Avoiding Bias when Aggregating Relational Data with Degree Disparity

David Jensen JENSEN@CS.UMASS.EDU
Jennifer Neville JNEVILLE@CS.UMASS.EDU
Michael Hay MHAY@CS.UMASS.EDU
Knowledge Discovery Laboratory, Computer Science Dept., Univ. of Massachusetts Amherst, Amherst, MA 01003 USA

Abstract

A common characteristic of relational data sets —*degree disparity*—can lead relational learning algorithms to discover misleading correlations. Degree disparity occurs when the frequency of a relation is correlated with the values of the target variable. In such cases, aggregation functions used by many relational learning algorithms will result in misleading correlations and added complexity in models. We examine this problem through a combination of simulations and experiments. We show how two novel hypothesis testing procedures can adjust for the effects of using aggregation functions in the presence of degree disparity.

1. Introduction

Many current techniques for inductive logic programming and relational learning use aggregation functions. These functions (e.g., AVG, MODE, SUM, EXISTS, COUNT, MAX, MIN) are used to summarize the complex and varying relational structure found in many learning tasks. For example, molecules have varying numbers of atoms and bonds, web pages have varying numbers of incoming and outgoing links, and movies have varying numbers of actors and producers.

If learning algorithms use aggregation functions without adjusting for the underlying structure of relational data, they can produce misleading models. In particular, if the number of items to be aggregated varies systematically with the target variable, then applying any one of a large set of aggregation functions will lead to an apparent correlation between the aggregated variable and the target variable.

The results in this paper complement previous results (Jensen & Neville, 2002) showing that concentrated linkage and autocorrelation can bias feature selection in algorithms for relational learning. Together, these results show the perils inherent in simple approaches to propositionalizing relational data, as well as other approaches that ignore the correlation between attribute values and relational structure.

1.1 An example of feature evaluation

Consider the problem of learning to predict the box office success of movies based on characteristics of the actors in the movies. Some fragments of a relevant data set are shown in Figure 1. Each movie is characterized by a binary class label indicating whether the movie made more than $2 million in its opening weekend. Each movie is linked to the set of actors that appear in the movie, and each of those actors are characterized by a set of twenty discrete attributes (e.g., the gender of the actor, whether the actor has won an award, etc.).

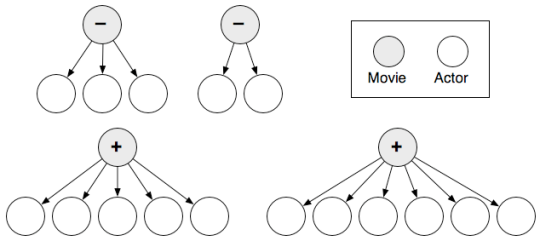

Figure 1. Example fragments of relational data about movies.

For many learning algorithms, a subtask of this problem is to determine whether any of the actor attributes predict movie success. Because each movie has different numbers of actors, relational learning algorithms often examine aggregations of the attribute values on actors. For discrete attributes, we might examine whether a particular value is the MODE of the values of a given actor attribute or whether a particular value EXISTS among all the possible values. MODE and EXISTS are often called *aggregation functions*, and such functions are common to many languages for handling relational data (e.g., SQL). In this case, the number of possible features, $|features| = kna = 200$, where k is the number of attributes (e.g., 20), n is the number of values per attribute (5), and a is the number of aggregation functions (2).

Are any of these features useful for prediction? That is, do any of them perform better than would be expected by chance alone? Happily, 79 of the 100 EXISTS features on actors appear to be useful in predicting the box office success of movies, using a standard chi-square test of statistical significance (α=0.05, adjusted for 200 tests). Ten of the 100 MODE features appear useful.

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

Unfortunately, these results demonstrate an important flaw in the evaluation of features in relational data. The attribute values discussed above were generated randomly, without respect to the box office receipts of the corresponding movies. Specifically, we simulated actor attributes by generating five-valued discrete attributes with the probability distribution {0.40, 0.30, 0.20, 0.05, 0.05}. Thus, the values of actor attributes should tell us nothing about the expected box office receipts of movies. Instead, the aggregated values of actor attributes reflect a difference in the *structure* of the movie data. This structure was not generated randomly, but reflects the actual structure of the Internet Movie Database (IMDb).

1.2 Heterogeneous structure

In the IMDb (www.imdb.com), the number of actors associated with any given movie varies systematically with class label. As shown in Figure 2, successful movies tend to have more actors than unsuccessful movies. Though subtle, the effect is highly significant ($p<2.2e\text{-}16$), if we compare movies based on whether they gross more than $2 million in their opening weekend. We call this systematic difference *degree disparity*.

