A Framework for Discovering Knowledge from Distributed and Heterogeneous Databases

Alex G. Büchner[§], Sarabjot S. Anand[§], David A. Bell[†], John G. Hughes[‡]

[§]Northern Ireland Knowledge Engineering Laboratory, University of Ulster
[†]School of Information and Software Engineering, University of Ulster
[‡]Faculty of Informatics, University of Ulster
{ag.buchner, ss.anand, da.bell, jg.hughes}@ulst.ac.uk

Abstract

Knowledge discovery from multi-databases requires support for handling heterogeneous schema integration and heterogeneous knowledge integration. The issue of heterogeneous schema integration using knowledge discovery has been addressed by [Dao95], [Hay90], [Kin96], and [Li94], among others.
In this paper we address the area of heterogeneous knowledge integration, which hasn't been dealt with thoroughly. We propose a framework for discovering knowledge from distributed and heterogeneous data sources. A methodology is introduced that allows the induction of association rules from horizontally, as well as vertically distributed data sources. We also present the architectural extensions to the developed Mining Kernel System ([Ana96b]) to support such discovery.

Keywords: knowledge discovery, distributed databases, heterogeneous databases, knowledge integration

1. Introduction

Knowledge discovery from multiple databases can be sub-divided into two areas: ***semantic heterogeneity integration*** in heterogeneous databases and ***semantic knowledge integration*** in distributed and heterogeneous databases.

Semantic schema integration is mainly concerned about guaranteeing semantic equivalence between different data sources, independent of their underlying data model ([Bel92], [Dre93]). Several solutions have been provided to tackle semi-automated heterogeneous schema integration, using knowledge discovery techniques, which are based on different comparison techniques, e.g. comparison of attribute names, including those with synonymous and homonymous meaning ([Hay90]), searching for related attribute values and domains ([Li94]) based on traditional statistics methods and neural networks, comparison of database meta information, such as cardinality, primary and foreign keys ([Rib95]), semantic integrity constraints, access grants, triggers, and so forth ([Lar89]), or comparison of relationships among entire sets of attributes ([Dao95]).

Semantic knowledge integration deals with knowledge that has been previously discovered. In order to use knowledge for further discovery, however, existing data mining techniques have to be modified, which is also an element of semantic knowledge integration. In this paper we describe a framework that provides those two features for mining data from multiple databases.

In section 2 problems of knowledge discovery from distributed and heterogeneous databases are outlined and preliminary information is given. Section 3 provides a framework for mining association rules from horizontally and vertically distributed data. Section 4 describes a prototype that has been developed to provide access to multiple data sources and mine data from them. Finally, conclusions are drawn, related work is evaluated, and further research is outlined.

2. Background and Related Work

Assuming that information about data - independent of its distribution and any form of semantic heterogeneity - is available, a data mining solution has to provide a facility to describe distributed data mapping (figure 1). This means that whenever semantic heterogeneity has been solved (manually or semi-automated), data mining solutions can be applied on distribution level, because semantic equivalence is guaranteed.

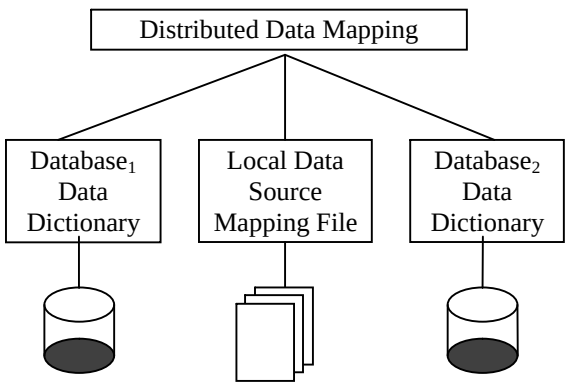

Figure 1: Distributed Data Mapping

The view described over a distributed and possibly heterogeneous set of data sources needs to be split into views on the individual data sources and discovery carried out on each individual site. This is essential for two reasons: Often the distributed and heterogeneous data sources do not provide facilities for performing joins across databases. Also, such joins - where available - are not very efficient, making data access a bottleneck within the discovery process. When splitting a distributed view into individual data definition files it is important to keep track of information that is required to combine knowledge discovered from individual data sources. The information required for the combination differs based on whether the data stored at the distributed and heterogeneous sites is based on a horizontal split of the data (i.e. the database relations at the different sites is the same) or a vertical split (i.e. the data stored at the different sites complement each other and their join provides additional information about each data tuple).

