_{P_k} \bigcup (C_{ij}, C_{mn})$;
  }
}
```

Table 4 lists the proposed *relationship derivation algorithm* which incrementally updates $\mathbf{TRS_{P_k}}$. As can be seen from Table 4, the inputs of the algorithm are the S_{eq} relation set, each cluster P_k, and the initial *total object class relation set.* For every pair of object classes in P_k, the $\mathbf{TRS_{P_k}}$ set is updated incrementally if one of the semantic relationships exists between these two object classes. This algorithm is executed iteratively on all the pairs of object classes in a cluster to explore the new semantic relationships among the object classes in that cluster. The newly discovered semantic relationships in each cluster can then be used to assist in the integration task in that cluster. In addition, if there exist multiple clusters in the heterogeneous database system, then this algorithm is applied to each cluster in the cluster hierarchy.

After the proposed *relationship derivation algorithm* is applied to the object classes in those databases in each cluster, R_{ij} where i<j, for d_i and $d_j \in$ some cluster P_k, can be obtained easily. Moreover, only R_{im} for i<=m needs to be constructed since R_{mi} is the transpose of R_{im}. For example,

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

The discovery of new semantic relationships is crucial in schema integration as integration is achieved by the detection/resolution of the semantic conflicts and the derived knowledge. For example, if two object classes belonging to two different databases in the same cluster are declared to have the equivalence relationship and are to be integrated, the relationships among their objects (attributes) can be easily derived. Considering the object classes *employee* in d_1 and *emp* in d_2, and knowing that the object *name* in *employee* is equivalent to the object *ename* in *emp*, this type of incompatibility can be detected and handled in the integration procedure. In addition, apart from its own objects, each object class acquires a set of objects (or methods) from object classes having a superclass relationship with the object class through inheritance. Therefore, our proposed logical reasoning-based knowledge discovery approach can be used as the pre-processing procedure for schema integration.

4 Conclusions

In this paper, we have proposed a new data mining approach which adopts the object-oriented paradigm to discover new semantic relationships among object classes from a network of heterogeneous databases in a distributed information-providing environment. In particular, we have suggested a logical reasoning-based mechanism that uses logical operators to automatically derive new semantic relationships from the existing knowledge. Various semantic knowledge pertinent to the object classes of the databases is explored. The use of logical operators is easy to understand and simplifies the data mining process. During the knowledge acquisition process, some semantic conflicts can be detected and later be resolved in the schema integration process. Hence, the proposed approach can be applied as the pre-processing procedure for schema integration. Furthermore, concepts such as the equivalence, superclass, and subclass are utilized to facilitate the creation of the integrated schemas. To enhance the capabilities of the proposed mechanism, further experiments will be conducted in the future.

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] 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.

[3] G. Piatetsky-Shapiro, "Knowledge discovery in real databases: A report on the IJCAI-89 Workshop," *AI Magazine*, Vol. 11, No. 5, Special issue, pp. 69-70, Jan. 1991.

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

[5] M-L. Shyu and S-C. Chen, "An object-oriented approach for managing a network of databases," to appear on IASTED Software Engineering and Applications (SEA'99).

[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.