December 2002

A MODULAR APPROACH TO RELATIONAL DATA MINING

Claudia Perlich
New York University

Foster Provost
New York University

A MODULAR APPROACH TO RELATIONAL DATA MINING

Claudia Perlich and Foster Provost
Stern School of Business
New York University
cperlich@stern.nyu.edu fprovost@stern.nyu.edu

Abstract

Advances in information technology have provided the infrastructure to collect huge amounts of business data from many different sources: customers, suppliers, competitors, patents, products, and technical support. As a result of the amount of collected data there is an increasing need for better mechanisms for automated data analysis and decision support. Data-mining researches and tool builders have addressed the challenge of automated data analysis, but so far have focused on algorithms that apply to feature-vector representations (single tables). Despite significant efforts in the development and improvement of data warehousing systems and on-line analytical processing (OLAP), we feel that the support of the analysis of data in relational format is insufficient. This work investigates approaches to automated analysis of relational data and develops a modular data-mining approach for noisy business applications.

Introduction and Motivation

Machine-learning research has developed a number of automated data-mining tools, many which have found successful application in business domains. Most of these modeling techniques require a feature-vector input representation: examples are given as a (single) table of fixed-length feature vectors (x_1, x_2, x_3, $x_{4,}$…, x_n). In contrast, a relational representation consists of multiple tables instead of just one. A simple feature-vector representation is fundamentally unsuitable for most business domains, which are characterized by transaction and interaction data. Transaction data conventionally are stored in a multi-table relational database, where each customer has a number of transactions. The example below shows two simplified tables that capture customer and transaction information and link the information via the key CustomerId.

Name	CustomerID	City	Birth	Group
Smith	1092323	Boston	1973	A
…	…	…	…	…

Transaction ID	CustomerID	Date	Price	Payment
9238192346192	1092323	3/12/02	87.78	Visa
…	…	…	…	…

Each transaction involves a product that might be related to other products. The probability of buying product X (for example a battery for IBM Thinkpad T22) might depend on its relationship to product Y (IBM Thinkpad T22) that the customer bought 4 weeks ago. The relational nature of customer data makes traditional analysis tools suboptimal for customer preference and behavior modeling. Similar arguments hold for business intelligence applications involving general business relationships, alliances, and technological dependencies. Figure 1 shows an example of the relational nature of the business environment based on the co-occurrence of firms mentioned in news stories. Every edge represents at least 250 story-based co-occurrences within several months in 1999 of news.

