December 2001

Clustering Database Objects for Semantic Integration of Heterogeneous Databases

Huimin Zhao
University of Arizona

Sudha Ram
University of Arizona

Clustering Database Objects for Semantic Integration of Heterogeneous Databases

Huimin Zhao
Department of Management
Information Systems
University of Arizona
hzhao@bpa.arizona.edu

Sudha Ram
Department of Management
Information Systems
University of Arizona
ram@bpa.arizona.edu

Abstract

Interschema Relationship Identification (IRI), i.e., determining the relationships between objects in heterogeneous database schemas, is critical to both the classical schema integration problem and the data cleansing and consolidation phase that precedes data warehouse development. In this paper we propose a cluster analysis-based approach to semi-automate the IRI process, which is typically very time-consuming and requires extensive human interaction. We apply multiple clustering techniques, including K-means, hierarchical clustering, and Self-Organizing Map (SOM), to identify similar database objects from heterogeneous databases based on a combination of features such as object names, documentation, schematic information, data contents, and usage patterns. Initial experimental results indicate that our approach performs better than existing approaches in the accuracy of identified interschema relationships. In addition, a prototype system we have developed provides users a visualization tool for the display of clustering results as well as for the incremental evaluation of candidate solutions.

Keywords: Heterogeneous databases, schema integration, interschema relationship identification, cluster analysis, self-organizing map

Introduction

Modern organizations often rely on many diverse databases to accomplish their daily business operations. However, an integrated data source is prerequisite for decision support applications, such as OLAP and data mining, which require simultaneous and transparent access to data from heterogeneous databases. This integrated data source may take the form of a logical view (e.g., in a federated database system) or a physical consolidated data repository (e.g., in a data warehouse).

Interschema Relationship Identification (IRI), i.e., determining the relationships between objects in different database schemas (Ram and Ramesh 1999), is a major bottleneck in generating such integrated data sources. Critical to both the traditional schema integration problem (Batini et al. 1986, Ram and Ramesh 1999) and the data cleansing and consolidation phase that precedes data warehouse development, the IRI process is typically very time-consuming and requires extensive human interaction. For example, the MITRE Corporation has performed an integration of several database systems for the U.S. Air Force over a period of several years and found that tremendous effort is required to determine interschema relationships (Clifton et al. 1997).

Several approaches have been proposed to automate the IRI process. One approach computes similarity coefficients between database objects using empirical formulas (Hayne and Ram 1990, Palopoli et al. 1999). Another approach clusters entities of multiple databases using SAS clustering procedures (Srinivasan et al. 1995). Neural network-based clustering and classification techniques have also been proposed and evaluated (Clifton et al. 1997, Ellmer et al. 1995). Past approaches have considered input features such as object names (Ellmer et al. 1995, Hayne and Ram 1990, Palopoli et al. 1999), documentation (Clifton et al. 1997), schematic information (Clifton et al. 1997, Ellmer et al. 1995, Srinivasan et al. 1995), data contents (Clifton et al. 1997), and usage patterns (Srinivasan et al. 1995).

In this paper we report our research on applying various clustering techniques to identify interschema relationships. A key differentiating feature of our approach is that we incorporate in our analysis all types of database objects (attributes, entities, and relationships) and multiple types of input features. We report our initial findings from experiments with various clustering techniques, including K-means, hierarchical clustering, and Self-Organizing Map (SOM). In addition, we discuss the importance of visualization and incremental user evaluation, enabled by a prototype system we have developed. The rest of the paper is organized as follows. Section 2 briefly reviews cluster analysis. Section 3 presents a classification of the semantic information associated with database objects that might be available for cluster analysis. Section 4 presents an experimental evaluation of the proposed approach. Section 5 concludes the paper with a discussion of future work.

Cluster Analysis

Cluster analysis is the technique of grouping similar objects into unknown groups. K-means and hierarchical clustering are two widely used clustering methods and are available in many statistical packages, including SAS, SPSS, SYSTAT and BMDP.

