

A General Approach to Predictive Modeling in Multi-Relational Domains

(Dissertation Overview)

1 Motivation and Introduction

Our ability to collect and store large amounts of data on business transactions, accounting data, and customer information has increased dramatically over the past 20 years. But unfortunately our ability to analyze those data has not increased proportionally despite the growing computational power of modern computers. One of the main reasons for this discrepancy is the mismatch between the data representations for storage and analysis. Owing to its expressive power and the ability to capture the complexity of business domains, the relational data model (introduced by Codd in 1970 [5]) has become the standard of data representation for business databases. The relational data model differs from the traditional feature-vector representation used in statistics, machine learning and data mining since it allows for multiple tables and thereby set-valued attributes and object identifiers that link objects in relationships. This expressiveness is vital to capture the complexity of economic players and their relationships. Those relationships, as in many social networks, carry a significant amount of valuable information. However, the data analysis tools that are currently available are not sufficient to enable domain experts to convert this information efficiently into valuable knowledge for decision support.

Figure 1 presents a simple example of a relational database with 3 tables describing initial public stock offerings (IPO). The arrows denote the relations (potentially one-to-n) between objects via the identifier “Ticker”. Consider the task of learning a model that classifies IPO’s into “TargetClass”. Clearly, the two background tables SIC and Underwriter may contain important and possibly predictive information. In order to take advantage of this information, a domain expert would have to construct manually a set of features that he suspects to be predictive, generate a feature-vector, and estimate a statistical or data-mining model. This process of manual feature construction becomes infeasible for databases with a large number of tables due to the vast amount of possible attributes and furthermore will not lead to any new previously unknown insights.

Given the availability of relational data, its expressiveness, and high information content, there is a strong need to develop automated methods that enable domain experts to take full advantage

of relational data. My work presents a new modeling approach that is capable of learning predictive models from a relational database directly without the manual effort of a domain expert constructing features.

The advantages of learning from relational databases can be summarized as follows:

- **Standard of Relational Format:** Most data are stored in relational databases. Relational modeling can be easily integrated into existing data warehousing systems and knowledge management systems.
- **High Expressive Power:** Relational data can represent more complex concepts than traditional attribute-value languages. Relationships between objects and set-valued attributes cannot be expressed in a feature-vector representation.
- **Integration of Background Knowledge:** It is very natural in relational databases to introduce additional background knowledge. The target class of an initial public offering might be predictable from knowledge about the venture capitalists that funded the firm. This information can simply be introduced by adding another table.

My approach applies an automated sequence of steps that transform the relational problem into a conventional feature-vector representation. The challenge of the task is to find a small set of predictive features rather than constructing all or some arbitrarily chosen features. In particular I present a novel feature construction method that incorporates the target class into the feature construction. Figure 2 gives an overview of the architecture of my prototype for Automated Construction of Relational Attributes (ACORA). My dissertation will consist of three main parts:

Framework: I present a comprehensive framework for relational modeling that relates existing relational modeling approaches, provides directions for extensions and for the integration of traditional tools of data analysis from statistics and machine learning. I have adopted a transformation-based view that has a long tradition in both machine learning and statistical analysis (as for instance the log transformation of a Cobb-Douglas production function to allow for linear analysis). With respect to relational tasks, I demonstrate the advantages of explicitly transforming the original data schema into a feature-vector representation. The framework addresses the important role of similarity and distance measures for feature construction and suggests a novel methodology for

target-dependent feature construction. I also derive a theoretical Bayesian perspective that justifies the use of density estimation and distances for feature construction.

Prototype: The second part of this work is the implementation and evaluation of a prototype for ACORA, based on the guidelines derived from the theoretical framework. Our main objective is to demonstrate the ability to automate the process of constructing predictive features for a variety of relational domains. In the light of this goal, we chose a modular and extendable system design that also incorporates feature selection and model estimation. The performance of ACORA is evaluated on a suite of real domains and compared to alternative relational learning approaches.

Analysis of Interaction Effects: An additional objective of this work is to investigate the interactions between domain properties, feature construction, estimation procedures, and generalization performance. Investigating interactions is important with respect to two issues: reliability across tasks and the automation of lower-level modeling activity. It is important for successful applications of machine learning in business domains to have good and reliable solutions to most problems rather than a perfect solution to some and no solution on others. Furthermore, the acceptance of a modeling technique by domain experts requires a minimum level of automation. Given the increased complexity of relational models, a good modeling approach should be able to suggest reasonable parameter settings based on interactions between domain properties and the modeling procedure. To address this issue I derive hypotheses about model performance from a bias-variance framework.

2 Related Work

A number of disciplines have developed algorithms, approaches, and theories for relational modeling tasks, but they have mainly focused on special types of relational domains or specific modeling objectives. The main challenge of a general relational learning approach is the high complexity of both data and potential models. This has motivated the development of a number of special-purpose algorithms, each of which limits the problem complexity according to the properties of the particular domain. One of the results of this segmentation is the lack of performance comparison in the existing literature of relational learning. The small amount of published results is reported on small benchmark data sets with strong theoretical foundations and very low noise (e.g., chemical

and biological domains). Based on those studies, very little can be said about expected performance on noisy business domains.

