December 2002

APPLYING CLASSIFICATION TECHNIQUES IN SEMANTIC INTEGRATION OF HETEROGENEOUS DATA SOURCES

Huimin Zhao
University of Wisconsin, Milwaukee

Sudha Ram
University of Arizona

APPLYING CLASSIFICATION TECHNIQUES IN SEMANTIC INTEGRATION OF HETEROGENEOUS DATA SOURCES

Huimin Zhao
School of Business Administration
University of Wisconsin, Milwaukee

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

Abstract

Entity identification, i.e., detecting semantically corresponding records from heterogeneous data sources, is a critical step in integrating the data sources. We propose a classification-based approach for entity identification. We apply multiple classification techniques drawn from statistical pattern recognition, machine learning, and artificial neural network to determine whether two records from different data sources correspond to the same real-world entity. We conduct experiments to evaluate the performance of different techniques. In this paper, we review several widely used classification techniques and attribute-matching functions and report some empirical results to demonstrate the utility of our approach.

Keywords: Database integration, entity identification, classification

Introduction

The need to integrate heterogeneous data sources is ubiquitous. Legacy databases developed over time in different sections of an organization need to be integrated for strategic purposes. Business mergers and acquisitions force the information systems previously owned by different institutions to be merged. Information needs to be shared or exchanged across the boundaries of cooperating enterprises. The rapid growth of the Internet creates new requirements for data integration.

To integrate a collection of heterogeneous data sources, either logically (e.g., using a mediator/wrapper architecture) or physically (e.g., building a data warehouse), a critical step is to identify the semantically corresponding records from the data sources. This problem has been called entity identification (Ganesh, et al. 1996), approximate record matching (Verykios, et al. 2000), merge/purge (Hernandez and Stolfo 1998), and record linkage (Fellegi and Sunter 1969; Winkler 1997). It has been shown to be a very complex and time-consuming task due to dirty data and semantic heterogeneities among different data sources. Winkler reported that the integration of several mailing lists for the U.S. Census of Agriculture in 1992 consumed thousands of person-hours even though an automated matching tool was used (Winkler 1997).

In this paper, we propose a classification-based approach for entity identification. We apply multiple classification techniques drawn from statistical pattern recognition, machine learning, and neural network to determine whether two records from different data sources correspond to the same real-world entity. We conduct experiments to evaluate the performance of different techniques. We have empirically evaluated our approach using real-world data and will report some experimental results in this paper.

Classification Techniques

Given a pair of records from semantically corresponding tables in two databases, we need to determine whether they represent the same real-world entity. This is a two-class (match or non-match) classification problem. Each pair of records to be compared is described in terms of a vector of features, $x = x_1, x_2, \ldots, x_m$). Each feature, x_1, is usually a distance or similarity measure for

comparing the values of the two records on semantically corresponding attributes. The objective is to assign the record pair to one of two classes, match (M) and non-match (N), based on the features.

A classification technique learns a general rule, called a classifier, from a set of sample record pairs, whose true classes are known. The learned classifier can then be used to predict the classes of other record pairs. Various classification techniques have been developed in statistical pattern recognition, machine learning, and artificial neural network. Figure 1 shows some widely used classification techniques.