Kohonen's SOM, an unsupervised neural network, has received much attention as an alternative to traditional clustering techniques recently (Kohonen 1995). SOM usually projects multi-dimensional data onto a two-dimensional map. Since the topological relationships among the input data are roughly preserved in the map, SOM can be used for cluster analysis.

In clustering problems, each input case is represented as a vector of features (variables), which may be empirically weighted. Other analyses, such as principal component analysis and factor analysis, can be performed prior to cluster analysis to reduce the dimensionality of the input vectors. All clustering methods require defining some measure of distance (e.g., Euclidean, Mahalanobis, and Cosine) between two cases.

Petersohn (1998) has empirically compared various clustering methods, including K-means, hierarchical clustering, and SOM. In general, no method has been found to be the best for all cases due to the highly empirical nature of cluster analysis. In our approach, we apply multiple clustering methods to cross-validate results in the context of IRI.

A Classification of Semantic Information about Database Objects

The choice of input features has an obvious impact on the performance of cluster analysis. Both missing relevant features and including noisy ones can lead to performance degradation. We classify the semantic information about database objects that might be used as input features for cluster analysis and discuss related technical issues in this section.

- Name: A general principle in database design is that database objects should be named to reflect their meanings. Some IRI approaches determine correspondences between database objects by comparing their names (Ellmer et al. 1995, Hayne and Ram 1990, Palopoli et al. 1999). However, there are various problems associated with object names: (1) They usually can not capture the semantics completely. (2) Phrases and acronyms rather than single words are more commonly used to name database objects. (3) The meaning of a database object changes as the associated business process evolves.

- Documentation: Database design documents usually contain descriptions of database objects. Sometimes these documents are stored in database dictionaries. An information retrieval tool called DELTA has been used to look for potential attribute correspondences based on descriptions about attributes (Clifton 1997). However, this information is often incomplete, incorrect, ambiguous, or simply not available in real world practices.

- Schematic information: Database objects representing similar real-world concepts should be modeled similarly and therefore should have similar structures (Clifton et al. 1997, Ellmer et al. 1995, Srinivasan et al. 1995). Schematic information, such as data types, length, and constraints, are usually stored in the system catalog of a DBMS. However, semantically similar concepts could often be modeled using different structures while semantically different concepts could have similar structures.

- Data content: Semantics are also embedded in the actual data stored in databases. SemInt (Clifton et al. 1997) uses population statistics, such as statistics on numeric values for numeric attributes and statistics on string lengths for character attributes. However, since the same attribute can often be defined using different data types, we posit using the same set of features (e.g., statistics on number of bytes) for different data types.

- Usage pattern: Usage patterns, such as update frequency and number of users or user groups, has been considered in clustering entities based on the assumption that the same entity should be accessed in similar manners in different systems (Srinivasan et al. 1995). Usage data are kept in the audit trail of modern DBMSs but may not be available in legacy systems.

- Business rule: Database schemas can not directly represent many complex business rules, which are often implemented using assertions, procedures, triggers, and application programs. However, in general, semantics embedded in codes are hard to extract.

- Users' mind and business process: Semantics that reside in users' minds or business processes can only be explored via interaction with users themselves.

From the above discussion, we observe: (1) Completely automating the IRI process is generally infeasible. Human intervention is necessary to capture the last two, and arguably the most important, categories of information. (2) Unlike in some other clustering problems where there are features that naturally discriminate input cases, no optimal set of features seems to exist for IRI due to the problems stated earlier. Appropriate features must be evaluated and selected in each particular case. (3) While database names and documents directly describe the meanings of database objects, schematic information, data contents, and usage patterns only indirectly reflect the semantics. We posit that direct semantic features are more important than indirect ones in semantic clustering. When it is infeasible to extract direct semantic features in some real-world hard cases, the performance of cluster analysis will inevitably degenerate. In our approach, we incorporate all available semantic information to achieve the best possible clustering results.