3. A Knowledge Discovery Framework

We now present a framework that supports association algorithms for horizontally as well as vertically distributed data[1]:

Horizontal Distribution
In the case of horizontal distribution, combination of discovered rules is straightforward. Different underlying uncertainty models provide techniques that can be used to integrate data.

For example, if the discovery process uses probability theory, a combination operator like the Linear Opinion pool or intersection hypergraphs ([McC96]) can be applied to the 'sifted nuggets'. If Evidence Theory is used as the underlying uncertainty model ([Ana96a]), the Proportional Belief Transfer Operator ([Ana96c]) could be used for the combination. The only information required for such a combination are the semantic equivalence transformations for data stored in the heterogeneous databases at different levels of coarseness.

Vertical Distribution
If data is distributed vertically across different sites, combining discovered information becomes more complicated. In such cases the join attributes that describe how tuples in one database are related to tuples in another database (usually implemented as foreign keys) must be kept track of. Each of the rules of knowledge discovered at the individual sites must have associated with it, a set of values of the join attributes from the database that satisfy that piece of knowledge. For example, consider an association algorithm that discovers the following large itemsets from two different data sources:

i_1: AccountBalance = High
OwnedProperty = True
support = 0.67

i_2: Salary > 25000
OwnedProperty = True
support = 0.85

Then we *cannot* conclude that

$i_1 \cap i_2$: AccountBalance = High
Salary > 25000
OwnedProperty = True

$$support_{max} = \frac{(|i_1| * 0.67 + |i_2| * 0.85)}{(|i_1| + |i_2|)}$$

is also large without any additional information[2]. The only possibility to calculate this new itemset's support value is to store the join attribute values supporting the two itemsets at the original distributed sites. In the above example if the two sites have a join attribute CustomerNumber, then we must calculate the intersection of the sets of the CustomerNumber values supporting the two itemsets i_1 and i_2 which gives support for the itemset $i_1 \cap i_2$.

At the individual data sources the discovery must take into account the relationship between the different data sources. Consider the two data sources of i_1 and i_2 in the example above. If the cardinality between the two data sources is 1:1, the number of tuples per employee at each site is identical. If the join attribute is unique (primary key), association algorithms discussed in literature ([Agr94]) can be applied at each of the sites. However, if the cardinality between two sites is 1:n data source 2 has a number of tuples associated with each employee record in data source 1. In this case, discovery can be carried out at data source 1 using simple association algorithms. However at data source 2 we must use an association algorithm that discovers associations between tuples with the same join attribute value as well.

Table 1: Data Source 1

Emp_no	Company_Years	Age	Car_Type	Type
10011	10	55	None	Labourer
10112	5	34	None	Labourer
10113	3	45	Porsche	Manager

Table 2: Data Source 2

Emp_no	Salary_Type	Income
10011	Basic Wages	15000
10011	Bonus	1200
10011	Overtime	1700
10112	Overtime	700
10112	Bonus	1300
10112	Basic Wages	17000
10113	Basic Wages	550000

[1] Hybrid fragmented data ([Ösz91]), i.e. vertically *and* horizontally distributed information, is not considered in here for simplicity.

[2] |S| denotes the cardinality of the set S

Consider the data sample shown in Tables 1 and 2. In data source 2 we must discover large itemsets of the type

Basic_Wages < X
Bonus < Y
Overtime < Z
support 0.xy {a_1, a_2, …, a_n}

instead of itemsets of the type

Salary_Type = A
Income < B
support 0.xy {a_1, a_2, …, a_n}

Such rules cannot be discovered using association algorithms suggested in [Agr94]. From the data in tables 1 and 2 we can discover a number of itemsets, some of which are given below.

