Mining Entity-Identification Rules for Database Integration

M. Ganesh and **Jaideep Srivastava**
Dept. of Computer Science
4-192 EECS Bldg., 200 Union St. SE
University of Minnesota, Minneapolis, MN 55455

Travis Richardson
Apertus Technologies, Inc.
7275 Flying Cloud Dr.
Eden Prairie, MN55344

Abstract

Entity identification (EI) is the identification and integration of all records which represent the same real-world entity, and is an important task in database integration process. When a common identification mechanism for similar records across heterogeneous databases is not readily available, EI is performed by examining the relationships between various attribute values among the records. We propose the use of distances between attribute values as a measure of similarity between the records they represent. Record-matching conditions for EI can then be expressed as constraints on the attribute distances. We show how knowledge discovery techniques can be used to automatically derive these conditions (expressed as decision trees) directly from the data, using a distance-based framework.

Introduction

Many large enterprises are currently faced with the need to integrate several heterogeneous and independently evolved operational data sources (Drew *et al.* 1993). The information infrastructure in such organizations has been developed over several years using a wide range of technologies from file systems to relational and object database management systems[1]. Effective utilization of available resources in such environments is critically dependent on obtaining a consistent and global view of all information in the organization.

Database integration is the process of merging together information represented in more than one component database system, so that a uniform view of the enterprise data is offered to all the users. This activity forms the core of several frequently performed information systems tasks such as database conversion, database synchronization, data/event replication, and data warehousing (Richardson & Srivastava 1995; Inmon 1992). Federated database systems (Sheth & Larson 1990) allow multiple heterogeneous databases to cooperatively provide a uniform integrated schema to the users while still retaining autonomy of local operations. All these paradigms of database integration require the knowledge about how data in different component databases relate to each other.

Database integration process includes two distinct tasks: *Global schema mapping* or *schema integration* resolves all conflicts due to schema mismatches such as homonyms (using same name for different attributes) and synonyms (using different names for the same attributes) (Batini, Lenzerini, & Navathe 1986). Mapping the records in the component databases to a uniform global schema makes it possible to treat all the records uniformly for further processing (Richardson & Srivastava 1995). When the component databases replicate data the mapped global database may contain multiple instances of the same real-world entity. Identification and integration of these instances is the second task in database integration. In this paper we focus on this *Entity Identification (EI)* (Lim *et al.* 1993) problem assuming the schema integration tasks have already been performed. EI problem deserves attention in the current environment where enterprises rely on robust information systems which can provide very accurate information. In many situations there are no common identification mechanisms, such as key attributes, to distinguish records instances which represent the same real-world entity. In this paper, we propose a method to derive the rules for EI directly from examples of similar instances of records, instead of requiring users to specify rules/conditions that identify instances of the same entity. The concept of attribute-value distances is introduced to facilitate this discovery. Since the rules are learned directly from the data, they are expected to be more comprehensive than could be specified by any single user.

[1] In this paper we use the term "database" to refer to a wide variety of data sources including relational and non-relational databases, file systems, etc.

Employee

Name	Address	City	Zip	State	Age	ID	TelNum	Salary
Johns Smith	935 Shady Oak	Fridley	55532	MN	28	333444555	421-5533	25000

Student

Name	Street	City	Zipcode	Birthdate	ID	Home Ph	Wagerate
John Smith	729 W. 17th	Fridley	55536	052266	1314156	421-5533	7.95

Mapped global table - Personnel

Name	Address	City	Zip	State	Age	TelNum	Wagerate
Johns Smith	935 Shady Oak	Fridley	55532	MN	28	421-5533	12.02
John Smith	729 W. 17th	Fridley	55536	MN	29	421-5533	7.95

Integrated global table - Personnel

Name	Address	City	Zip	State	Age	TelNum	Wagerate
John Smith	729 W. 17th	Fridley	55536	MN	29	421-5533	19.97

Figure 1: Database integration example

Entity Identification Framework

Figure 1 describes a complete example of the integration steps. Two tables, *Employee* and *Student*, from two different databases are integrated to obtain a single table *Personnel*. The key attributes in the two tables *ID* are homonyms, i.e. their meanings differ even though the names are identical. The *ID* attribute in the *Employee* table refers to social security number whereas the *ID* in the *Student* table corresponds to an university Id. There are no common keys between the two tables. After resolving schema conflicts such as name mismatches between attributes, a mapped global table *Personnel* is created. Entity identification is then performed on the mapped global table to determine entities that possibly occur more than once in this table. This step is required since the component database tables have no common key. The rule used to identify instances of the same entity is given as:

Match any 2 of (*Name, Address, TelNum*)

Two record instances in the mapped global table with names "Johns Smith" and "John Smith" have the same value in the *"TelNum"* field and have *"Name"* values which are very similar. Functions that determine whether each of the attribute value pairs match or do not match are defined for each attribute field. If the match function for the *"Name"* field reports similar names such as the pair above to be a match then the two record instances will be identified as the same enitity. The resulting integrated global table is shown with only one instance for each matching entity sets. Conflicts among attribute values of the integrated instances are resolved (determining the correct value for the attribute fields if different instances have different values for the same field) using user specified rules.