Experimental Analysis

We have developed a prototype system called SOM Interactive Clustering (SIC) and experimented on several heterogeneous databases. Due to the space limit, in this section we only describe a simple constructed example, which consists of two university databases. Figure 1 shows the database schemas specified using the Unifying Semantic Model (USM), an extended Entity-Relationship model (Ram et al. 1999).

We incorporate in our analysis of attributes all available features, including similarity between attribute names or owner (entity or relationship) names, schematic information (primary key, data type, length, precision, and not null constraint), and statistics (percentages of null values, digits, and white spaces, number of distinct values, and statistics on number of bytes). We linearly normalize each feature into the range of [0, 1], perform principle component analysis to reduce the input dimensionality from 49 to 14, and then use K-means, hierarchical clustering, and SIC to cluster attributes on the 14 components. Due to the space limit, we only display the results generated by SIC (Figures 2 and 3).

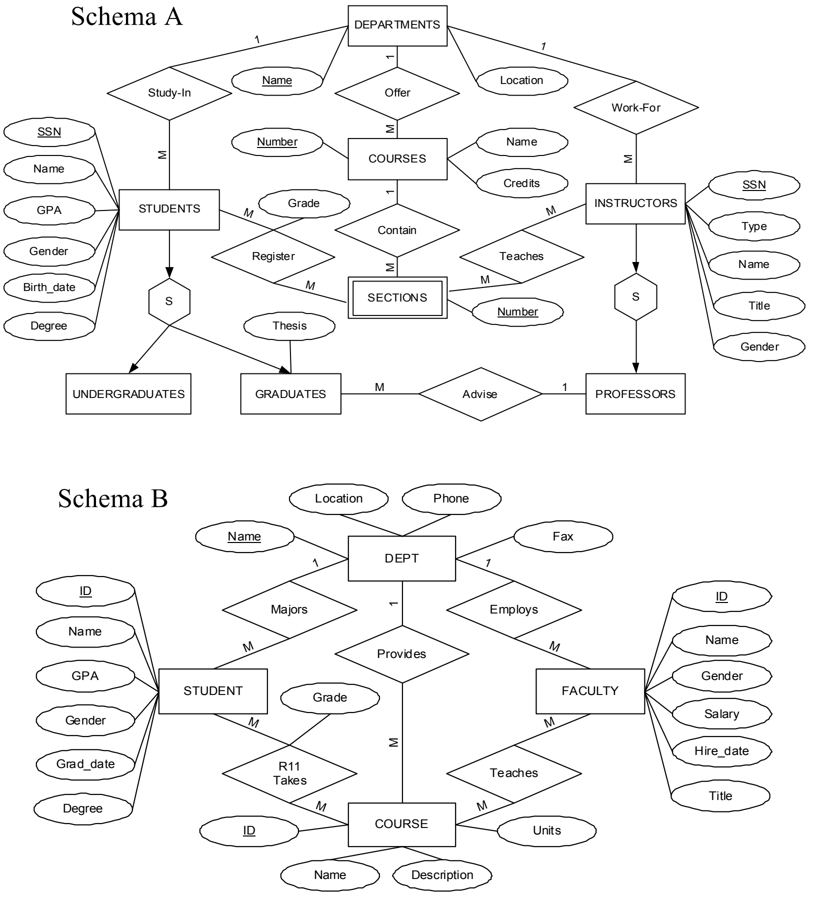
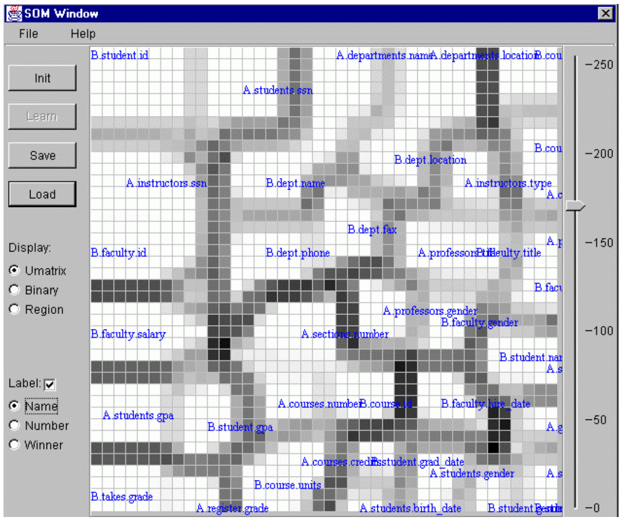