2.1 The Relational Data Model

A relational domain is defined by a schema, which specifies a set of tables. We do not require any further degree of normalization beyond the defined so-called “first normal form” that only requires all attributes to be atomic (either numeric or categorical). Every table X is associated with a set of typed attributes $At(X)$. A particular attribute $A \in At(X)$ will be denoted $X.A$. The type $T(X.A)$ associates an attribute with a set of possible values $V(T(X.A))$. In particular, A can be numeric in which case $V(T(X.A)) = [-\infty, +\infty]$ or categorical.

All categorical attributes where the size of $V(T(X.A))$ is larger than a some lower bound are potential object references, in particular if they appear in more than one table (e.g., “Ticker”). Such attributes will be referred to as “reference attributes”. Note that our definition of a reference attribute is much less strict than the requirements for foreign keys in relational databases. A *foreign* key $X.K$ must be a candidate key (unique across the table) in a table X and for k to be a foreign key it has to appear in a second table Y such that for every occurrence of a value k in Y there must be exactly one element in table X with that value. In other words, there must be a 1-to-n relationship between table X and Y . We diverge from the strict notion of foreign keys mainly because we do not require more than first normal form for relational schema and because many relationships between objects are n-to-m.

2.2 Two Naive Approaches

Before discussing existing work we will first examine two naive approaches on the example of the IPO task that highlight the particular problems of transforming a relational learning tasks into a feature vector:

Join: Executing the universal join over all references attribute produces a single table in a feature-vector format. However, the result of an universal join, as shown for the example in Table 1, has a number of undesirable properties with respect to model estimation. A first observation is the multiplication of rows. The original table had 4 and the result has 10. For realistic domains with many 1-to-n relationships the size increases exponentially in the number of tables. Additionally,

the prior distribution of the target class has changed. In the original table it was 0.66 whereas now it is 0.5. Another observation is that we now make two predictions for the test case TAR without constraining them to be identical. There is no good theoretical argument as to how multiple predictions should be combined. All this reflects only the symptoms of a bigger problem: the evidence for every observation has been scattered across multiple rows. Learning procedures commonly assume that the observations in the rows are independently and identically distributed, which is clearly violated in this table. For instance a target concept were positive IPO's had MS as underwriter and not SAL cannot be induced from this representation.

Appending Rows: An obvious answer to the problems in the previous approach is to append the information from different tables to rows in the original table and thereby maintaining only one row per case. A possible result is shown in Table 2. This table has as many columns as the sum of the maximum number of related objects (3 in the case of underwrites for KLV and 2 in the case of SIC codes for YFT). Note that after creating binary dummy variables for the categorical values the missing values (*) disappear. However, this does not apply to numeric values. If one attribute of the related objects is continuous, the length of the feature vector would be the maximum number of related objects. This will result in very sparse feature vectors that are not suitable for learning. Even in the categorical case, creating dummies for all appearing SIC codes and Banks (the IPO domain has 500 banks and 417 SIC codes) will result in huge feature vectors that are likely to induce variance errors during model estimation.

2.3 Alternative Approaches

Inductive Logic Programming has produced a number of approaches that are capable of learning relational classification models from multi-relational data. Muggleton and De Raedt [17] provide a very good introduction to ILP theory, methods, and implementations. ILP methods learn a set of existentially unified first-order Horn clauses that can be applied as a classifier. An instance is classified as positive if any of the clauses is true. ILP systems have a number of disadvantages that make them unsuitable for business domains: sensitivity to noise and contradicting information, limitation to classification tasks rather than probability estimation, limited support of numeric features, and commonly unacceptable run-time behavior. Our experimental results contrast the performance of the ILP implementation FOIL [24] with ACORA.

Database-driven approaches include two SQL extensions that enable users to learn rules from relational databases. DBMiner [9] integrates a discovery module DMQL into an SQL-accessible database. MSQL [10] is a similar modeling extension to SQL. Both methods require a very detailed a priori specification of the expected model form. Both methods lack a sufficient degree of automation to enable efficient data analysis with an acceptably small amount of user effort. Neither of the two was available for a performance comparison.

Distance-based approaches to relational learning predefine a distance metric for relational data. The procedure presented by [11] distinguishes numeric attributes, categorical attributes, and object links. The distance between two objects and is the defined as the weighted sum across all attributes. The distance between categorical attributes is 1 if their values are different and 0 if equal. The distance of numeric attributes is the difference normalized by the range of the attribute. If an attribute is a foreign key to a set of other objects, the algorithm estimates the distance between all pairs of objects and takes the minimum distance across all pairs. The algorithm proceeds recursively until a maximum depth is reached. Similar to the naive approach of joining all tables only the information of one object (with minimum distance) will affect the prediction and more complex concepts cannot be expressed.

Transformation-based approaches have become more dominant over the recent past and cover a variety of methods that similarly to ACORA transform the relational domain into a feature vector. Originally ILP methods were applied to construction binary features. Newer approaches introduce numerical aggregation methods for feature construction. Knobbe [12] applied SQL operators successfully to a banking domain. Morik and Brockhausen [16] implemented a similar prototype called TOLKIEN as part of the MiningMart system. Probabilistic relational models (PRM [8]) also use standard SQL functionality for aggregation in combination with a constrained Bayesian Network [18] model. The main extention of my work is the development of novel aggregation methods to construct task-specific predictive features. In particular categorical features with high dimensionality (many possible values as in the example of banks) cannot be aggregated using standard SQL operators. Simple aggregation methods for categorical variables include the construction of dummies for all possible values if the dimensionality was smaller than some arbitrary cutoff (commonly around 50) or to choose the most common value.