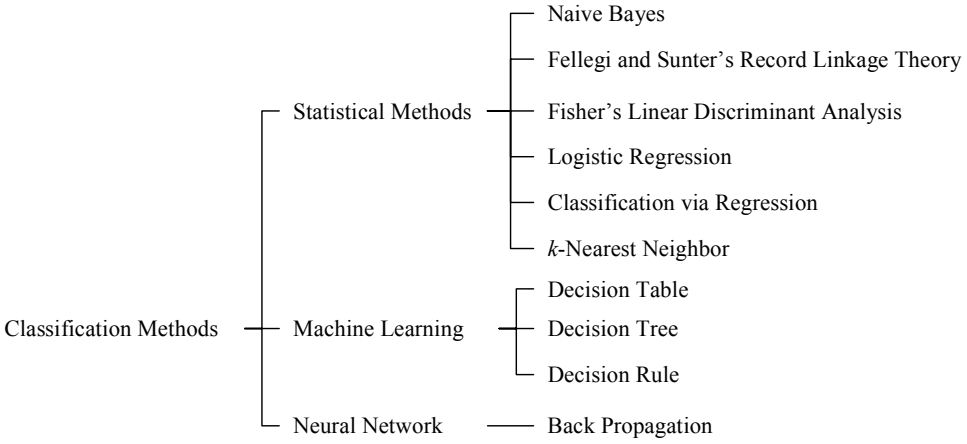

Figure 1. Some Classification Techniques

The Bayes method estimates the odds ratio, $\frac{\Pr(M|\mathbf{x})}{\Pr(N|\mathbf{x})}$, and compares it to a threshold value to determine the class of a record pair. The threshold value is determined by the prior probabilities of the two classes and the relative costs associated with two types of errors, i.e., false matches and false non-matches. The Bayes method provides the intuitively optimal classification. It, however, can seldom be applied directly without making simplifying assumptions because of the explosive combinations of feature values, especially when some of the features are continuous. Naive Bayes, which assumes that the features are conditionally independent, is commonly used instead. Continuous features usually need to be discretized prior to classification. Fellegi and Sunter's record linkage theory (1969) extends the basic Bayes method to maintain acceptable levels of error rates. The decision space is divided into three areas, match, non-match, and unclassified, based on two thresholds. The unclassified record pairs then need to be manually reviewed.

Fisher's linear discriminant analysis (LDA) is one of the most widely used statistical classification methods. LDA uses a line (in a two-dimensional feature space) or hyper-plane to separate the two classes. The coefficients in the linear model are chosen in such a way to separate the two classes as much as possible, assuming that the features follow normal distributions. Logistic regression assumes that the logarithm of the odds ratio of match to non-match is linearly related to the features; the decision boundary between the two classes is still linear in the original feature space. Logistic regression makes no assumption about the distributions of the features and has an advantage over LDA when many features are categorical. In practice, however, logistic regression and LDA often produce very similar results. Regression techniques, such as linear regression, can also be used for classification. A regression model can be derived from the training data to predict the class using the features. *k*-nearest neighbor techniques simply memorize the training data and classify each new case into the majority class of the *k* cases in the training data that are closest to the new case.

Machine learning techniques generate decision tables, trees, or rules that are easily understood and are most compatible with human reasoning. A decision table learner selects several most discriminating features to form a lookup table, which will be used to classify new cases. Decision tree techniques follow a "divide and conquer" strategy and build tree-like sequential decision models. C4.5 is one of the most widely used decision tree techniques (Quinlan 1993). A decision tree can be easily translated into a set of decision rules; each leaf node of the tree corresponds to a rule. There are also rule-induction techniques that can produce more general classification rules. A special form of simple classification rules is 1R (for "1-rule"), which uses a single most discriminating feature to determine the class of a case.

Back propagation is one of the most widely used neural network techniques for classification. Neural networks are highly interconnected networks, which learn by adjusting the weights of the connections between nodes on different layers. A neural network may have an input layer, an output layer, and zero or many hidden layers. Theoretically, a neural network with two or more hidden layers can approximate any function and has the potential to achieve the lowest possible error rates. However, neural networks are "black boxes"; it is hard to interpret the rules they follow in classifying cases. The training of a neural network is often not trivial; it takes experience and experimenting to adjust the parameters, such as the number of nodes on each layer and the learning rate.

Past approaches for entity identification have been based on a particular classification technique, such as record linkage (Fellegi and Sunter 1969; Winkler 1997), logistic regression (Pinheiro and Sun 1998), and decision tree (Ganesh, et al. 1996; Haimowitz, et al. 1997; Tejada, et al. 2001; Verykios, et al. 2000). However, the performance of a classification technique varies in different situations; there is no single technique that is always superior to others in every problem. In our approach for entity identification, we conduct experiments to select the best technique in each particular situation.

Attribute-Matching Functions

The classification of a record pair is based on a vector of features, each of which is usually a distance or similarity measure for comparing the values of the two records on semantically corresponding attributes. Given two (base or derived) attributes whose domains are A_1 and A_2, an attribute-matching function is a mapping, $f: A_1 \times A_2 \rightarrow [0,1]$, which returns a degree of match between every pair of values drawn from A_1 and A_2 , where 1 indicates a perfect match.