Figure 1. USM Schemas of Two University Databases

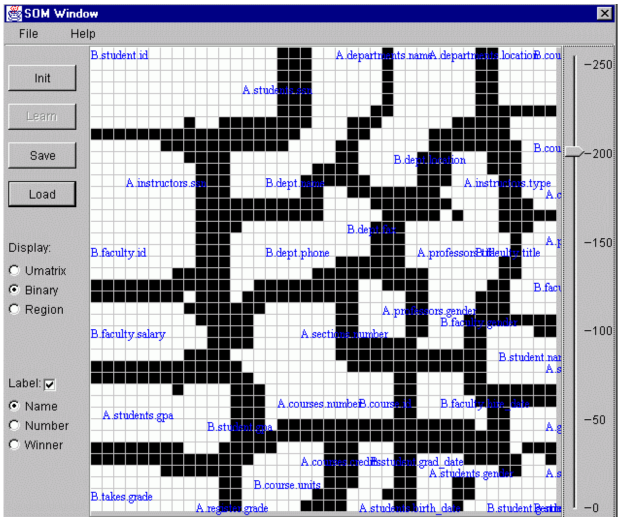
The results indicate that all three clustering methods achieve similar clustering accuracy. However, when K is larger than 10, K-means cannot correctly group attributes A.INSTRUCTORS.Gender (i.e., attribute Gender of entity INSTRUCTORS in database A) and B.FACULTY.Gender because the



Display:
● Umatrix
○ Binary
○ Region

Label: ☑
● Name
○ Number
○ Winner

Figure 2. An Attribute Map



Display:
○ Umatrix
● Binary
○ Region

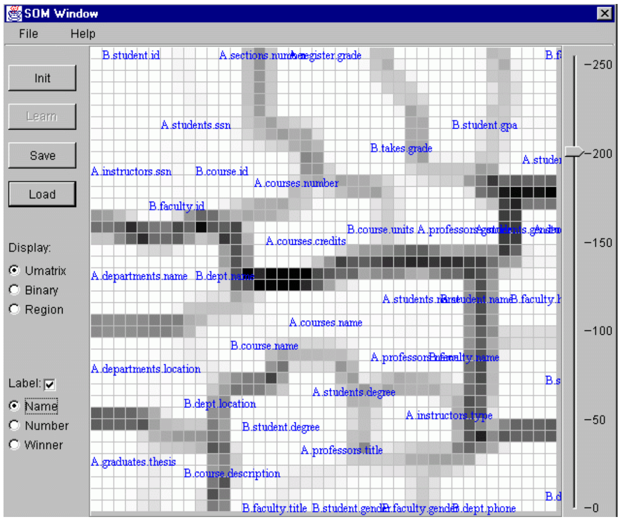
Label: ☑
● Name
○ Number
○ Winner

Figure 3. Binary Mode of Figure 2

former is character (M/F) and the latter numeric (1/2). Similarly, hierarchical clustering groups these two attributes only at relatively high levels. On the other hand, SIC visualizes the degree of similarity between these two attributes more appropriately. The pair A.STUDENTS.Gender with B.STUDENT.Gender exemplifies the stated problem.