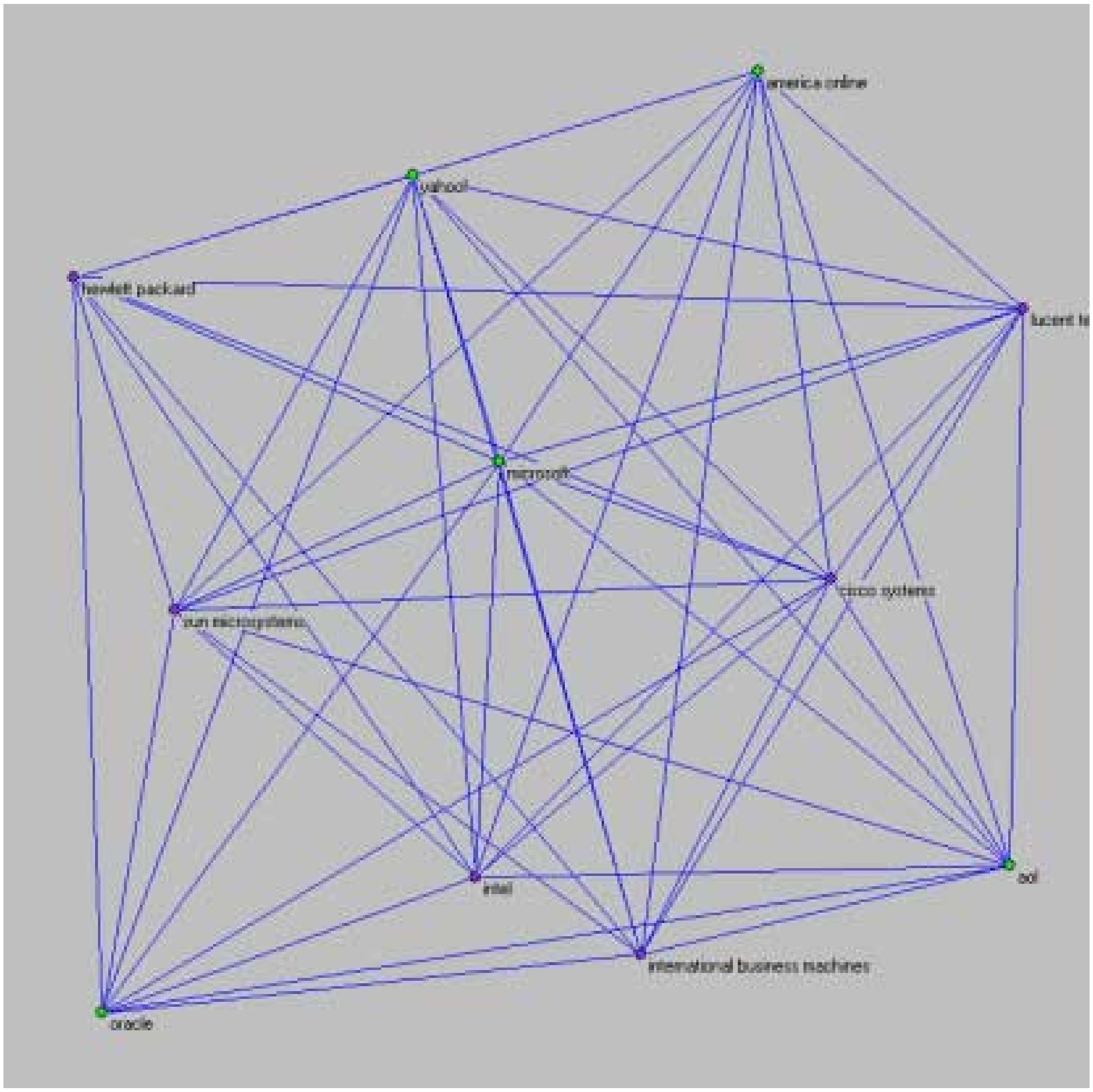

Figure 1. More than 250 Co-Occurrences of Business in News Stories Within One Year

There are a number of advantages associated with the ability to learn from relational data.

- **Multiple tables.** Handling multiple tables is natural in relational learning. Relational models can be integrated easily into data warehousing systems and knowledge management systems.

- **High expressiveness.** Relational models can represent more complex concepts than traditional models based on feature vectors. Structural concepts and relationships between objects are difficult to represent using a non-relational model.

- **Integration of background knowledge.** Domain knowledge can be encoded and given as explicit background knowledge. The use of background knowledge has been linked for instance by Aronis et al. (1996) to performance improvement. The source of the background knowledge can be a database, or an expert. Traditionally a human expert had to identify important background factors that can be used for modeling. But in many cases expert knowledge on a general problem class is limited and the number of possible features is almost infinite.

We propose to evaluate approaches to relational learning and develop new methods to improve the performance and the degree of automation. We are particularly interested in classification and the estimation of the probability of class membership. The next section presents a novel framework for relational classification and gives an overview of existing approaches.

A Modular Framework of Relational Data Mining

Relational data-mining approaches can be classified into two main categories. The objective of the first is to build a relational model in the form of first-order-logic rules directly from relational data. Such a rule could be the following: `Buy(Customer,P1)←Bought(Customer,P2) AND BatteryOf(P2,P1)`. This rule predicts that Customer will buy product `P1` if `P1` is the battery of a product that he bought before.