In summary, existing relational methods fall short on the following issues:

1. Automation: Few of the existing approaches are fully automated, either with respect to the aggregation methods or with respect to the search strategy. ILP and binary feature construction require significant technical knowledge to specify search heuristics for particular domains. One ILP exception is FOIL that can be run with little specification beyond the data schema. However the default parameters do not estimate predictive models but rather descriptive as noted by De Raedt [6]. The transformation-based methods are generally more automated than other approaches.

2. Reliability on a variety of domains: Current implementations of relational learners are often criticized for low performance on noisy domains or domains with dominantly highdimensional categorical attributes. The field has not agreed on guidelines about the conditions under which each approach is likely to perform well.

3. Usability by domain experts: Most existing learners require the low-level specification of search constraints that assumes substantial technical understanding of the implementation.

4. Run time: Most implementations of ILP as well as binary feature construction suffer from unacceptable run times on data sets as small as 1000 observations.

5. Expressive power: All discussed methods impose significant constraints on the potential models (either directly on the model form or through a limited set of aggregates for feature construction).

3 ACORA, a Prototype

Figure 3 gives an overview of the four main modules of ACORA’s system architecture: exploration using joins, feature construction using aggregation, feature selection, and model estimation.

3.1 Exploration

ACORA requires as input a typed relational schema as defined in section 2.1. It will deduce potential reference attributes that can be used for joins based on the attribute names and types, and it constructs a graph representing the tables as nodes and reference attributes as edges. The exploration uses a breadth-first search starting from the target table, joining over reference attributes corresponding to the edges. The result of every path is a set of objects that are related to the target objects. Table 3 shows the result of the path from the IPO table to the Underwriter on “Ticker”

grouped by the rows in the IPO table.

3.2 Automated Feature Construction through Density Estimation

The background tables (Underwriter and SIC) can only improve classification performance if the bank underwriting IPO's of the positive class are different from banks of IPO's in negative class. Assuming a theoretical framework where the entities in the background tables are drawn independently from two different distributions D_1 and D_0 , we developed the following methodology that is described more extensively by Perlich and Provost [20]. The **first aggregation step** estimates the class-conditional distributions of banks from the training set in the IPO table. Supposing an order on the values of the categorical attributes (mapping from bank b to a vector position i) the value for D_n at position i is defined as the class-conditional prior:

$D_n[i] = \frac{O_n(b)}{\sum_{b \in B} O_n(b)}$ where $O_n(b)$ is the number of occurrences of bank b corresponding to vector position i underwriting an IPO of class n . The resulting estimates of the class-conditional distributions for our 3 IPO's in the example in table are given by: $D_1=[0.25,0.5,0.25,0,0]$ and $D_0 = [0,0.5,0,0,0.5]$ using the order [DJL,MS,GS,LEH,SAL]. The **second aggregation step** estimates the unconditional densities over banks for every IPO f : $DB_f[i] = \frac{O_f(b)}{\sum_{b \in B} O_f(b)}$ where $O_f(b)$ is the number of occurrences of bank b corresponding to vector position i as underwriter of IPO f . The estimates for the IPO-specific distributions in our example are: $DB_{KLV} = [0.33,0.33,0.33,0,0]$, $DB_{YFT} = [0,0.5,0,0,0.5]$, $DB_{ERG} = [0,1,0,0,0]$, and for the test case $DB_{TAR} = [0,0.5,0,0.5,0]$.

The **third aggregation step** constructs features from the class-conditional densities and the document densities through the application of various vector distance measures including: cosine, Euclidean, and Mahalanobis [14], a variance-adjusted Euclidean distance. The new target table after adding the Euclidean distances (ED) between the class-conditional densities and the IPO densities (features $e_1(f)=ED(D_0, DB_f)$ and $e_0(f)=ED(D_1, DB_f)$) is shown Table 4. A simple but effective extension is the construction of the differences between the vector distances $e_1(f) - e_0(f)$ reflecting whether the case is closer to the density estimate of class 1 or class 0.

One can observe a number of interesting properties of the outlined feature construction method: **Dimensionality Reduction:** The use of vector distances compresses the high-dimensional space of possible categorical values into two (one for each class) dimensions per vector-distance metric. This quality allows the exploration of related entities to a much greater depth without incurring

major variance errors during model estimation.

Discriminative Information Preservation: The loss of discriminate information is minimal. Significant differences in the class-conditional distributions will be reflected in the vector distances. If indeed the two class distributions are identical, the difference in the distances should be close to zero for all cases and the feature would be discarded during feature selection.

Efficiency: The total complexity of the aggregation is $O(n * k * \log(k))$, where n is the size of the table after the join on ID and k is the number of possible categorical values. The conditional class distribution can already be estimated during the join execution. One additional pass over the resulting table is required to construct the case-specific distributions and distances. The $k * \log(k)$ factor reflects the use of hashes to store intermediate results. But even the estimation of the commonly used mode of a categorical distribution would exhibit the same overall complexity.