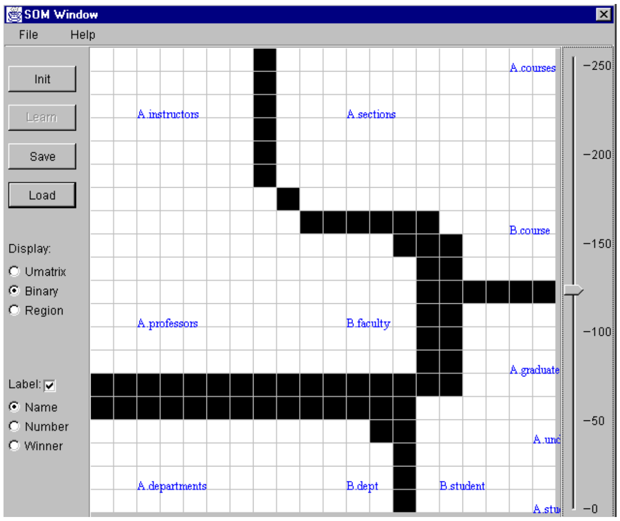
To compare our approach with existing ones, we have conducted a series of experiments incorporating different sets of features that have been used by existing approaches. For example, in one experiment we use only schematic information and statistics on data contents, similar to those used by SemInt (Clifton et al. 1997). The clusters reflect structural rather than semantic similarity (Figure 4).

Many semantically different attributes with similar structures and data patterns cannot be separated due to the limited input features, while semantically similar attributes defined using different data types can be missed. All date attributes (A.STUDENTS.Birth_Date, B.STUDENT.GRAD_DATE, B.FACULTY.Hire_Date) are clustered as similar. A.STUDENTS.Name, A.INSTRUCTORS.Name, and A.COURSES.Name are clustered together. A.INSTRUCTORS.Gender and B.FACULTY.Gender are clustered as very different. SemInt (Clifton et al. 1997) has encountered similar problems. When used in a real database integration project for the US Air Force, SemInt generated relatively big clusters (the average size is 30).



Display:
● Umatrix
○ Binary
○ Region

Label: ☑
● Name
○ Number
○ Winner



Display:
○ Umatrix
● Binary
○ Region

Label: ☑
● Name
○ Number
○ Winner

Figure 4. An Attribute Map (Using Schematic

Figure 5. An Entity Map

In cluster analysis we are concerned with two types of errors, grouping of dissimilar objects (Type II) and failing to group similar objects (Type I). While Type II errors can be corrected in follow-up human evaluation, Type I errors are more dangerous for IRI. In general, given a data set, decreasing the number of clusters reduces Type I errors, but increases Type II errors and therefore the amount of human effort. In the worst case, all database objects are manually evaluated to avoid potential type I errors. By including naming information, we reduce the number of incorrect groupings of dissimilar attributes (Type II error) from 14 to 4 and the number of missed similar attribute pairs (Type I error) from 2 to 0 at the distance threshold shown in Figure 3.

The same set of techniques can be used to cluster entities and relationships too. In our experiments, the input features of entities include similarity between entity names, percentage of similar attributes, and numbers of attributes, relationships, and instances. The input features of relationships include similarity between relationship names, similarity between participating entities or attributes, cardinality, and numbers of attributes and instances. Figure 5 shows an entity map generated by SIC.

While we have not found significant differences in the accuracy of results between the three methods, SIC is intuitively better than the other two in terms of the visualization of results. We use the U-matrix method (Costa and de Andrade 1999) to present results of SIC. On a map consisting of output nodes, each input case corresponds to a best-matched node called "response". The responses of similar input cases are located close to each other. Gray levels indicate relative distances between neighboring output nodes and therefore boundaries between clusters.