$i_{1.1}$: Company_Years < 6
support 0.67 {10012, 10113}

$i_{1.2}$: Employee_Type = Labourer
Car_Type = None
support 0.67 {10011, 10112}

$i_{2.1}$: Basic Wages < 20000
Bonus < 1500
Overtime < 2000
support 0.67 {10011, 10112}

Now combining these itemsets we get

$i_{1.1} \cap i_{1.2}$: Company_Years < 6
Basic_Wages < 20000
Bonus < 1500
Overtime < 2000
support 0.33 {10012}

$i_{1.2} \cap i_{2.1}$: Employee_Type = Labourer
Car_Type = None
Basic Wages < 20000
Bonus < 1500
Overtime < 2000
support 0.67 {10011, 10112}

The first itemset would not be a large itemset anymore if the minimum support threshold was 2/3. The shown simplistic scenario lead to an implosion of itemsets. The opposite, i.e. an explosion of itemsets cannot result, because the maximum number of integrated itemsets is the minimum number of itemsets of all participating data sources.

If the cardinality between entities is m:n, a similar technique can be applied. Due to the fact that m:n relationships are modelled using two 1:n relationships, the given techniques can be re-used. The only supplementary requirement is the need of keeping track of the attribute join fields.

4. Prototypical Work

The Mining Kernel System MKS ([Ana96b]) is a data mining prototype developed at the University of Ulster to support the Data Mining Process. MKS consists of 2 main modules: The Interface Module and the Mining Module. The Interface Module consists of the Virtual Data Library for accessing disparate data sources, the Knowledge Input-Output Library for interfacing with the Knowledge Bases, and the User Interface. The Mining Module consists of the Evidence Theory Library allowing algorithms with Evidence Theory as the underlying uncertainty model, the Set Manipulation Library for performing set operations, the Information Theoretic Library providing Information Theoretic Measures, the Statistical Library providing statistical functionality and the Knowledge Representation Library for manipulation of knowledge.

The Virtual Data Library (VDL), allows every MKS algorithm to access data as defined by the user in the Data Source Mapping (DSM) file in the form of a logical data tuple, no matter what the underlying structure of the data is[3]. The DSM file resembles the definition of a relational view. On top of these DSM files, a Distributed Data File (DDF) has to be created, which also allows the user to define a view over distributed data sources and includes facilities such as computable expressions to solve semantic equivalence. Alternatively, DSM files can be created automatically from a given DDF file, which supports schema evolution of single databases. An example DSM and DDF file is given below, bold font indicating MKS keywords:

```
# A sample Ingres DSM file
TYPE            Ingres
SOURCE          /mks/ingres/customers.ing
MAPPING
Number          Key
Postcode        Postcode
DOB             DateOfBirth
income          Income
WHERE CLAUSE
                Income > 20000
```

Figure 2: A sample DSM file

```
# A sample DDF file
SOURCE MAPPING
Ingres          /mks/ingres/customers.ing       i
Oracle          /mks/oracle/customers.ora       o
Unix            /mks/dsm/customers.map          u
ATTRIBUTE MAPPING
Postcode        u.postcode, i.Postcode, o.person.zip
DoB             i.DOB, (u.age - TODAY)
Income          i.income, (o.salary - o.travelexpenses)
WHERE CLAUSE
                i.Key = u.Code AND i.key = o.Number
                Income > 25000 AND
                DoB < 29/11/1968
```

Figure 3: A sample DDF file

Based upon the given virtual data layer, algorithms, which re-use functionality from the Mining

[3] To allow identical access to flat text files, a separate mapping file is supported, which simulates a data dictionary.

Module, can be used to discover knowledge from the named data sources.

5. Conclusions and Further Work

A framework for knowledge discovery from multiple database has been presented, which supports association rules from horizontally as well as vertically distributed data sources. Data mining techniques that are applied on contents level on each individual database, have been combined with information on structure level (primary and foreign keys) to allow semantic knowledge integration in multi-databases.

