Iterative Classification in Relational Data

Jennifer Neville and David Jensen

{jneville | jensen}@cs.umass.edu
Knowledge Discovery Laboratory
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610

Abstract

Relational data offer a unique opportunity for improving the classification accuracy of statistical models. If two objects are related, inferring something about one object can aid inferences about the other. We present an iterative classification procedure that exploits this characteristic of relational data. This approach uses simple Bayesian classifiers in an iterative fashion, dynamically updating the attributes of some objects as inferences are made about related objects. Inferences made with high confidence in initial iterations are fed back into the data and are used to inform subsequent inferences about related objects. We evaluate the performance of this approach on a binary classification task. Experiments indicate that iterative classification significantly increases accuracy when compared to a single-pass approach.

Introduction

The structure of relational data presents a unique opportunity to use knowledge about one object to inform inferences about related objects. The goal of this work is to explore how conventional techniques for constructing and using classification models can be used in new ways to exploit this opportunity. Specifically, we investigate using simple Bayesian classifiers in an iterative fashion to improve classification accuracy by exploiting relational information in the data.

The hypothesis underlying this approach is that if two objects are related, inferring something about one object can assist inferences about the other. We call this approach *iterative classification.* Inferences made with high confidence in initial iterations are fed back into the data to strengthen inferences about related objects in subsequent iterations. Experimental evidence shows that iterative classification leads to a significant increase in classification accuracy when compared with a single-pass approach. This suggests that there are distinctive characteristics of relational data that can be used to improve classification accuracy.

Simple Bayesian classifiers (SBCs) take traditional attribute-value data as input. In order to use SBCs with relational data, we flatten the data first by calculating intrinsic and relational attributes about individual objects. However, we maintain a relational representation of the data and flatten dynamically only when needed by the classifier. Retaining the relational representation makes it possible to extract data, perform a series of calculations and then feed the results back into the relational structure for use in future calculations. The ability to perform iterative calculations in this manner is one of the benefits of maintaining a relational data representation. For example, some measures of centrality in social network analysis (Wasserman and Faust 1994) can only be calculated in such an iterative fashion. Kleinberg's Hubs and Authorities algorithm for Web searching (1998) also uses iterative calculations in this manner.

Relational Classification

Relational data sets present a special opportunity for improving classification. The opportunity exists if, when two objects are related, inferring something about one object can help you infer something about the other. For example, if two people jointly own a business, and one of them is identified as a money launderer, then it may be more likely that the other is also involved in money laundering. The ability to exploit associations among objects in this manner has applications in many fields with relational data, including epidemiology, fraud detection, ecological analysis and sociology.

A relational classification technique, which uses information implicit in relationships, should classify more accurately than techniques that only examine objects in isolation. Relational classification techniques could be particularly useful in domains with abundant information about the relationships among objects but only limited information about the intrinsic properties of those objects. For example, relational classification might be applied to identify potential money-laundering operations based on bank deposits and business connections (Jensen 1997). In such a situation, the existence of an employee making large cash deposits for more than one business gives little information as to the legitimacy of those businesses. Many service and retail companies have high volumes of cash sales and it's not uncommon for a person to be employed by more than one company. However, if one of the businesses is discovered to be a front company for money laundering, then the related businesses are more

J. Neville and D. Jensen (2000). Iterative classification in relational data. *Proceedings of the AAAI 2000 Workshop Learning Statistical Models from Relational Data*. AAAI Press. pp. 42-49

likely to be front companies as well. In this case, the relationship provided by a common depositor is more useful in the context of knowledge about the related companies.

There are multiple ways to approach classification in a relational context. One approach ignores related objects and builds classifiers based only on the properties of an object in isolation. Another approach looks at the properties of both the object and its related objects in a static manner, by taking a snapshot of the relational context at some time prior to classification. A third approach uses properties of related objects and dynamically updates those properties as predictions about related objects change. Iterative classification uses the latter approach, applying SBCs in a dynamic way to fully leverage the structure of relational data.