Rules for EI, specify relationships among attribute values of instances of the same entity. These rules are obtained from users who have the knowledge of the data semantics. In many cases the users are not able to specify the exact relationships between attribute values which relate instances of the same records. It is however possible to provide a set of example records which represent the same entity. The record-matching rules are therefore implicitly present in the classification of the records into distinct entities. We describe a framework for learning these rules using the ideas from clustering and classification algorithms (Agrawal, Imielinski, & Swami 1993; Han, Cai, & Cercone 1992; Quinlan 1993). The rules we learn are in the form of decision trees, although any classification system which could explain the learned rules to the users may be used in its place instead.

Figure 2 shows our framework for performing entity identification. A data analyst selects a few records from the mapped global database and labels all instances that represent the same entity with a unique entity Id. This set of records, classified by their entity Id, form a training sample for a learning system. The learning system discovers the attribute value relationships among records which represent the same enitity and also the relationships which classify the records as dissimilar. The relationships thus discovered form the rules which are used on the unclassified records to perform entity identification. It is desirable to have the learning module explain the rules in an understandable form to the data analyst so that s/he can evaluate them. Depending on the results of this evaluation the data analyst can alter the training samples to improve the quality of rules learned. The integrated instances are also monitored by the data analyst for evaluation of the performance of the EI rules.

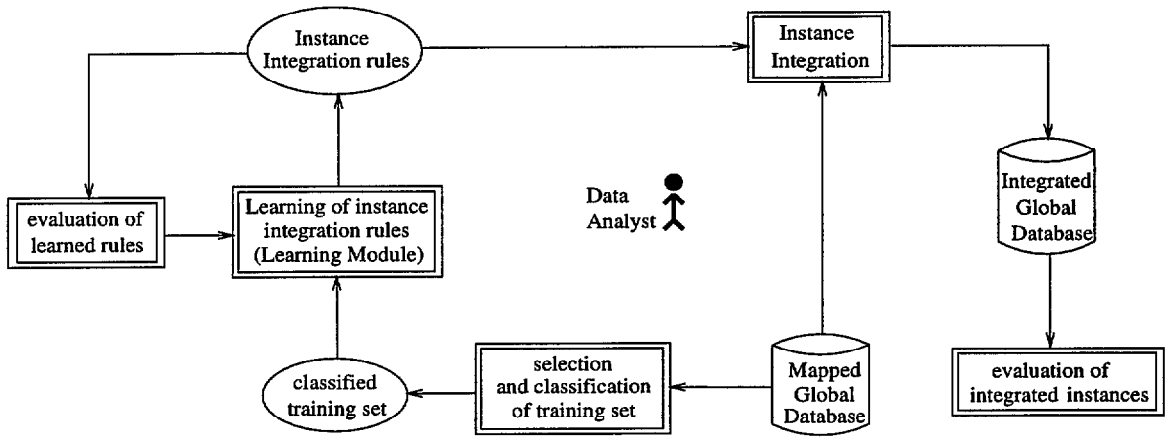

Figure 2: Entity Identification framework

Attribute Distance Functions

EI problem can be viewed as determining the clusters to which the individual records belong, where each cluster represents a real-world entity. We use an initial set of records whose entity identities (EIDs) are known as the training set. The record-matching conditions for any pair of records is an appropriate combination of the relationships between each pair of attribute values in the records. The relationships between similar attribute value pairs is then measured as a distance between their values. A similar pair of attribute values will have a smaller distance between them, compared to a dissimilar pair of values which fall into distant regions of the description space. We measure distances between all such pairs of attribute values to form a vector of distances for a given pair of data records. These distance records are categorized into two classes – "Match" and "No-Match" – depending on whether they measure the distances between two entities with similar EIDs or not. These distance records then form the training set for a learning system which will induce the relationships between attribute distances as the conditions for the source records to match. Entity identification problem is thus reduced to a classification problem and we can use any classification system to learn the rules for EI.

Figure 3 explains the details of the learning module. The training data for the rule induction process is generated from a set of classified records. Using a set of N such records we generate $\frac{N(N-1)}{2}$ distance records, comparing each record against another. Each type of attribute distances is measured using functions which are specific to the attribute type. e.g. distances between two strings may be measured using the edit distance, and distance between two names may be measured using the soundex function (Knuth 1973). These distance records are then assigned the label "Match" if the corresponding classified records have the same EIDs, and the label "No-Match" otherwise. Using these distance records as a training sample, a rule learning system induces the conditions under which a pair of records are similar. These conditions are then applied to any pair of records in the same mapped global database to perform EI.

