Integration of Data Sources through Data Mining

Andreas Koeller
Montclair State University, USA

INTRODUCTION

Integration of data sources refers to the task of developing a common schema as well as data transformation solutions for a number of data sources with related content. The large number and size of modern data sources make manual approaches at integration increasingly impractical. Data mining can help to partially or fully automate the data integration process.

BACKGROUND

Many fields of business and research show a tremendous need to integrate data from different sources. The process of data source integration has two major components.

Schema matching refers to the task of identifying related fields across two or more databases (Rahm & Bernstein, 2001). Complications arise at several levels, for example

- Source databases can be organized by using several different models, such as the relational model, the object-oriented model, or semistructured models (e.g., XML).
- Information stored in a single table in one relational database can be stored in two or more tables in another. This problem is common when source databases show different levels of normalization and also occurs in nonrelational sources.
- A single field in one database, such as *Name,* could correspond to multiple fields, such as *First Name* and *Last Name,* in another.

Data transformation (sometimes called instance matching) is a second step in which data in matching fields must be translated into a common format. Frequent reasons for mismatched data include data format (such as *1.6.2004* vs. *6/1/2004*), numeric precision (*3.5kg* vs. *3.51kg*), abbreviations (*Corp.* vs. *Corporation*), or linguistic differences (e.g., using different synonyms for the same concept across databases).

Today's databases are large both in the number of records stored and in the number of fields (dimensions) for each datum object. Database integration or migration projects often deal with hundreds of tables and thousands of fields (Dasu, Johnson, Muthukrishnan, & Shkapenyuk, 2002), with some tables having 100 or more fields and/or hundreds of thousands of rows. Methods of improving the efficiency of integration projects, which still rely mostly on manual work (Kang & Naughton, 2003), are critical for the success of this important task.

MAIN THRUST

In this article, I explore the application of data-mining methods to the integration of data sources. Although data transformation tasks can sometimes be performed through data mining, such techniques are most useful in the context of schema matching. Therefore, the following discussion focuses on the use of data mining in schema matching, mentioning data transformation where appropriate.

Schema-Matching Approaches

Two classes of schema-matching solutions exist: schema-only-based matching and instance-based matching (Rahm & Bernstein, 2001).

Schema-only-based matching identifies related database fields by taking only the schema of input databases into account. The matching occurs through linguistic means or through constraint matching. Linguistic matching compares field names, finds similarities in field descriptions (if available), and attempts to match field names to names in a given hierarchy of terms *(ontology).* Constraint matching matches fields based on their domains (data types) or their key properties (primary key, foreign key). In both approaches, the data in the sources are ignored in making decisions on matching. Important projects implementing this approach include ARTEMIS (Castano, de Antonellis, & de Capitani di Vemercati, 2001) and Microsoft's CUPID (Madhavan, Bernstein, & Rahm, 2001).

Instance-based matching takes properties of the data into account as well. A very simple approach is to conclude that two fields are related if their minimum and maximum values and/or their average values are

equal or similar. More sophisticated approaches consider the distribution of values in fields. A strong indicator of a relation between fields is a complete inclusion of the data of one field in another. I take a closer look at this pattern in the following section. Important instance-based matching projects are SemInt (Li & Clifton, 2000) and LSD (Doan, Domingos, & Halevy, 2001).

Some projects explore a combined approach, in which both schema-level and instance-level matching is performed. Halevy and Madhavan (2003) present a *Corpus-based* schema matcher. It attempts to perform schema matching by incorporating known schemas and previous matching results and to improve the matching result by taking such historical information into account.

Data-mining approaches are most useful in the context of instance-based matching. However, some mining-related techniques, such as graph matching, are employed in schema-only-based matching as well.

Instance-Based Matching through Inclusion Dependency Mining

An *inclusion dependency* is a pattern between two databases, stating that the values in a field (or set of fields) in one database form a subset of the values in some field (or set of fields) in another database. Such subsets are relevant to data integration for two reasons. First, fields that stand in an inclusion dependency to one another might represent related data. Second, knowledge of foreign keys is essential in successful schema matching. Because a foreign key is necessarily a subset of the corresponding key in another table, foreign keys can be discovered through inclusion dependency discovery.

The discovery of inclusion dependencies is a very complex process. In fact, the problem is in general NP-hard as a function of the number of fields in the largest inclusion dependency between two tables. However, a number of practical algorithms have been published.

De Marchi, Lopes, and Petit (2002) present an algorithm that adopts the idea of *levelwise discovery* used in the famous Apriori algorithm for association rule mining. Inclusion dependencies are discovered by first comparing single fields with one another and then combining matches into pairs of fields, continuing the process through triples, then 4-sets of fields, and so on. However, due to the exponential growth in the number of inclusion dependencies in larger tables, this approach does not scale beyond inclusion dependencies with a size of about eight fields.

A more recent algorithm (Koeller & Rundensteiner, 2003) takes a graph-theoretic approach. It avoids enumerating all inclusion dependencies between two tables and finds candidates for only the largest inclusion dependencies by mapping the discovery problem to a problem of discovering patterns (specifically cliques) in graphs. This approach is able to discover inclusion dependencies with several dozens of attributes in tables with tens of thousands of rows. Both algorithms rely on the antimonotonic property of the inclusion dependency discovery problem. This property is also used in association rule mining and states that patterns of size k can only exist in the solution of the problem if certain patterns of sizes smaller than k exist as well. Therefore, it is meaningful to first discover small patterns (e.g., single-attribute inclusion dependency) and use this information to restrict the search space for larger patterns.

Instance-Based Matching in the Presence of Data Mismatches

Inclusion dependency discovery captures only part of the problem of schema matching, because only *exact* matches are found. If attributes across two relations are not exact subsets of each other (e.g., due to entry errors), then data mismatches requiring data transformation, or partially overlapping data sets, it becomes more difficult to perform data-driven mining-based discovery. Both false negatives and false positives are possible. For example, matching fields might not be discovered due to different encoding schemes (e.g., use of a numeric identifier in one table, where text is used to denote the same values in another table). On the other hand, purely data-driven discovery relies on the assumption that semantically related values are also syntactically equal. Consequently, fields that are discovered by a mining algorithm to be matching might not be semantically related.

Data Mining by Using Database Statistics

The problem of false negatives in mining for schema matching can be addressed by more sophisticated mining approaches. If it is known which attributes across two relations relate to one another, *data transformation* solutions can be used. However, automatic discovery of matching attributes is also possible, usually through the evaluation of statistical patterns in the data sources. In the classification of Kang and Naughton (2003), interpreted matching uses artificial intelligence techniques, such as Bayesian classification or neural networks, to establish hypotheses about related attributes. In the uninterpreted matching approach, statistical features, such as the unique value count of an attribute or its frequency distribution, are taken into consideration. The underlying assumption is that two