For example, in a data set we describe below, a relational data structure represents companies, their subsidiaries, corporate stockholders, officers and board members. Companies are linked indirectly through stockholders and through people serving simultaneously on several boards (see figure 1). Such an interlocking structure allows the creation of both *intrinsic* and *relational* attributes. Intrinsic attributes record characteristics of objects in isolation — for example, company type or officer salary. Relational attributes summarize characteristics of one or more related objects — for example, a company's number of subsidiaries or the maximum salary of any board member.

Relational attributes fall into two categories which we will call *static relational* and *dynamic relational*. Any intrinsic attribute has the potential to be predicted by an SBC model; from the same company data we could predict any of the intrinsic attributes mentioned above. Static relational attributes use *known* intrinsic attributes of related objects and as such they can be computed without the need for inference. The values of static relational attributes remain constant over the course of classification. Dynamic relational attributes use *inferred* intrinsic attributes of related objects so they require that at least some related objects be classified before the attribute can be computed. The values of dynamic relational attributes may change as classification progresses and additional inferences are made about related objects.

For example, if we were predicting company type, then static relational attributes might record the number of board members who have the title CEO or the average salary of officers. Dynamic relational attributes might record the most prevalent type of corporate stockholder or the maximum number of subsidiaries that share the same type. Both of these latter attributes are dynamic and relational because they reference the company type of related objects, the very thing we are trying to infer about the primary object. For notational simplicity, the remainder of this paper will refer to intrinsic and static relational attributes as *static attributes*, and dynamic relational attributes as *dynamic attributes*.

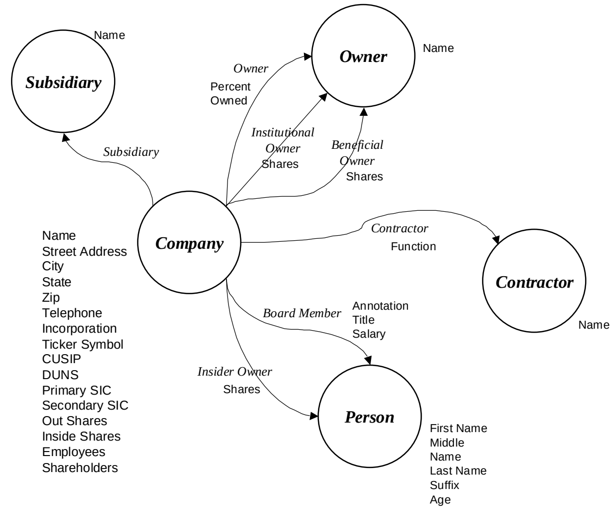

Figure 1: Corporate data ontology

In a relational corporate data set, knowing the type of one company might help us infer the type of another company to which it is related, and vice versa. For instance, we may find that individuals tend to serve on boards of companies with the same type, so if a person is on the board of both company X and company Y, and company X is a bank, then company Y is more likely to also be a bank. Or we may find that companies tend to own stock in companies with the same type, so if a company owns both company X and company Y, and company X is a bank, then company Y is more likely to be a bank. In situations of this type, the relations among objects assist the inferences.

In iterative classification, a model is built using a variety of static and dynamic attributes. When training the model, the class labels of all objects are known and consequently the values of all dynamic attributes are also known.

The trained classifier is then applied to previously unseen examples for which the class labels are unknown. Initially, because class labels of related objects are unknown, values of all dynamic attributes are also unknown. However, their values can be estimated as classification progresses. At the onset, the classifier makes predictions for all objects based only on the values of static attributes. Classifications made with high probability are accepted as valid and are written into the data as *known* class labels. SBCs are useful for iterative classification because each prediction has an associated probability estimate that can be used to guide iterative classification.

After some percentage of the most certain classifications are "accepted" the classifier starts the next iteration, recalculating dynamic attributes in light of this new information and proceeding with classification once again. At each iteration, additional dynamic attributes are filled in and a greater percentage of classifications are accepted.

Because each prediction is both recalculated and reevaluated for each iteration, a prediction about a given object may change over the course of iterations. If the probability associated with a particular prediction falls out of the top percentage of accepted predictions, the inference will be removed from the data. Also, if the predicted class label changes for a particular object (and the prediction is accepted), the new class label will be written into the data for that object.

After a given number of cycles, when all classifications have been accepted, the process terminates. We conjecture that iterative classification will produce more accurate predictions of class values than conventional classification involving intrinsic and static relational attributes alone.