[McC96] used a purely statistical approach to combine continuous or ordinal data from domain related distributed data sources using intersection hypergraphs. The solution is limited to numerical data only and does not provide any support for vertically distributed data.

[Rib95] extended the INLEN architecture to discover association rules from distributed databases. The AQ algorithm has been modified to handle primary and foreign key information from two data sources. The discovered knowledge is then applied to any other databases which is part of the distribution. Our approach differs in that it supports association rules from horizontally and vertically distributed data, it provides semantic equivalence functionality, and thus, is fully prepared to discover knowledge from heterogeneous databases.

Work in progress includes the following: Support for classification algorithms using distributed learning techniques such as [Cha96]'s arbiter trees, improved implosion algorithms to derive appropriate association rules, independence of the underlying data model, incorporating distributed domain knowledge, and an overall support for heterogeneous database systems, i.e. heterogeneous schema *and* knowledge integration as part of a holistic data mining solution.

6. References

[Agr94] R. Agrawal, R. Srikant, Fast Algorithms for Mining Association Rules in Large Databases, in *Proc. of the 20th VLDB Conf.*, pp. 487 - 499, 1994.

[Ana96a] S.S. Anand, D.A. Bell, J.G. Hughes: EDM: A general framework for Data Mining based on Evidence Theory, in *Data & Knowledge Engineering*, 18:189-223, 1996.

[Ana96b] S. S. Anand, B. W. Scotney, M. G. Tan, S. I. McClean, D. A. Bell, J. G. Hughes, I. C. Magill, Designing a Kernel for Data Mining, To appear in *IEEE Expert Special Issue on Data Mining*, 1996.

[Ana96c] S.S. Anand, D.A. Bell, J.G. Hughes: Aspects of Handling Uncertainty in Knowledge Discovery, in *Proc. of the 6th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems IPMI'96*, pp. 903-909, 1996.

[Bel92] D.A. Bell, J. Grimson: Distributed Database Systems. Addison-Wesley, 1992.

[Cha96] P.K. Chan, S.J. Stolfo: Sharing Learned Models among Remote Database Partitions by Local Meta-learning, in *Proc. of the 2nd Int. Conf. on Knowledge Discovery & Data Mining (KDD-96)*, pp. 2-7, 1996.

[Dao95] S. Dao, B. Perry: Applying a Data Miner to Heterogeneous Schema Integration, in *Proc. of the 1st Int. Conf. on Knowledge Discovery and Data Mining (KDD-95)*, pp. 63-68, 1995.

[Dre93] P. Drew, R. King, D. McLeod, M. Rusinkiewicz, A. Silberschatz: Report of the Workshop on Semantic Heterogeneity and Interoperation in Multidatabase Systems, in *ACM SIGMOD RECORD*, 22(3):47-56, 1993.

[Hay90] S. Hayne, S. Ram: Multi-user view integration system (MUVIS): An expert system for view integration, in *Proc. in the 6th Int. Conf. on Data Engineering*, pp. 402-409, 1990.

[Lar89] J.A. Larson, S.B. Navathe, R. Elmasri: A theory of attribute equivalence in database with application to schema integration, in *Transaction on Software Engineering*, 15(4):449-463, 1989.

[Li94] W.S. Li, C. Clifton: Semantic Integration in Heterogeneous Databases Using Neural Networks, in *Proc. of the 20th VLDB Conf. Santiago, Chile*, pp. 1-12, 1994.

[McC96] S.I. McClean, B. Scotney: Distributed Database Management for Uncertainty Handling in Data Mining, in *Proc. of the UNICOM Data Mining Seminar*, pp. 104-118, 1996.

[Ösz91] M.T. Öszu, P. valdurez: Principles of Distributed Database Systems, Prentice Hall, Englewood Cliffs NJ, 1991.

[Rib95] J.S. Ribeiro, K.A. Kaufmann, L. Kerschberg: Knowledge Discovery from Multiple Databases, in *Proc. of the 1st Int. Conf. on Knowledge Discovery and Data Mining (KDD-95)*, pp. 240-245, 1995.