Domain Independence: The density estimation does not require any prior knowledge about the application domain and therefore is suitable for a variety of applications beyond text classification.

Monotonic Relationship: The use of differences of vector distances transforms the categorical attribute into a numerical feature that is monotonic in the probability of class membership. This makes logistic regression a natural choice for the model induction step.

Task-Specific Feature Construction: The advantages outlined above are possible because I use the target value during feature construction. This practice requires splitting the training set into two separate portions for 1) the class-conditional density estimation and feature construction and 2) the estimation of the classification model. Having fewer data points for model induction increases the risk of overfitting and motivates feature selection as well as a more biased model category such as logistic regression.

In addition to the presented vector distances between the class-conditional densities, we constructed a number of alternative features including the number of related objects, counts for each of the 5 most common categorical values (an extension of the mode), counts for the 5 most discriminative categorical values (where the differences between entries in D_0 and D_1 is maximal), and vector distances to the unconditional density D_{All} estimated over all training examples. The last feature helps us to evaluate whether the performance of the proposed method is mostly due to the dimensionality reduction or caused by the conditioning on the class label.

3.3 Feature Selection

Following the feature construction, ACORA selects randomly a small set of features from the extended target table, where the probability of selection is proportional to the AUR (Area under the Receiver Operating Curve, [2]) of a linear logistic classification model estimated on the particular feature. This process is repeated 10 times and the results from the 10 models are averaged into a final prediction. We are not aware that this combined approach of feature selection and model bagging has been used before in classification and compare it to model estimation without feature selection or bagging. Preliminary results confirm that for some domains feature selection plays a very important role. Since the target was used to construct the features, the performance cannot be evaluated on the same data. In particular, ACORA uses only a subset of the training cases from the target table for feature construction and the remaining part for feature selection and model estimation. Note that feature selection may be less important for learning methods that are robust to high dimensionality, such as Support Vector Machines [1].

3.4 Model Estimation

The final step is model estimation. ACORA offers a variety of learners including linear regression, logistic regression, decision trees [23], naive Bayes, and regression trees [26].

4 Experiments and Evaluation

We present here some results on the competitive performance of ACORA on a number of classification and probability estimation tasks from different domains.

4.1 Initial Public Offerings

The database consists of three tables

- IPO(Date,Size,Price,Ticker,Exchange,Runup)
- UNDER(Ticker,Bank)
- SIC(Ticker,SIC)

and covers 2750 initial public stock offers (IPO) during the years 1990 to 2000. Each offer has one or more underwriting banks (UNDER). The attributes of the IPO table are the date (Date) of

the offer, the number of shares (Size), the price of a share (Price), the ticker symbol (Ticker), the exchange on which the stock was offered (Exchange), and the percentage increase in price over the first trading day (Runup).

Consider a number of classification tasks for this database to evaluate ACORA’s performance. The first task predicts whether an offer will experience a price increase of more than 50%. The second task is to predict whether an offer was listed under the SIC code 7372 (adding a class attribute of one of the SIC codes in the SIC table was 7372 and removing the table SIC). The next two tasks were to identify whether an offer was traded on the NASDAQ or on the NYSE respectively. The tasks are not complementary since there are a total of 6 exchanges. The last classification task was to predict whether an offer would have more shares than 10000 (replacing the Size variable with a binary target).

Tables 5 and 6 show the out-of-sample performance on a set of 750 IPO’s in terms of classification accuracy and area under the receiver operating curve (AUR) for probability estimation for the different methodologies: no feature construction (No), binary feature construction using an extension of FOIL (FOIL), logic-based classification using FOIL (FOIL), and ACORA’s feature construction (A). For all methodologies that involved feature construction we estimated consecutively two classification models using logistic regression (LR) and the decision tree implementation C4.5 [23]. The baseline performance without feature construction uses all original attributes in the IPO table except for categorical variables with more than 200 possible values (Ticker and SIC). This base set of features serves also as the starting point for the feature construction methods that add new features.

The Runup and SIC=7372 tasks were very hard: no method is able to improve the accuracy over the prior. On the other tasks, ACORA (A LR and A C4.5) outperforms FOIL and binary feature construction, if only marginally in the case of NYSE classification.

For the ranking comparison we used the frequency counts at the leaves for the decision tree¹. FOIL is not able to provide ranking and is therefore not presented in the comparison. All the results show that the features constructed by ACORA provide predictive power. Figure 3 shows learning curves for classification accuracy, including error bars of \pm one standard deviation for the experiments exploring 12 joins and using logistic regression for model estimation. The learning

¹This is not an optimal strategy for ranking using trees, and improvements have been proposed in [22].

curves show that increasing the training-set size always improves the generalization performance. The graph also highlights the different performance levels across ACORA’s feature construction methods with increasing complexity using no relational features (NO), unconditional density distances (VD), conditional density distances (VDNP= e_1 and e_0) and differences in conditional density distances (VDD= $e_1 - e_0$).