If two corresponding attributes in different data sources share the same format and store perfect data, we can compare them using simple equality. However, we frequently observe both schema level and data level discrepancies. Semantically corresponding attributes often have different formats in different databases. There are many kinds of data errors in most real-world data sources. Transformation functions are needed to convert corresponding attributes into compatible formats. Approximate attribute-matching functions are needed to measure the degree of similarity between two attribute values.

There are many approximate string-matching methods. Stephen (1994) reviewed several string distance measures, including Hamming distance, Levenshtein's metrics, longest common substring, and *q*-grams. Budzinsky (1991) compared twenty string comparison methods, which account for spelling and/or phonetic errors.

Numeric attributes measured on interval or ratio scales can be compared using normalized distance functions. Dictionaries, or lookup tables, can be developed to bridge different coding schemes for attributes measured on categorical scales. There are also tools, e.g., Potter's Wheel (Raman and Hellerstein 2001) and the software for name standardization and address standardization developed at the U.S. Bureau of Census (Winkler 1999), that help to clean up and standardize some particular types of attributes.

If multiple matching functions can be used to compare a pair of corresponding attributes, we may need to evaluate the functions and select the most discriminating one for classification, because some classification techniques, e.g., LDA and logistic regression, are sensitive to highly correlated features. We can compute the correlation between each matching function and the class (match or non-match) based on the training examples for classification and select the matching function that is most correlated to the class. There are also heuristics that help us choose appropriate matching functions. For example, Soundex is good for comparing human names; edit distance measures are good for comparing strings with spelling errors. We can also construct arbitrarily complex matching functions by combining multiple matching functions and transformation functions.

Empirical Evaluation

We empirically evaluate our proposed approach using real-world heterogeneous data sources. In this paper we report on some of our experiments with partial book catalogs that we have extracted from two leading online bookstores. The proliferation of the Internet in the recent years has motivated many vendors to setup online storefronts, where potential customers can browse their electronic product catalogs (E-Catalogs) and buy selected product items. The rapid growth of such online stores has resulted in the need for integration of multiple E-catalogs, which belong to different vendors and are typically heterogeneous (Ram and Zhao 2001).

The two catalogs share some common attributes about books, such as ISBN, authors, title, retail price, our price, cover type, number of pages, edition, publisher, publish month, publish year, average rating, and sales rank. There is a common key attribute, i.e., ISBN, between the two catalogs. Identifying corresponding records from these two catalogs is actually trivial. We chose this

case for empirical evaluation because the ISBN could provide us convenient and objective training and testing data to evaluate various classification techniques. We trained classifiers based on other attributes common to the two catalogs while withholding the ISBN, which served as the judge of the correctness of the results. In the integration of general heterogeneous data sources, however, when we do not have such a common key among the different sources, some domain experts must manually match some records to train and test classifiers.

The types of attribute-matching functions we used include: (1) Exact comparison: Some attribute pairs, such as edition, publish month, and publish year, were required to match exactly. (2) Approximate string matching: There were some typographical discrepancies in author names and titles. We used Levenshtein's edit distance metric to compare two strings. (3) Normalized distance: Some attributes, such as list price, our price, and number of pages, of the same book were different, but deemed to be close. We normalized the difference between two numbers into the range of [0,1]. (4) Dictionary: Dictionaries helped to resolve coding differences. Most of the publishers were named differently in the two catalogs. For example, one publisher was named "John Wiley & Sons" in one catalog and "Wiley, John & Sons, Incorporated" in the other. Cover types were sometimes coded differently too. For example, the cover of some books was coded as "Paperback" in one catalog and "Textbook Paperback" in the other. For some attributes, we designed multiple matching functions and selected the most discriminating one via correlation analysis. It turned out that approximate comparison was better for some attribute pairs, including author and title, while exact comparison was better for others, including list price, our price, and number of pages. We implemented the matching functions in Oracle PL/SQL.

We evaluated several classification techniques, including 1R, naive Bayes, logistic regression, classification via linear regression, decision table, J4.8 decision tree (a variant of C4.5), back propagation neural network, and *k*-nearest neighbor, which have been implemented in the Weka system (Witten and Frank 2000). We ran each technique 200 times on a sample data set with 1404 match examples and 1404 non-match examples; every time, 66% of the examples were randomly re-sampled for training while the remaining examples were set aside for testing.