Alternatively relational data mining can be achieved by first transforming the relational data into a feature-vector representation followed by the application of a standard technique such as decision-tree induction or logistic regression to learn a predictive model. DeRaedt (1998) demonstrated that it is only possible in special cases to convert relational data into a propositional format without loss of information, and that the resulting feature space is exponential in the number of parameters. However, the objective of feature construction is not to perform a complete transformation, but rather to construct a small but relevant set of features from the relational background—features that capture the relational information for the learning task.

We will refer to the former approach as *direct relational learning* and to the later as the *feature-construction approach*. We argue that the feature-construction approach has several advantages:

- Standard ("propositional") learning techniques are numerous and substantially more mature than relational learning techniques. The feature-construction approach allows this maturity to "solve" a large part of the relational learning problem.
- The freedom of choice of a propositional learner allows different behavior depending on the learning objective (classification, probability estimation, interpretation, etc.), using the same data.
- The modularity of the approach simplifies the choice of system parameters by a domain expert (see below), and the addition of expert-suggested transformations.
- The run-time behavior can be adjusted by adding or subtracting constraints on aggregation and number of features.

A Modular Approach to Feature Construction from Relational Data

The relational nature of the input format manifests itself in two forms: set-valued attributes and structural connections (via keys) between relations. For instance, in order to classify customers in the given example based on their previous transactions, every customer has to be linked to his transaction via a join over both tables. Feature construction from relations can be framed as a three-step procedure. The first exploration step is the joining of the target relation with background relations. The second step is the aggregation of multiple values per target entry using aggregation operators based on the type of the value (e.g., mean for numeric values like price). The last step is the assessment of the quality of the feature with respect to the classification task, for example (simply) via the application of a threshold to a simple measure like correlation for numeric values or purity for categorical values. This framework is very general and can be instantiated with different exploration strategies, aggregation methods, feature-selection heuristics, and stopping criteria.

Related Work

This section presents a short overview over existing work on relational learning and other related topics.

OLAP: Data warehousing has addressed the issue of relational data analysis with on-line analytical processing. OLAP enables an analyst to create high-dimensional representations from joins of multiple relations and to select subsets and aggregations. The current capabilities of OLAP systems provide an infrastructure for data analysis but lack mechanisms for automated analysis and predictive modeling. OLAP can serve as the lower level on top of which a more automatic relational analysis tool could operate.

Traditional Data Mining: Machine learning and data mining have developed a suite of algorithms for automated learning of predictive models from feature vectors (x_1, x_2, x_3, x_4...x_n). The reader is referred to a standard textbook (Mitchell 1997) for an overview of existing methods. We integrate decision trees and logistic regression into the presented modular system (but others could be chosen).

SNA: Social Network Analysis (SNA) (Scott 1991) was developed in the social sciences for the analysis of the interaction of individuals, with particular focus on reputation and centrality measures. SNA was successfully employed by Sparrow (1991) for profiling to investigate money laundering. Limitations of SNA are its focus on data description rather than prediction and the restriction to a single object type. We integrate some operators from SNA in our modular framework.

Learning from relational databases: We are aware of only two SQL extensions that enable users learn rules from data. DBMiner by Han et al. (1996) integrates a set of discovery modules into an SQL-accessible database including a graphical user interface and a new extension of SQL called DMQL to request rules. MSQL proposed by Imielinski and Virmani (1999) is a similar modeling extension to SQL. Both methods require a very detailed specification of the model form. In the case of DMQL, the form of the rule has to be stated explicitly including all tables. The user has to know all potential factors in order to formulate such a rule. MSQL similarly requires all possible rules to be stored in advance in a specified table. Both methods lack a sufficient degree of automation to enable efficient data analysis with minimal user interaction.