4.2 Citation-Based Document Classification

We also evaluate the density-based feature construction on the CORA database [15]. It contains 4200 publications in the field of Machine Learning that are categorized into 7 classes: Rule Learning, Reinforcement Learning, Theory, Neural Networks, Probabilistic Methods, Genetic Algorithms, and Case-Based Reasoning. Rather than using the text we only rely on the author and citation information presented in three tables:

- DOCUMENT(id,class)
- AUTHOR(id,author)
- REFERENCE(id,id)

The domain has 4007 unique authors with an average of 2.1 authors per paper and a total of 90,000 citations between documents.

ACORA learns separate binary classification models for each of the 7 classes and predicts the final class with the highest probability score across the 7 model predictions. Figure 4 shows ACORA’s performance relative to two alternative relational learning methods: a Probabilistic Relational Model (PRM) with the results reported in [27], and a Simple Relational Classifier (SRC, Macskassy and Provost 2003) that uses the known class labels of related documents. The latter method is closely related to Chakrabarti’s [4] work, which draws from the theoretical framework of Markov Random Fields (MRF). The Simple Relational Classifier iteratively propagates the evidence of known class labels of related papers under the assumption that documents from a particular scientific field will dominantly cite previously published papers in the same field. We did not include any ILP method since none could outperform the naive model of predicting the majority class (accuracy of about 30%). A Naive Bayes classifier on the full text achieves a accuracy close to the PRM performance. ACORA clearly outperformns any other method by a large margin, achieving a very high accuracy of more than 80% on only 400 training examples. In summary, these

results are very promising with respect our ability to take advantage of the relational structure and automate the process of transforming a complex domain into a feature-vector representation using density distances.

5 Conclusion

By developing a new target-based feature construction methodology (using density distances) for learning from relational data I have been able to design an automated methodology for predictive modeling from relational data representations that shows superior performance on a number of classification and probability estimation tasks across different domains.

This work furthermore contributes a comprehensive framework for modeling of relational domains and derives a number of guidelines for the development of the transformation-based approach. We introduced a novel methodology that automates the task-specific construction of predictive features without requiring prior domain knowledge beyond a schema definition. The introduction of class-specific density estimation increases significantly predictive power over alternative methods.

This work contributes to a body of existing research on relational modeling with regard to (1) new methods for automated feature construction from sets of objects based on reference vectors and different vector distances; (2) comparative performance evaluation of different relational approaches including binary feature construction, logic-based classification, and target-specific feature construction; (3) assessment of interactions between feature construction methods and model class (decision trees profit from providing differences of the vector distances; whereas linear models have the ability to construct differences internally); (4) recommendations for the applicability of different methods depending on domain properties (e.g., inherent uncertainty has a negative impact on the relative performance of logic-based approaches, whereas an increase relevance is more suitable for logic-based approaches); (5) relating existing distance metrics in a unifying framework; (6) generalization of relational modeling approaches from mainly classification to probability estimation and regression; (7) improvement of classification and probability estimation performance.

If further evaluation of ACORA confirms our favorable preliminary results, potential applications are many, ranging from managerial decision support, personalization, fraud detection and direct marketing to scientific analysis of complex domains.

Prediction Task

Background Knowledge

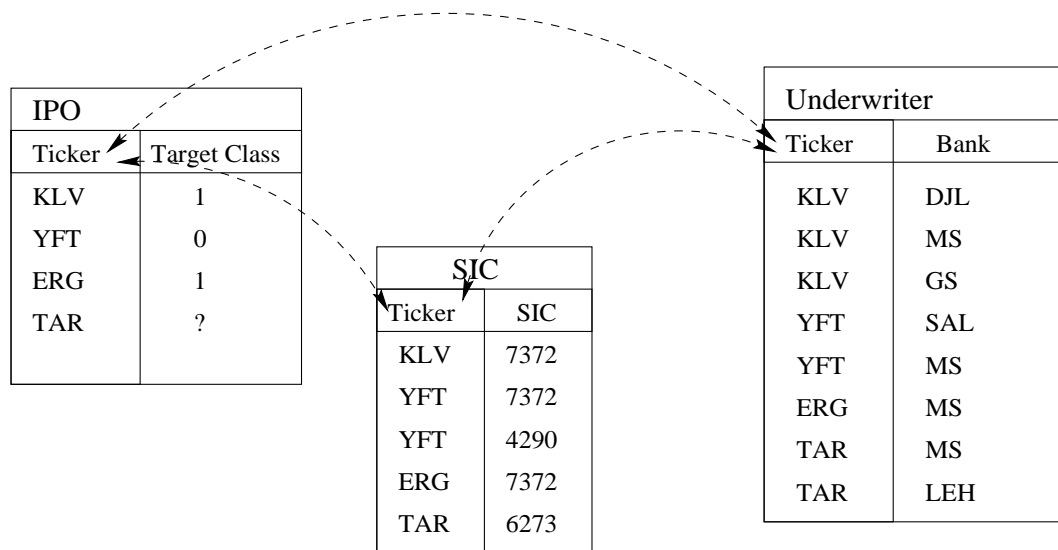


Figure 1: Example for IPO Schema

Ticker	SIC	Bank	TargetClass
KLV	7372	DJL	1
KLV	7372	MS	1
KLV	7372	GS	1
YFT	7372	SAL	0
YFT	7372	MS	0
YFT	4290	SAL	0
YFT	4290	MS	0
ERG	7372	MS	1
TAR	6273	MS	?
TAR	6273	LEH	?