Table 1 summarizes the accuracies and running times of the ten techniques in the experiment. An ANOVA test (Table 2) shows that some of the techniques performed significantly differently in terms of classification accuracy ($F(9, 1990) = 1072.500$, $p < 0.05$). Table 3 shows the result of a Sheffé's post hoc test. Six homogeneous (in terms of accuracy) subsets of techniques are recognized at the level of significance $\alpha = 0.05$. 1R was the least accurate; naive Bayes and back propagation neural network were the most accurate.

The performance of the techniques, however, may vary in different applications. In another experiment, where we compared two databases about property items maintained at two departments of a large public university, J4.8 decision tree, logistic regression, and back propagation neural network performed the best.

The amounts of time consumed by the techniques in training and testing were quite diverse. *k*-Nearest Neighbor techniques do not learn any generalized structure from the training examples and compare every new example to every training example. They were much slower than other techniques in testing. Back propagation neural network was much slower than other techniques in training.

Conclusion and Future Work

We have described a classification-based approach for entity identification and presented some experimental results. Since the performance of various classification techniques varies in different situations, it is necessary to conduct experiments to select the best techniques for each particular application.

We are also evaluating multiple classifier systems and cost-sensitive classification methods. Multiple classifiers can be combined in various ways, such as concatenating, bagging, boosting, and stacking, to improve classification accuracy. Discriminant functions learned by statistical methods, such as LDA and logistic regression, can be used as additional input features to train decision trees or rules. Such a concatenation of multiple classifiers may perform better than each of the base classifiers. Multiple classifiers of the same type can be gathered, via bagging or boosting, into a committee, or ensemble, to "vote" for the results. In bagging, base classifiers are learned independently based on different training data and are given the same weight in the final voting. In boosting, base classifiers are learned sequentially and are weighted differently according to their performance. Stacking trains another classifier, called a meta classifier, to make the final classification decision based on the predictions of multiple base classifiers of different types.

Table 1. Summary of Classification Results

	Accuracy (%)		False Matches (%)		False Non-matches (%)		Training Time (Seconds)		Testing Time (Seconds)	
Method	Mean	StdDev	Mean	StdDev	Mean	StdDev	Mean	StdDev	Mean	StdDev
1R	96.581	0.524	1.650	0.884	5.190	0.970	0.11	0.05	0.02	0.03
Naive Bayes	99.240	0.260	1.248	0.475	0.272	0.197	0.14	0.03	0.08	0.03
Linear Regression	98.315	0.361	0.075	0.101	3.296	0.723	1.57	0.34	0.11	0.10
Logistic Regression	99.088	0.277	0.689	0.398	1.135	0.513	0.41	0.11	0.06	0.08
J4.8	98.871	0.317	1.009	0.593	1.250	0.626	0.67	0.26	0.03	0.06
Decision Table	99.009	0.327	1.047	0.655	0.936	0.570	4.25	0.52	0.06	0.04
Back Propagation	99.306	0.271	0.586	0.356	0.804	0.471	210.45	46.44	0.21	0.08
1-Nearest Neighbor	99.130	0.350	0.880	0.461	0.861	0.563	0.01	0.01	24.70	1.91
3-Nearest Neighbor	98.298	0.391	1.619	0.685	1.786	0.565	0.01	0.00	33.32	3.71
5-Nearest Neighbor	98.326	0.368	1.368	0.549	1.981	0.591	0.01	0.01	33.72	3.07

Table 2 ANOVA. Test of the Accuracies of the Classification Techniques

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1197.955	9	133.106	1072.500	0.000
Within Groups	246.975	1990	0.124		
Total	1444.930	1999			

Table 3. Sheffé's Post Hoc Test: Homogeneous Subsets

	Subset for alpha = .05					
Method	1	2	3	4	5	6
1R	96.581					
3-Nearest Neighbor		98.298				
Linear Regression		98.315				
5-Nearest Neighbor		98.326				
J4.8			98.871			
Decision Table			99.009	99.009		
Logistic Regression				99.088		
1-Nearest Neighbor				99.130	99.130	
Naive Bayes					99.240	99.240
Back Propagation						99.306
Sig.	1.000	1.000	0.084	0.221	0.380	0.941

In the experiment we have described in this paper, we compared different techniques in terms of plain accuracy using a balanced data set. In real-world applications, however, neither are the two classes (i.e., match and non-match) of record pairs balanced, nor are the costs associated with the two types of errors (i.e., false match and false non-match) symmetric. Different weights should be given to the two classes, according to the prior probabilities of the two classes and the relative costs of the two types of errors.