Iterative Classification Algorithm

1. Build model on fully labeled training set
2. Apply trained model to test set of N instances. For each iteration i : 1 to m
   a. Calculate values for dynamic relational attributes
   b. Use model to predict class labels
   c. Sort inferences by probability
   d. Accept k class labels, where $k = N (i / m)$
3. Output final inferences made by model on test set

Necessary conditions

We conjecture that a relational data set must exhibit several characteristics before an iterative classification approach will improve accuracy over a single-pass approach. An initial outline of these characteristics is given below; however, further investigation is needed to determine the exact nature and scope of these conditions.

First, using static attributes alone should not maximize accuracy. If a classifier can make highly accurate predictions without dynamic attributes, there is little room for improvement via iteration. Also, if an inference about one object *does not* inform subsequent inferences about related objects, then dynamic attributes will not aid classification. The relevance of dynamic attributes can be gauged with a single "full knowledge" classification pass — where the true class labels of related objects are used to calculate the values of dynamic attributes. Such a test indicates the effectiveness of the dynamic attributes if the inferences made by the model were 100% accurate; the test reveals the ceiling accuracy for the chosen set of attributes. If the ceiling accuracy is not significantly higher than the floor accuracy (using only static attributes), iteration will produce no discernible effect.

Second, the data set must be sufficiently connected. An iterative approach uses relational structure to maximize the use of its inferences. The results of classification are spread through the relational structure via dynamic attributes, so if the data are sparsely linked, then there is less opportunity to make use of prior inferences. However, what constitutes "sufficient" linkage is not clear, and it may vary significantly across data sets. Both the degree of linkage, as well as the type of linkage, may affect the results of iterative classification. Further exploration is needed to determine the success of iterative classification for various types of relational structures.

Finally, there must be information present in the data to catalyze the iteration process. Initial classifications are made using only static attributes; therefore the classification model must have a way of making some initial inferences accurately. If none of the initial inferences are correct, then all subsequent predictions will be misled by those inferences that are accepted. This condition, combined with the first, implies the need for "islands of certainty."

Islands of certainty denote knowledge from which some, but not all, objects can be classified accurately, with high confidence. Examples of islands of certainty include a highly predictive static attribute that is missing in many instances but known for some, a static attribute for which some values are highly predictive of particular class labels but other values are not, or a partially labeled data set.

The inferences made from islands of certainty catalyze iterative classification, leading to correct dynamic attribute calculations and improving predictions about related objects. Without such islands, the performance of iterative classification may degrade. Future work should explore the extent of this degradation and determining the size, type and number of islands needed for successful iterative classification.

Experiments

Our experiments use a data set which records intrinsic and relational features of publicly traded corporations. The data are drawn from documents filed with the US Securities Exchange Commission (SEC). Due to the size of the database, we chose to work with data from only two industries, banks and chemical companies. Data are maintained separately for each industry in the SEC database, so substantial consolidation was needed to combine data from two industries.

The data consist of companies, their board members and officers, stockholders, contractors and subsidiaries. The data set contains 2142 central companies (892 chemical companies and 1250 banks). It also contains 18679 related companies: 5201 corporate owners, 969 contractors, and 12509 subsidiaries. Owners, contractors, and subsidiaries do not have the same intrinsic attributes as the banks and chemical companies, so we chose to represent then as separate objects. In addition to these objects, the data set also contains 25591 people who serve as officers and directors of the companies.

We selected a relatively simple task: to classify companies as to their industry, either bank or chemical, using both relational and intrinsic attributes. Classification of companies by type is a surrogate task intended to illustrate the potential of iterative classification in other domains with similar organizational structure, such as fraud detection or money laundering analysis. Iterative classification is not restricted to binary classification tasks. Because an SBC produces a posterior probability estimate for each class label, the approach could easily be used for classes with more than two labels. Multiple class labels, however, would make the queries for calculating and updating attribute values more complex, and complicate ROC curve analysis.

The data ontology is shown in figure 1. Nodes in the graph represent objects in the data set. Links in the graph correspond to possible relationships among objects. Italicized labels indicate link or object type. All other labels correspond to intrinsic data associated with the links and objects. A distinctive feature of this ontology is that companies are never linked to other companies directly; they are only linked indirectly through people, owners and contractors.