Inductive Logic Programming: Learning from relational databases has been linked by Morik and Brockhausen (1996) to inductive logic programming (ILP) methods. ILP algorithms (for an overview see Muggleton and DeRaedt 1994) are capable of learning relational classification models from multi-table data. The result of ILP methods is a set of existentially unified first-order Horn clauses that can be applied as a classifier. The following example shows a predictive rule learned by an ILP approach that predicts that a customer X will buy the product Y, if Y is a battery of a product Z that he bought after September 2001: Buy(X,Y)←Battery_Of(Y,Z) AND Bought(X,Z,D) AND (D>200109)

A disadvantage of ILP traditionally has been its sensitivity to noise. However recent implementations have attempted to address this issue (we plan to assess their success). Another problem resulting from the logic background of ILP is its inability to estimate probabilities. This is a significant drawback for applications in business domains since we need probability estimates for decisions involving expected cost/benefit tradeoffs.

Propositionalization: Propositionalization is the construction of a feature-vector representation from a relational one (enabling the application of a prepositional learner). There exist a number of systems (e.g., LINUS, STILL, REPART discussed by Kramer et al. 2001) that perform binary propositionalization on relational data automatically. Those efforts are closely related to ILP and the constructed features have the form of logical clauses. However they are not capable of performing numeric aggregation. We are aware of only two previous attempts to use aggregation for propositionalization. Knobbe et al. (2001) applied SQL operators successfully to a banking domain. Morik and Brockhausen (1996) implemented a prototype for propositionalization called TOLKIEN as part of the MiningMart system, which also uses SQL queries to construct a feature vector from relational data. Both approaches are limited to the expressiveness of SQL, which provides only a small subset of possible aggregation operators. Neither system provides a method for automated feature selection.

Performance Comparison: The literature on relational learning lacks general comparison of different methods on realistic noisy domains. Most results are reported on small benchmark datasets with strong theoretical foundations (e.g. chemical and biological domains). Kramer et al. (1998) presented evidence that propositionalization approaches can outperform ILP implementations.

Comparative Study

We propose to compare a set of four existing ILP implementation (FOIL by Quinlan 1990, PROGOL by Muggleton 1995, LIME by McGreath and Sharma 1998, and TILDE by Bockeel and DeRaedt 1998) to our feature-construction prototype that uses different aggregation operators and feature selection in combination with logistic regression or decision trees. We collected six relational datasets from noisy business problems including customer service in banking, international patent references, industry groups, initial public offerings, and email communication in addition to 10 well-studied benchmark data sets from the ILP literature. The data can be characterized in terms of size, degree of uncertainty, connectivity, number of different object types, and number of features per object.

One objective of the prototype implementation is the development of heuristics for automated selection of aggregation operators based on simple data description. Other system parameters are the number of constructed features and the choice of propositional learner.

Our main study compares the classification and probability estimation performance of our prototype against the performance of ILP implementations for different parameter settings. The methodology follows closely that of Perlich et al. (2001) estimating learning curves of generalization performance as a function of training size using 10-fold cross validation. We evaluate classification performance on accuracy and probability estimation (i.e. ranking capability) on the area under the ROC ("Receiver Operating Characteristic") curve as proposed by Bamber (1975).

Implications

Depending on the results the implications may be many. Besides providing deeper understanding about relational learning, we hope to provide practical advice to practitioners who would like to use such systems, as well as providing heuristics that automatically find good settings for learning parameters as well as declarative language bias, given a task characterization. The second study may show that current technology is not yet satisfactory for automated knowledge discovery in noisy business domains. The complexity of realistic relational tasks may still require a domain expert to guide the discovery process. In this case we will outline requirements that need to be addressed by machine learning research to enable the application of those tools successfully to business domains. Once this is achieved these tools could fill the gap between push-button data mining systems and manual data exploration in existing OLAP systems and provide valuable decision support.

Summary

Having identified shortcomings of existing methods for mining relational data, we suggest a modular, feature-construction approach. We will instantiate the approach with feature-construction operators, drawing on sources including ILP methods, existing "propositionalization" operators, social network analysis (SNA), and graph analysis. This integration will enable the learning of complex relationships. For example, the modular approach can integrate ILP-produced clauses, standard propositional operators, social network analysis centrality or reputation measures, graph compression techniques, and expert-provided, domain-specific operators to generate potentially valuable features; these would be filtered through a feature selection method, and finally given as input to a standard learner (chosen based on task characteristics). We feel that given the diversity and complexity of relational problems, a modular approach can be tailored by a domain expert to a specific application and is more likely to succeed than an opaque, monolithic learning system.