The techniques we have described in this paper are useful for detecting instance-level correspondences across data sources. A related problem is the identification of schema-level correspondences (Zhao and Ram 2001). Techniques for solving the two problems can be incorporated into an iterative procedure, so that correspondences on the two levels can be evaluated incrementally (Ram and Zhao 2001).

These techniques have many potential applications in the semantic integration of heterogeneous data sources. The National Institute of Standards is funding the development of a distributed Master Patient Index (MPI), which is intended to allow care providers to be able to access the medical records of all patients in the U.S. (Bell and Sethi 2001). There is also a need for a massive cooperation of information systems in the national security area. Information about an individual needs to be retrieved from many systems maintained by different organizations around the country, including police departments, motor vehicle departments, and airlines. Many companies may benefit from a consolidated customer database built upon a variety of data sources. We are currently conducting a case study at an application service provider for airlines. Our proposed approach and techniques are used to identify duplicate passengers and duplicate ticket reservations.

References

Bell, G. B. and Sethi, A., "Matching Records in a National Medical Patient Index," *Communications of the ACM* (44:8), 2001, pp. 83 - 88.

Budzinsky, C. D., "Automated Spelling Correction," Statistics Canada, Unpublished Report, 1991.

Fellegi, I. P. and Sunter, A. B., "A Theory of Record Linkage," *Journal of the American Statistical Association* (64:328), 1969, pp. 1183-1210.

Ganesh, M., Srivastava, J., and Richardson, T., "Mining Entity-identification Rules for Database Integration," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, 1996, pp. 291-294.

Haimowitz, I. J., Gur-Ali, O., and Schwarz, H., "Integrating and Mining Distributed Customer Databases," in *Proceedings of KDD-97*, 1997, pp. 179-182.

Hernandez, M. A. and Stolfo, S. J., "Real-world Data Is Dirty: Data Cleansing and the Merge/purge Problem," *Data Mining and Knowledge Discovery* (2:1), 1998, pp. 9-37.

Pinheiro, J. C. and Sun, D. X., "Methods for Linking and Mining Massive Heterogeneous Databases," in *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998, pp. 309-313.

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

Ram, S. and Zhao, H., "Detecting Both Schema-level and Instance-level Correspondences for the Integration of E-Catalogs," in *Proceedings of the Eleventh Annual Workshop on Information Technology and Systems (WITS'01)*, New Orleans, Louisiana, USA, 2001, pp. 193-198.

Raman, V. and Hellerstein, J. M., "Potter's Wheel: An Interactive Data Cleaning System," *The VLDB Journal* 2001, pp. 381-390.

Stephen, G. A., *String Searching Algorithms*: World Scientific Publishing Co. Pte. Ltd., 1994.

Tejada, S., Knoblock, C. A., and Minton, S., "Learning Object Identification Rules, for Information Integration," *Information Systems* (26:8), 2001, pp. 607-633.

Verykios, V. S., Elmagarmid, A. K., and Houstis, E. N., "Automating the Approximate Record-matching Process," *Information Sciences* (126:1-4), 2000, pp. 83-98.

Winkler, W. E., "Matching and Record Linkage," in *Record Linkage Techniques - 1997*, 1997, pp. 374-403.

Winkler, W. E., "Record Linkage Software and Methods for Merging Administrative Lists," in *Exchange of Technology and Know-How*: Eurostat: Luxembourg, 1999, pp. 313-323.

Witten, I. H. and Frank, E., *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementation*. San Francisco, California: Morgan Kaufmann Publishers, 2000.

Zhao, H. and Ram, S., "Clustering Database Objects for Semantic Integration of Heterogeneous Databases," in *Proceedings of AMCIS 2001*, 2001, pp. 357-362.