In our experiments, we used four attributes for each company: 1) the state of incorporation (static); 2) the number of subsidiaries (static); 3) whether the company is linked to more than one chemical company through its board members (dynamic); and 4) whether the company is linked to more than one chemical company through its insider owners (dynamic). Informal tests with additional attributes showed no substantial improvement in accuracy, so for efficiency reasons the attributes were limited to these four.

Sampling

Devising a disjoint training and test set was challenging. Partial sampling of linked data can bias statistical estimates of relational attributes (Jensen 1998). Fractional sampling of linkage in the data can produce under- and over-estimates of attributes that will reduce the effectiveness of an induction algorithm. SBCs assume that the distribution of features is comparable between training and tests sets, so their effectiveness depends on a sampling procedure that produces similarly linked training and tests sets. Also, because iterative classification involves inferences made about linked companies, a desirable sampling procedure would retain as much linkage to other companies as possible.

The sampling procedure used is similar to the exhaustive approach described by Jensen (1998). The process for creating two samples A & B from the set of all companies is given below.

This approach produces two disjoint subsets — the core of each sample. By definition companies in core A have no links to companies in sample B. Likewise, companies in core B have no links to companies in sample A (see figure 2). The resulting size of the cores depends on the degree of linkage in the data set. If the objects are highly linked then there will be very few objects in the core.

Because the success of iterative classification in the corporate data depends on linkage among companies, we removed all companies from the sample with no links to other companies. This improved the statistical power of our evaluation by focusing on the portion of the task to which iterative classification is most applicable. It also reduced the total number of companies in the data set to 1088. In order to increase the number of companies in the core of each sample, the definition of the core was relaxed. Because the only dynamic attributes used for classification involved links through people (insider owners or board members), the core objects were defined as those that have no links *through people* to companies in the other sample. Links to companies in the other sample through corporate owners and contractors however, were allowed. Core A therefore consists of those companies in sample A that have no links through people, to companies in sample B. The distribution of banks and chemical companies in both the samples and the cores are outlined in table 1.

Sampling Procedure

1. Initialize *X* to the set of all company objects.
2. Do until *X* is empty:
   a. Do until a company is placed in sample A:
      i. Randomly pick a company *x* and remove from *X*.
      ii. Gather all objects one link away from *x*.
      iii. If any of these objects is in sample B, discard *x*. Otherwise place *x* in sample A, along with all objects one link away from *x*.
   b. Do until a company is placed in sample B:
      i. Randomly pick a company *y* and remove from *X*.
      ii. Gather all objects one link away from *y*.
      iii. If any of these objects is in sample A, discard *y*. Otherwise place *y* in sample B, along with all objects one link away from *y*.
3. For all discarded companies, randomly place half in sample A and half in sample B.
4. Label all companies in sample A that have no links to sample B as objects in the core of sample A. Label sample B similarly.

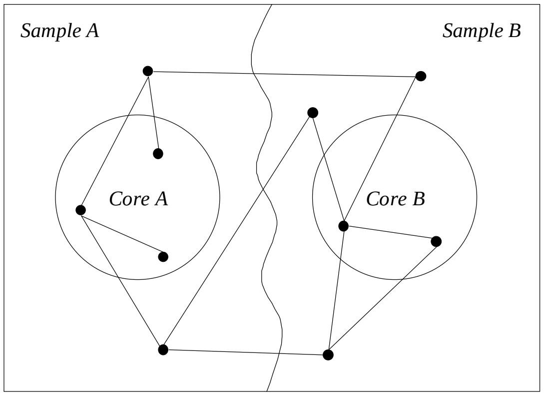

Figure 2: Example of indirect company linkage in samples

	Number of banks	Number of chemicals	Total number of companies
Sample A	230	316	546
Core A	170	113	283
Sample B	236	306	542
Core B	189	113	302

Table 1: Distribution of samples and cores

	% Accuracy on Core B	% Accuracy on Core A
Static	69.2	68.6
Iteration 1	72.2	78.1
Iteration 10	75.2	80.9
Full Knowledge	78.1	80.9

Table 2: Classification accuracies

Experimental Procedure

Using the two samples A and B we performed a two-fold cross validation test of iterative classification. The small number of objects in the resulting cores, when sampled for more than two sets, prohibited the use of more than two disjoint samples. The classifier was trained on a fully labeled sample A and then tested on sample B with 10 iterations. Because the 10th iteration has only 90% of the inferences available for dynamic attribute calculation, a final classification pass (11th iteration) was also included which used 100% of the inferred class labels.