Table 1: Full Join on Ticker

Ticker	SIC1	SIC2	Bank1	Bank2	Bank3	TargetClass
KLV	7372	*	DJL	MS	GS	1
YFT	4290	7372	SAL	MS	*	0
ERG	7372	*	MS	*	*	1
TAR	6273	*	MS	LEH	*	?

Table 2: Extended Table

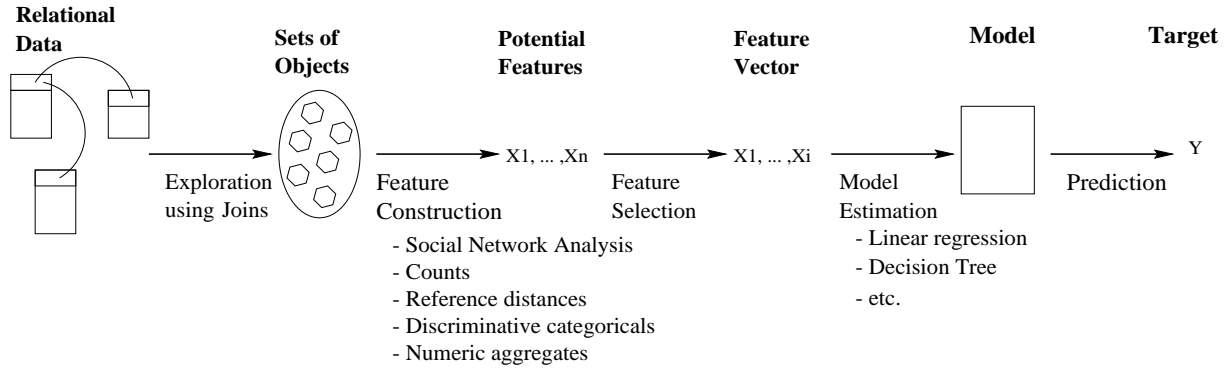


Figure 2: ACORA Architecture

References

- [1] B.E. Boser, I.Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learning Theory*, pages 144–152, 1992.
- [2] A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. In *Pattern Recognition*, volume 30(7), pages 1145–1159, 1997.
- [3] L. Breiman. Bagging predictors. In *Machine Learning*, volume 24(2), pages 123–140, 1996.
- [4] S. Chakrabarti, B. Dom and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of ACM SIGMOD*, 1998.
- [5] E.F. Codd. A relational model of data for large shared data banks. *Comm. ACM*, 13 (6):377–387, 1970.
- [6] L. De Raedt. Attribute value learning versus inductive logic programming: The missing links (extended abstract). In *ILP98*, volume 1446 of *LNAI*, pages 1–8. Springer-Verlag, 1998.
- [7] P.A. Flach. *The logic of learning: A brief introduction to inductive logic programming*, 1998.

Ticker	Bank
KLV	DJL
KLV	MS
KLV	GS
YFT	SAL
YFT	MS
ERG	MS
TAR	MS
TAR	LEH

Table 3: Result of Exploration-Join on “Ticker”

Ticker	e_1	e_0	$e_1 - e_0$	TargetClass
KLV	0.04	0.49	-0.45	1
YFT	0.37	0	0.37	0
ERG	0.37	0.5	-0.125	1
TAR	0.37	0.5	-0.125	?

Table 4: Target Table with New Features

- [8] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, pages 1300–1309, 1999.
- [9] J. Han, Y. Fu, W. Wang, J. Chiang, W. Gong, K. Koperski, D. Li, Y. Lu, A. Rajan, N. Stefanovic, B. Xia, and O.R. Zaiane. DBMiner: A system for mining knowledge in large relational databases. In *Proc. 1996 Int’l Conf. on Data Mining and Knowledge Discovery (KDD’96)*, pages 250–255, Portland, Oregon, 1996.
- [10] T. Imielinski and A. Virmani. Msql: A query language for database mining. In *Data Mining and Knowledge Discovery*, volume 3(4), pages 373–408, 1999.
- [11] M. Kirsten, S. Wrobel, and T. Horvath. Distance based approaches to relational learning and clustering. In Dzeroski Lavrac, editor, *RDM*, pages 213–232. Springer Verlag, 2001.

Task	Prior	No LR	No C4.5	A LR	A C4.5	FOIL	FOIL LR	FOIL C4.5
Runup	0.964	0.964	0.964	0.964	0.964	0.96	0.95	0.955
7372 Sector	0.86	0.86	0.855	0.857	0.86	0.837	0.84	0.83
NASD	0.53	0.67	0.664	0.76	0.766	0.679	0.64	0.69
NYSE	0.67	0.74	0.735	0.81	0.81	0.771	0.8	0.8
Size	0.72	0.786	0.786	0.835	0.826	0.744	0.76	0.77

Table 5: Classification Accuracy

Task	No LR	No C4.5	A LR	A C4	FOIL LR	FOIL C4.5
Runup	0.57	0.5	0.77	0.5	0.54	0.55
IT Sector	0.7	0.5	0.78	0.759	0.73	0.66
NASDAG	0.68	0.658	0.83	0.851	0.74	0.73
NYSE	0.76	0.793	0.888	0.893	0.81	0.8
Size	0.87	0.787	0.885	0.874	0.85	0.79