The expected contributions of this work are twofold. First, the proposed modular approach can be expected to provide semi-automated solutions to a set of complex relational learning problems in noisy domains. We believe that this would be a valuable extension of existing OLAP systems. Second, this work will augment the machine learning literature and methodology on relational modeling. We will clarify the limits of current approaches and provide a classification schema for relational problems.

References

Aronis, J., Provost, F., and Buchanan, B. "Exploiting Background Knowledge in Automated Discovery," Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996, pp. 355-358.

Bamber, D. "The area above the ordinal dominance graph and the area below the receiver operating characteristic graph," Journal of Mathematical Psychology (12), 1975, pp. 387-415.

Bockeel, H., and De Raedt, L. "Top-down induction of first order logical decision trees," Artificial Intelligence (101(1-2)), 1998, pp. 285-297.

De Raedt, L. "Attribute Value Learning versus inductive logic programming: The missing links (extended abstract)," Proceedings of the Eights International Conference on Inductive Logic Programming, Springer Verlag, Berlin, 1998, pp. 1-8.

Han, J., Fu, Y., Wang, W., Chiang, J., Gong, W., Koperski, K., Li, D., Lu, Y., Rajan, A., Stefanovic, N., Xia, B., and Zaiane, O. R. "DBMiner: A System for Mining Knowledge in Large Relational Databases," Proceedings of the International Conference on Data Mining and Knowledge Discovery, 1996, pp. 250-255.

Imielinski, T., and Virmani, A. "MSQL: A query language for database mining," Data Mining and Knowledge Discovery 3(4), 1999, pp. 373-408.

Knobbe, A. J., De Haas, M., and Siebes, A. "Propositionalisation and Aggregates," Lecture Notes in Artificial Intelligence (2168), Springer Verlag, Berlin, 2001, pp. 277-288.

Kramer, S., Lavrac, N., and Flach, P. "Propositionalization Approaches to Relational Data Mining," Relational Data Mining, Springer Verlag, 2001, pp. 262-291.

Kramer S., Pfahringer, B., and Helma, C. "Stochastic Propositionalization of Non-determinate Background Knowledge," International Workshop on Inductive Logic Programming, 1998, pp. 80-94.

McCreath, E., and Sharma, A., “Lime: A System for Learning Relations” The 9th International Conference on Algorithmic Learning Theory(ALT98), Lecture Notes in Artificial Intelligence (1501), Springer-Verlag , 1998, pp. 336-374.

Mitchell T. M. Machine Learning, McGraw-Hill, 1997.

Morik, K. and Brockhausen P. “A Multistrategy Approach to Relational Knowledge Discovery in Databases,” Proceedings of the 3rd International Workshop on Multistrategy Learning, AAAI Press, 1996, pp. 17-28.

Muggleton, S. “Inverse entailment and Progol,” New Generation Computing (13), 1995, pp. 245-286.

Muggleton, S., and De Raedt, L. “Inductive Logic Programming; Theory and methods,” Journal of Logic Programming (19,20), 1994, pp. 629-679

Perlich, C., Provost, F., and Simonoff, J. “Tree Induction vs. Logistic Regression: A Learning-curve Analysis,” To appear in the Journal of Machine Learning Research, CeDER Working Paper #IS-01-02, Stern School of Business, New York University, NY, NY 10012, 2001.

Quinlan, R. “Learning Logical Definitions from Relations,” Machine Learning (5), 1990, pp. 239-266.

Scott, J. Social Network Analysis: A Handbook, Newbury Park, CA: Sage Publications, 1991.

Sparrow, M. K. “The application of network analysis to criminal intelligence: An assessment of the prospects,” Social Networks (13), 1991, pp. 251-274.