During training, the dynamic attributes of sample A make use of some of the class labels in sample B but this does not include any of the companies in core B. When testing on sample B, the classifier makes inferences about all the companies in sample B; however, accuracy is measured only on the fully disjoint companies in core B. The companies of sample A must be fully labeled during the testing process in order to prevent biasing the attribute calculation of companies in sample B that are not in core B. In the second test, the classifier is trained on sample B and tested on sample A.

Results

Accuracy results for the two test sets are shown in table 2; accuracy refers to the rate of correct predictions made by the model for the objects in the test set. The “Static” accuracy results are from a single classification pass using only static attributes of the test set, where the values for the dynamic attributes are all missing. “Iteration 1” and “Iteration 10” are the accuracy results after the first and tenth iteration respectively. “Full knowledge” indicates the accuracy results of a single classification pass using all attributes, where the dynamic attributes are calculated with complete knowledge of the true class labels of all related companies.

McNemar’s test (Sachs 1982) was used the compare the difference in classification accuracy between the 1st iteration and 10th iteration. The McNemar statistic tests the null hypothesis that the differences in frequencies of correct and incorrect classifications in each iteration represent random variations in the class labels. Combining the results from both cross-validation trials, the value of the McNemar statistic was 5.558, which indicates the difference in classifications from the 1st to the 10th iteration is significant at the 2% level.

Accuracy results over the course of iterations for each cross-validation run are shown in figure 3. Accuracy increases steadily throughout the classification procedure except for a drop in the final pass (11th iteration). Dynamic attribute calculations in the final pass include the inferences for which the SBC model is most uncertain — the bottom 10%. This suggests that an improvement in classification could be achieved by the use of a threshold for accepting predictions, instead of accepting the top percentage.

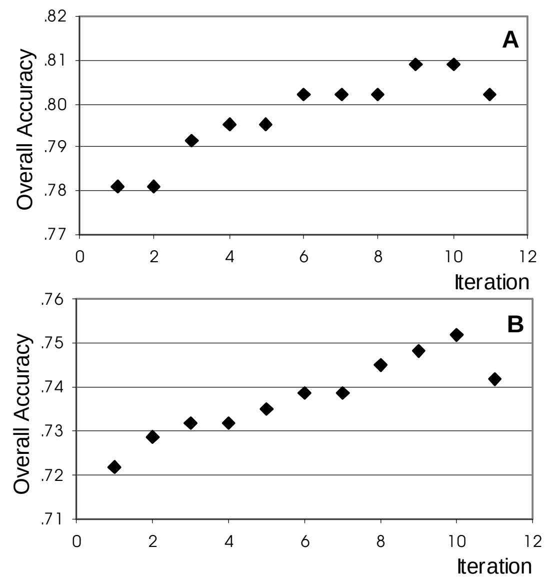

Figure 3: Accuracy results on core objects for each iteration

Because accuracy maximization assumes equal misclassification cost for false positive and false negative errors, the use of classification accuracy as a primary metric to compare classifiers is not always an indication of superior performance for other costs and class distributions (Provost, Fawcett and Kohavi 1998). Receiver Operating Characteristic (ROC) analysis is an alternative means to evaluate the error tradeoffs associated with a given model.

ROC curves for the SBC models on the 1st and 10th iterations are shown in figure 4. The curves show the predictive ability of each model across all possible error costs and class distributions. Each SBC model is represented in ROC space by a curve corresponding to its true positive rates and false positive rates (TP, FP), as the probability threshold between classes is varied between zero and one.

An ROC curve maps a classifier's performance as the confidence threshold for acceptance of its predictions is varied between the extremes of accepting no classifications to accepting all classifications. If a model dominates the ROC space it can be regarded as the "best" predictive model for all domains, no matter what the cost and class distributions are in the test environment.