Table 6: Ranking Ability Evaluated by Area Under ROC

- [12] A. Knobbe, M. De Haas, and A. Siebes. Propositionalisation and aggregates. In *LNAI*, volume 2168, pages 277–288, 2001.
- [13] S.A. Macskassy, and F. Provost. A Simple Relational Classifier. In *Multi-Relational Data Mining at SIGKDD 2003*.
- [14] P.C. Mahalanobis. *On the generalized distance in Statistics*, volume 12. 1936.
- [15] A.K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of Internet portals with machine learning. *Information Retrieval*, 3(2): pages 127–163, 2000.
- [16] K. Morik and P. Brockhausen. A multistrategy approach to relational knowledge discovery in databases. In *Proceedings of the 3rd International Workshop on Multistrategy Learning*, pages 17–28. AAAI Press, 1996.
- [17] S.H. Muggleton and L. DeRaedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19 & 20:629–680, May 1994.
- [18] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.
- [19] C. Perlich and F. Provost. A modular approach to relational data mining. In *American Conference on Information Systems (AMCIS)*, 2002.
- [20] C. Perlich and F. Provost. Aggregation-Based Feature Invention and Relational Concept Classes. In *Proceedings of the Ninth ACM SIGKDD*, 167-176, 2003.
- [21] C. Perlich, F. Provost, and J. Simonoff. Tree induction vs. logistic regression: A learning-curve analysis. *To appear in the Journal of Machine Learning Research.*, 2002.

- [22] F. Provost and P. Domingos. Tree induction for probability-based rankings. *To appear in Machine Learning.*
- [23] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann Publishers, Los Altos, California, 1993.
- [24] J.R. Quinlan and R.M. Cameron-Jones. Foil: A midterm report. In P. Brazdil, editor, *Proceedings of the 6th European Conference on Machine Learning*, volume 667, pages 3–20. Springer-Verlag, 1993.
- [25] B. Pfahringer S. Kramer and C. Helma. Stochastic propositionalization of non-determinate background knowledge. In *International Workshop on Inductive Logic Programming*, pages 80–94, 1998.
- [26] L. Torgo and J. Pinto da Costa. Clustered partial linear regression. In *Machine Learning: ECML 2000, 11th European Conference on Machine Learning, Barcelona, Catalonia, Spain, May 31 - June 2, 2000, Proceedings*, volume 1810, pages 426–436. Springer, Berlin, 2000.
- [27] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 870–878, 2001.