Experimental Evaluation and Conclusions

We have carried out preliminary experiments to evaluate the effectiveness of our approach to EI in database integration. The experiment used a set of 1100 homogeneized data records from a business customer database. Each of these records in the database represents one of 20 real-world entities. The records were labelled with corresponding EIDs and the number of records corresponding to each of the entities ranged from 6 to a maximum of 524. We measured the effectiveness of the EI rules learned by their performance on unseen test cases and the size of the decision trees. From the data set various sizes of workloads were obtained (50, 100, and 200) and training sets of various sizes are drawn from the corresponding workloads. In our experiments we have varied the training set sizes as 10, 20, 40, 60, 80, and 100% of the test sets. These training sets are used to generate the decision trees for classifying the distance records into one of the two categories, Match or No-Match. The well-known C4.5 algorithm (Quinlan 1993) was used as the rule learning engine. Decision trees learned were then used to classify the corresponding test sets from which the training sets were drawn.

This method was able to achieve very high effectiveness for EI. In most cases the pairwise match errors

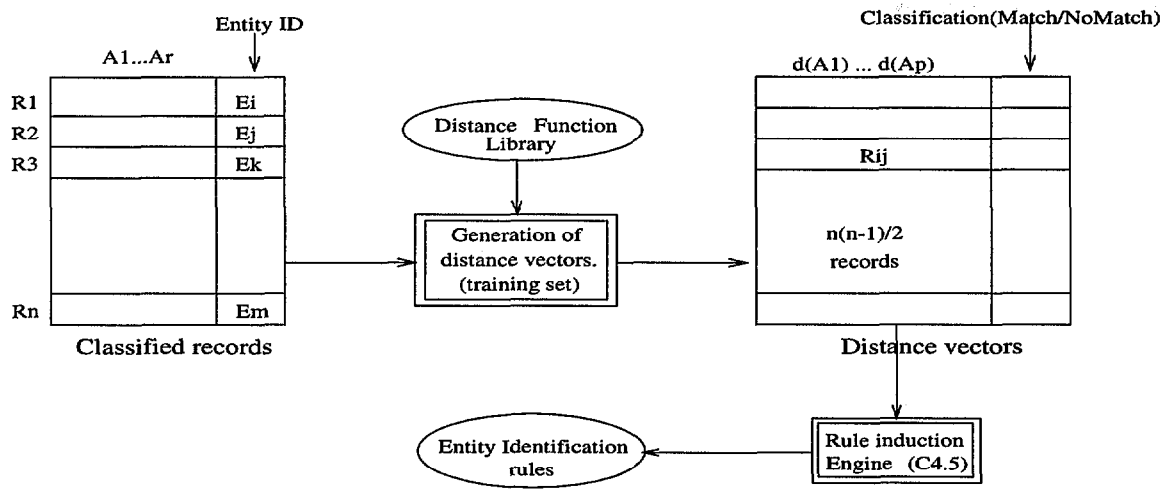

Figure 3: Learning of EI rules

are $\leq 0.2\%$, and the number of records misclassified were $\leq 1\%$. Reasonably small error rates are achieved with about 20% of the data set provided as training set sizes. As we go beyond 40% of the data set as the training set size, the tree sizes increase while bringing only marginal improvement in the classification error rate. At this stage the rules learned have become sensitive to the specific data provided as examples. The size of the decision trees in our experiments varied between 3 and 20 nodes.

We have introduced a framework for the mining of the EI rules directly from examples of integrated instances of entities, to obtain precise rules. Direct application of learning algorithms to this problem is not viable because of the inability to provide examples of all possible entities which may occur in a data set. The concept of attribute-distance functions have been introduced to facilitate this process. Results obtained from our experiments demonstrate that this approach to EI is comparable to the best accuracies of methods where users specify the EI rules without the corresponding human effort. In future work we plan to include attribute-value conflict resolution part of EI. We are currently developing efficient algorithms for performing EI with the distance-based approach and using the distance values computed from the induced rules.

References

Agrawal, R.; Imielinski, T.; and Swami, A. 1993. Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engg.* 5(6):914–925.

Batini, C.; Lenzerini, M.; and Navathe, S. B. 1986. A comparitive analysis of methodologies for database schema integration. *ACM Computing Surveys* 18(4):323–364.

Drew, P.; King, R.; McLeod, D.; Rusinkiewicz, M.; and Silberschatz, A. 1993. Report of the workshop on semantic heterogeneity and interoperation in multidatabase systems. *SIGMOD Record* 22(3):47–56.

Han, J.; Cai, Y.; and Cercone, N. 1992. Knowledge discovery in databases: An attribute-oriented approach. In *Proc. of the 18th VLDB Conference*, 547–559.

Inmon, W. H. 1992. *Building the Data Warehouse.* Wiley-QED.

Knuth, D. E. 1973. *The Art of Computer Programming; Vol 3: Sorting and Searching.* Reading, MA: Addison-Wesley.

Lim, E.-P.; Srivastava, J.; Prabhakar, S.; and Richardson, J. 1993. Entity identification in database integration. In *Proc. of the 9th Int'l Conf. on Data Engg.*, 294–301.

Quinlan, J. R. 1993. *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Richardson, T., and Srivastava, J. 1995. Enterprise/integrator: Using object technology for data integration. In *Proc. of Workshop on Legacy Systems and Object Technology, at OOPSLA 95.*

Sheth, A. P., and Larson, J. A. 1990. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys* 22(3):183–236.