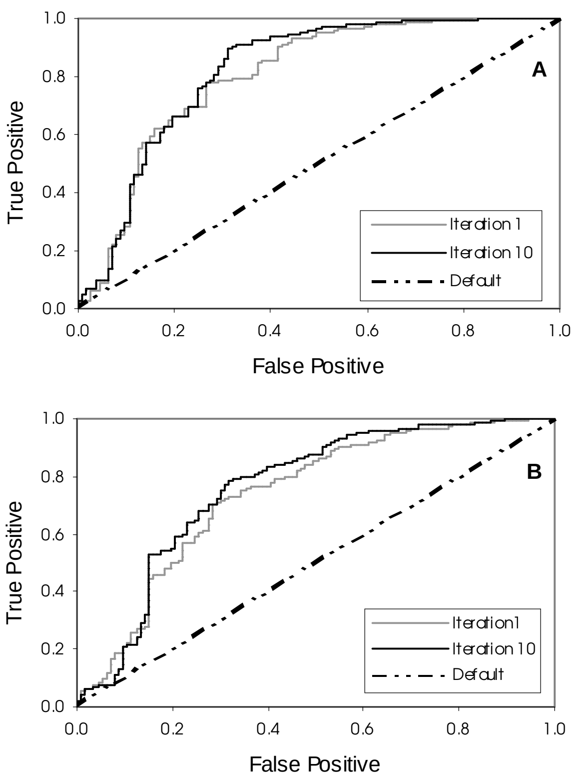

Figure 4: ROC Curves for classification on sample core objects

Discussion

The accuracy results imply some interesting conclusions regarding iterative classification in this domain. First, our window for improvement in this data set is quite small, with approximately a 10% difference between the floor and ceiling accuracies. The floor accuracy can be lowered artificially by dropping static attributes. This was attempted but the iterative approach failed without the inclusion of both static attributes. This indicates the importance of having strong static attributes as islands of certainty from which to jumpstart the iterative process. The limited variety of links in the data set constrained the number of potentially predictive dynamic attributes, so raising the ceiling accuracy was difficult.

Next, the improvement of accuracy in the 1st iteration compared to the static approach is noteworthy. The difference between classification in the 1st iteration and the static test is that during the 1st iteration some dynamic attributes values are known. For companies with less than two links to other companies through people, we can return a value of false for the dynamic attributes without any knowledge of the company type. This suggests that dynamic attributes whose value can be determined with certainty from a small amount of evidence may be quite helpful to the iterative process.

Also, it is worth mentioning that in the second trial on Core A, iterative classification was able to match the accuracy of classification with full knowledge. This shows the power of iterative classification to classify as if it had full knowledge of the surrounding environment.

Finally, the ROC curves show that the 10th iteration performs better than, or equal to, the 1st iteration for most thresholds. However, the ROC curves show that the primary effect of iteration occurs late in the curve when the probability of a company being a bank is relatively low. This may indicate that dynamic attributes are more helpful in the case of predicting chemical companies and do little to increase the probabilities associated with predictions of banks.

Related Work

Previous work of the WebKB project investigated classification in a relational context (Craven et al. 1998). WebKB used both SBCs and FOIL, a greedy covering algorithm for learning function-free Horn clauses, to label web pages automatically. Relationships among pages, as encoded by their hyperlinks, are used along with intrinsic attributes to improve classification accuracy.

"Co-training" is an iterative approach to learning models (Blum and Mitchell 1998, Mitchell 1999) that was applied to the WebKB labeling task. Experiments show that a large number of unlabeled instances can be used to boost the performance of a learning algorithm when only a small set of labeled instances is available. Multiple classifiers are learned on independent sets of attributes, from a common set of training examples. Each classifier is run and its most confidently predicted positive and negative instances are added to the training set. The classifiers are relearned with the larger, augmented training set, and the process is repeated. By using the same training data, the classifiers each profit from the predictions of other classifiers. Co-training is tested in a relational context; however, it can be applied to attribute-value data as well. This method uses iteration for learning models instead of using iteration in the application of learned models, as does iterative classification.