A hierarchical clustering result is desired in that it enables users to start from very similar objects and incrementally evaluate less similar objects. Though K-means is a nonhierarchical method, it is possible to manually generate a hierarchical result by running K-means several times with different numbers (Ks) of clusters. Using our SIC, users can vary the distance threshold on a slider and obtain clustering results on different distance levels interactively. The lower the threshold, the tighter the clusters. SIC provides users a visualization tool for the display of clustering results as well as for the incremental evaluation of candidate solutions.

Conclusion and Future Work

We have described a cluster analysis-based approach to semi-automate the IRI process and presented some initial experimental findings. We argue that no optimal set of features exists for IRI and therefore feature evaluation and selection must be performed depending on particular applications. We use a uniform set of techniques to cluster all types of database objects (attributes, entities, and relationships) and incorporate a more complete set of semantic information than past approaches. While our initial experiments indicate that SIC and other methods such as K-means and hierarchical clustering achieve similar accuracy, SIC provides additional benefits of offering visualization and incremental evaluation.

Our experiments indicate that using direct semantic indicators such as names of database objects are more important than indirect semantic indicators such as those used by SemInt (Clifton 1997). However, in real-world heterogeneous databases, comparison of names is not always feasible due to the problems we have discussed. In such cases the quality of the semantic clustering can degenerate seriously. The integrator must bear in mind that, although useful in reducing the amount of human interaction, semantic clustering can only provide limited support and should not replace careful human evaluation.

Another related problem we are currently researching is the identification of instance level correspondences. We construct attribute-matching operators based on the results of IRI (i.e., schema level correspondences) and use machine learning techniques to learn the rules to detect duplicate entities from heterogeneous databases. We are in the process of integrating the two phases into a complete database integration system and validating its utility on several databases of a large public university's warehouse operations.

References

Batini, C., Lenzerini, M., and Navathe, S. B., "A Comparative Analysis of Methodologies for Database Schema Integration." *ACM Computing Surveys* (18:4), 1986, pp. 323 - 64.

Clifton, C., Housman, E., and Rosenthal, A., "Experience with a Combined Approach to Attribute-Matching Across Heterogeneous Databases," in *Data Mining and Reverse Engineering (Proc. of DS-7)*, 1997, pp. 429-51.

Costa, J. A. F., and de Andrade, N. M. L., "Cluster Analysis Using Self-Organizing Maps and Image Processing Techniques," in *IEEE SMC'99 Conference Proceedings*, Vol. 5, 1999, pp. 367-72.

Ellmer, E., Huemer, C., Merkl, D., and Pernul, G., "Neural Network Technology to Support View Integration," in *Proc. of the 14th Int. Conf. on Object-Oriented & Entity Relationship Modeling (OOER'95)*, 1995, pp. 181-90.

Hayne, S., and Ram, S., "Multi-User View Integration System (MUVIS): An Expert System for View Integration," in *Proc. of the Sixth International Conference on Data Engineering*, 1990, pp. 402-10.

Kohonen, T., *Self-Organizing Maps*, Berlin: Springer, 1995.

Palopoli, L., Sacca, D., and Ursino, D., "Semi-Automatic Techniques for Deriving Interscheme Properties from Database Schemes," *Data & Knowledge Engineering*, (30:3), 1999, pp. 239-73.

Petersohn, H., "Assessment of Cluster Analysis and Self-Organizing Maps," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, (6:2), 1998, pp. 136-49.

Ram, S., Park, J., and Ball, G., "Semantic Model Support for Geographic Information Systems," *IEEE Computer*, (32:5), 1999, pp. 74-81.

Ram, S., and Ramesh, V., "Schema Integration: Past, Present and Future," in *Management of Heterogeneous and Autonomous Database Systems*, San Francisco: Morgan Kaufmann, 1999, pp. 119-56.

Srinivasan, U., Ngu, A., and Gedeon T., "Concept Clustering for Cooperation in Heterogeneous Information Systems," in *Database Applications Semantics (Proc. of DS-6)*, 1995, pp. 481-99.