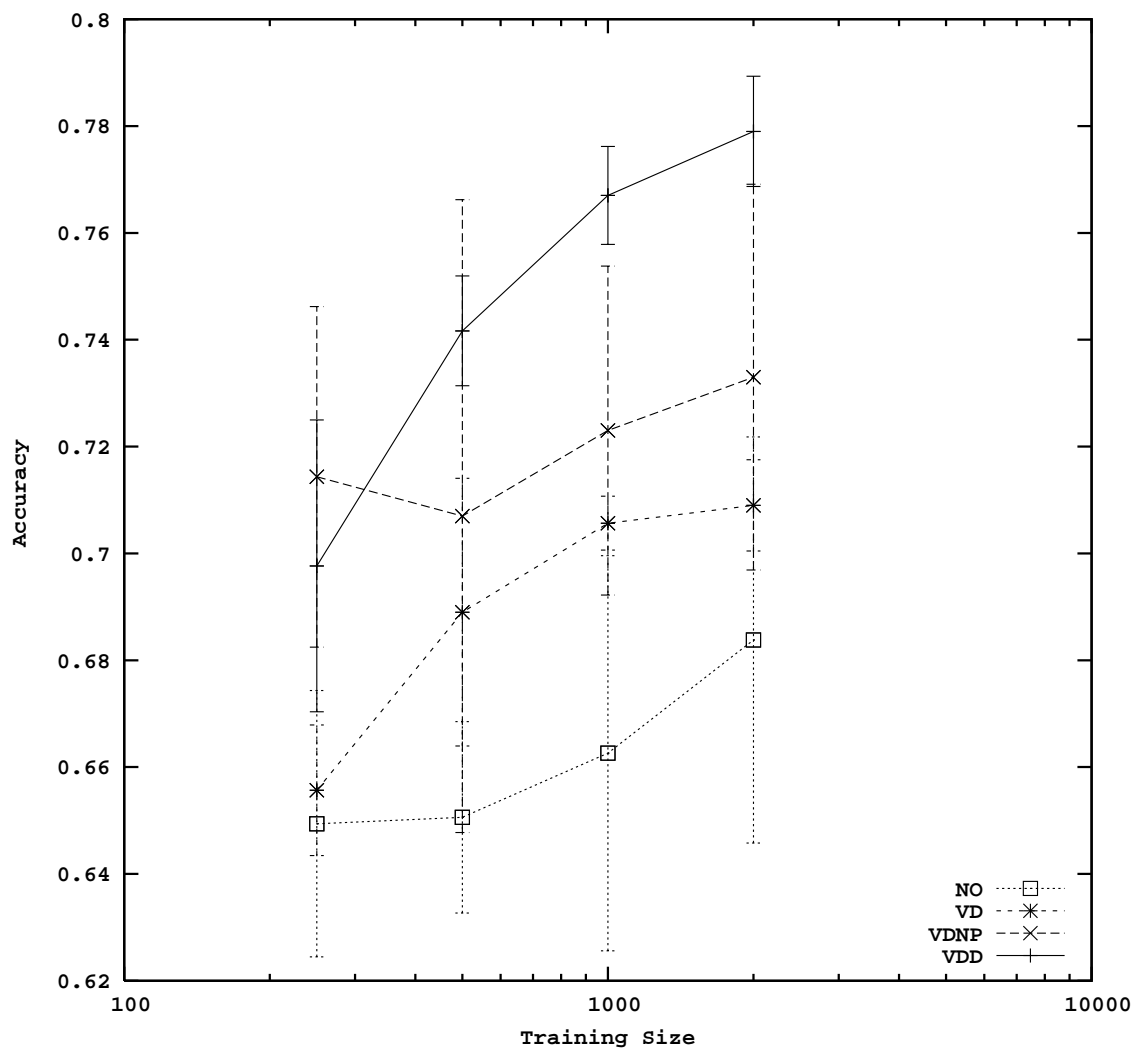


Figure 3: Classification Accuracy on the IPO Domain

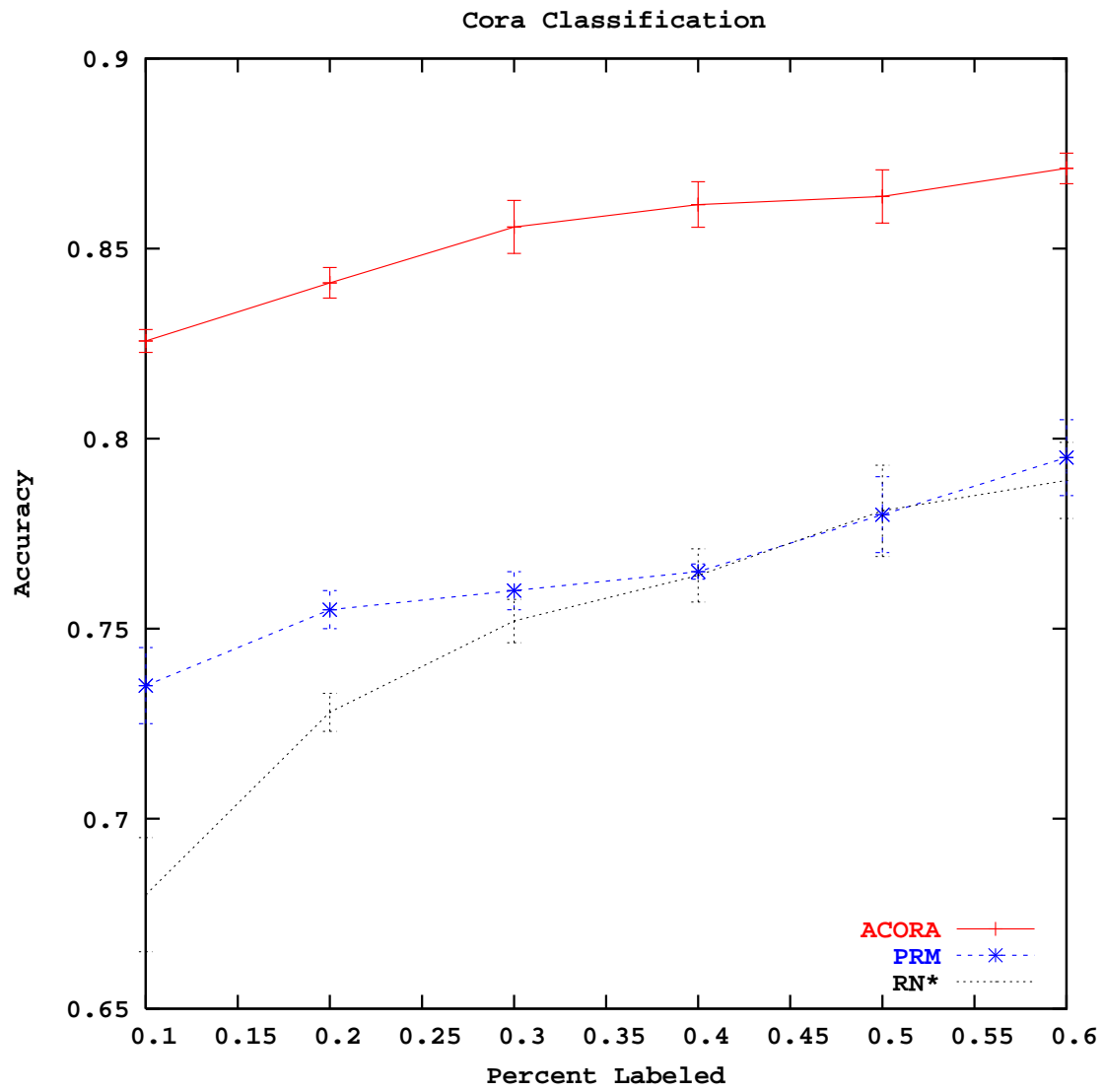


Figure 4: Classification Accuracy on the CORA Domain