Slattery (2000) has investigated using relational information in the test set to classify web pages more accurately. FOIL-HUBS is an extension of FOIL inspired by the Hubs & Authorities algorithm (Kleinberg 1998). FOIL-HUBS identifies the existence of hubs for each target class (e.g., student-hubs point to many student pages) and hub weights contribute to the probability that pages pointed to by the hubs are of a particular class. FOIL-HUBS employs an iterative classification scheme to predict class labels and estimate hub weights, which is similar to our own algorithm for iterative classification, but it is limited to domains where hub nodes exist. In contrast, our work represents an initial attempt to provide a uniform framework for the calculation and use of a wider range of dynamic attributes, albeit within a simpler model representation (SBCs as opposed to function-free Horn clauses).

Freidman et al. (1998) have investigated the use of a relational framework to make sophisticated probabilistic inferences. They have shown how to learn probabilistic relational models (PRMs) from relational databases. PRMs extend the applicability of Bayesian networks techniques (Heckerman 1995), and allow the properties of an object to depend probabilistically on both intrinsic and relational attributes. As currently applied, PRMs do not use initial inferences to inform later inferences about related objects. However, PRMs could be used in the same way that SBCs are used for iterative classification in the work reported here.

The Expectation-Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977) is similar to in spirit to iterative classification, but it addresses a somewhat different problem. The EM algorithm uses a two-step iterative procedure to find the maximum-likelihood estimate of the parameters of an underlying distribution (a model) from a data set containing incomplete or missing data (Bilmes 1998). The first step of EM (the "expectation" step) finds the expected value of missing data values, given the current model. The second step of EM (the "maximization" step) finds the maximum-likelihood model, given the inferred data. After replacing the current model with the new model, the process repeats. In contrast to iterative classification, EM readjusts the model in the second step, rather than adjusting the values of attributes that serve as inputs to the model. Thus, it is a method of learning a model given attribute-value data, rather than a method of applying a learned model to relational data.

Kleinberg (1998) developed an iterative algorithm, called Hubs & Authorities, for Web searching based on the network structure of hyperlinked pages on the Web. The algorithm uses a graph structure, with nodes corresponding to web pages and directed links indicating the presence of hyperlinks between pages. Given the task of identifying authoritative pages, two mutually reinforcing attributes are defined: hub weight and authority weight. The weights are calculated in an iterative fashion by feeding the values of one attribute into the calculations of the other. The iterative nature of this algorithm is similar to our approach in that it maintains and updates attribute values throughout the procedure. However, the algorithm assumes the values of both attributes are known for each instance and starts by assigning equal weights to all pages. It does not use a predictive model to assign weight values.

Conclusions and Future Work

A number of conclusions can be drawn from this work about the potential of iterative classification. We have shown that there is an opportunity to use relations in data to increase classification accuracy, and that an iterative approach exploiting this opportunity can produce a significant improvement in accuracy for a binary classification task in the corporate data set.

We have outlined several necessary conditions for successful application of iterative classification. For iterative classification to improve on a static approach, a data set should exhibit the following characteristics: insufficient predictive power from static attributes and useful dynamic attributes, rich relational structure, and islands of certain knowledge from which to jump start the iterative process. Expansion and formal verification of these ideas is an important area for further investigation.

In addition to presenting opportunities for discovery, relational data also offer several challenges. Devising a sampling procedure that does not bias statistical estimates of relational attributes is a difficult task. As the relational data structure becomes more complex, our opportunities for improving classification increase, but so do the challenges of sampling. Future work would be aided by the use of naturally disjoint data sets with similar distributions such as the university web sites used by Slattery (2000).

Formulating useful dynamic attributes is also challenging. It is difficult to define the value of a dynamic attribute when some, but not all of the related class labels have been inferred. Because the classifier is trained on full knowledge, dynamic attribute values expressing partial knowledge can bias or mislead the predictions of the classifier. A few incorrect inferences could have a “snowball effect,” with the dynamic attributes cascading the mistakes throughout the test set. For this reason it is important to use dynamic attributes whose values are either known with complete certainty or not at all. *Threshold* attributes are a good example of this type of “robust” attribute, where the value is known as soon as a particular value threshold is exceeded. Both dynamic attributes used in this experiment are examples of threshold attributes. Future work includes both establishing the effects of threshold attributes on iterative classification, and determining other types of robust attributes.

Attributes that combine probabilistic evidence of all related class labels are a potential alternative to threshold attributes. Instead of accepting the top percentage of predictions, or those exceeding a threshold, the algorithm would accept all predictions. The values of these probabilistic attributes are then determined by a combination of the probabilities associated with the inferred class labels of related objects. As the certainty of predictions change over the course of iterations, the attribute values could be dynamically updated. This is an area that requires additional exploration.

A potential pitfall of the specific variety of iterative classification explored here is that SBCs often produce biased probability estimates. SBCs are known to produce optimal class predictions in a wide variety of domains; however, SBC probability estimates are biased except under conditions of attribute independence. Future work includes exploring iterative classification with other methods that produce more accurate probability estimates such as Bayesian networks or PRMs (Freidman et al. 1999). We will also investigate the use of a threshold for accepting predictions instead of accepting a percentage determined by the number of iterations.

Another direction for future work involves extending the iterative procedure for prediction of multiple object types by simply combining the results of multiple classifiers. Each classifier would make use of the dynamic attributes filled in through the efforts of the other classifiers. In this sense the classifiers would collaborate with each other to improve accuracies for both classification tasks. Caruana (1997) has investigated the collaboration of multiple models for learning under the hypothesis that multiple, related learning tasks share the same representation, and learning one helps with learning another. A relational approach would be similar but would involve the collaborative application of models instead.

Acknowledgments

Matt Cornell and Hannah Blau provided important contributions to the work reported here. Foster Provost and James Allan provided valuable comments and suggestions on earlier drafts of this work. Bob Tenney, Kendra Moore, Chris White, and Jack Chiang of AlphaTech Corporation provided valuable assistance with obtaining and cleaning the data used in our experiments.

This research is supported, in part, by a University of Massachusetts Faculty Research Grant and by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Office of Scientific Research (AFOSR) under contract F49620-97-1-0485. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of the University of Massachusetts, DARPA, AFOSR, or the U.S. Government.

References

Bilmes, J. (1998). A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. TR-97-021. International Computer Science Institute, Berkeley, California.

Blum, A. and T. Mitchell. (1998). Combining labeled and unlabeled data with cotraining. *Proceedings of the 11th Annual Conference on Computational Learning Theory*. ACM. pp. 92-100.

Caruana, R. (1997). Multitask learning. *Machine Learning* 28: 41-75.

Craven M., D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery (1998). Learning to extract symbolic knowledge from the World Wide Web. *Proceedings of the 15th National Conference on Artificial Intelligence*. pp. 509-516.

Dempster, A., N. Laird, and D. Rubin (1977). Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*. 39: 185-197.

Friedman N., L. Getoor, D. Koller, and A. Pfeffer (1999). Learning probabilistic relational models. *Proceedings of the International Joint Conference on Artificial Intelligence*. pp. 1300-1307.

Heckerman, D. (1995). A tutorial on learning with Bayesian networks. Microsoft Research Technical Report MSR-TR-95-06.

Jensen, D. (1997). Prospective assessment of AI technologies for fraud detection: A case study. *AAAI Workshop on AI Approaches to Fraud Detection and Risk Management.* pp. 34-38.

Jensen, D. (1998). Statistical challenges to inductive inference in linked data. *Seventh International Workshop on Artificial Intelligence and Statistics*.

Kleinberg, J. (1998). Authoritative sources in a hyperlinked environment. *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*. pp. 668-677.

Mitchell, T. (1999). The role of unlabeled data in supervised learning. *Proceedings of the 6th International Colloquium on Cognitive Science*.

Provost, F., and T. Fawcett, and R. Kohavi (1998). The case against accuracy estimation for comparing induction algorithms. *Proceedings of the Fifteenth International Conference on Machine Learning*. pp. 445-553.

Sachs, L. (1982). *Applied Statistics: A Handbook of Techniques*. New York, NY: Springer-Verlag. pp. 363-364.

Slattery, S. (2000). Unsupervised structural inference for web page classification. To appear in: *17th International Conference on Machine Learning*.

Wasserman, S., and K. Faust (1994). *Social Network Analysis: Methods & Applications.* Cambridge, UK: Cambridge University Press.