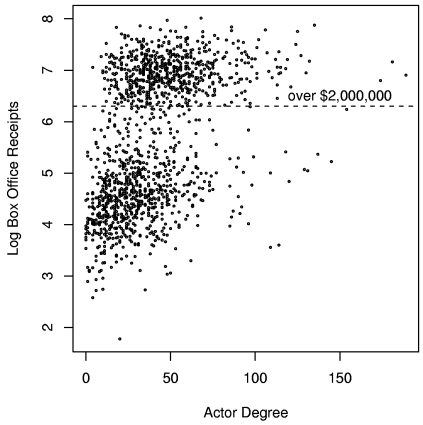

Figure 2. Actor degree varies with box office receipts

Given degree disparity, nearly *any* aggregated attribute can show apparent correlation with the class label. Some of these effects are obvious. For example, the SUM of a continuous attribute such as actor age will be much higher for movies with many actors than for those with few. Other effects are relatively clear when you consider the effects of degree disparity. For example, the MIN or MAX values of a particular continuous attribute of actors will tend to be larger, given the opportunity to select from a larger number of actors. Similarly, the probability that a particular value EXISTS will be higher, given a larger number of actors.

Given that we can recognize degree disparity, can we account for its effects? One option is to adjust the calculation of chi-square to account for the effects of degree disparity. We discuss the details of this adjustment in more detail in Section 5, but the effect of making this adjustment is shown in Figure 3. From right to left, the figure shows the sampling distribution of chi-square for a conventional calculation, the sampling distribution for a corrected calculation, and a theoretical sampling distribution. Clearly, the corrected distribution is a far better approximation to the theoretical sampling distribution.

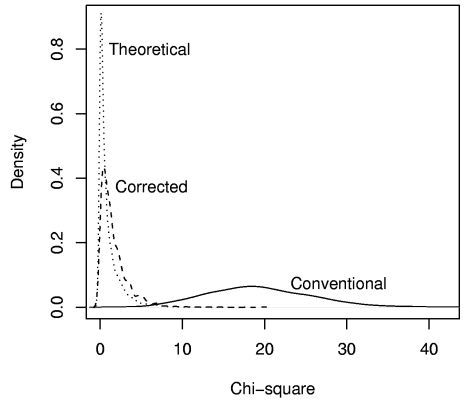

Figure 3. Theoretical, corrected, and conventional distributions of the chi-square statistic given the actor degree disparity in the IMDb data.

In the next two sections of this paper, we discuss how aggregation functions are used in relational learning algorithms, and we define degree disparity. The next section details the misleading correlations that can result when aggregation functions are used in the presence of degree disparity. Then we present two types of significance tests that can be used to adjust for the biases introduced by degree disparity and present experimental evidence that these corrections result in more understandable models. Finally, we conclude with some discussion and pointers to future work.

2. Aggregation in Relational Learners

Many algorithms for relational learning use aggregation functions.[1] Perhaps the most common approach is to use EXISTS in an explicit or implicit manner, although the full range of aggregation functions are used by some techniques. Specifically, learning algorithms can apply aggregation as either a pre-processing step or as part of the actual learning procedure.

Some techniques "propositionalize" relational data and then apply a conventional non-relational learning algorithm. This preprocessing often uses aggregation functions. For example, Krogel and Wrobel (2001) use AVG,

[1] Aggregation is one of many characteristics of the knowledge representations employed in inductive logic programming and relational learning. Other characteristics of knowledge representation and reasoning systems that are not discussed in the paper include variable bindings, functors, relational skeletons, and slot chains.

MIN, MAX, SUM, and COUNT functions as part of their RELAGGS approach. RELAGGS propositionalizes relational data into features that are then supplied to either C4.5 or an algorithm for learning a support-vector machine.

Other algorithms have been specifically designed for relational learning. These techniques often employ aggregation functions directly in their model representations. For example, probabilistic relational models (PRMs) use MODE, AVG, MEDIAN, MAX, MIN, and SIZE. PRMs (Getoor, Friedman, Koller & Pfeffer, 2001) learn a form of graphical model that represents the joint probability distribution over individual and aggregated values of particular attributes. Similarly, Knobbe, Siebes, and Marseille (2002) present a general framework for aggregation, and demonstrate the framework in the context of rule learning. Their approach uses COUNT, MIN, MAX, SUM, and AVG. Finally, our own work on learning relational probability trees (Neville, Jensen, Friedland, & Hay, 2003) creates dichotomous divisions within a tree using COUNT, PROPORTION, MIN, MAX, SUM, AVG, MODE, and EXISTS.

Other approaches to learning in relational data make heavy use of EXISTS. For example, Popescul, Ungar, Lawrence, and Pennock (2002) adapt logistic regression to the problem of relational learning, using EXISTS to create logical features that serve as independent variables in the regression equation. Blockeel and De Raedt (1998), use EXISTS to create logical features for induction of relational classification trees. Several other systems use similar techniques (e.g. Kramer, 1996).

Although the use of aggregation functions is a frequent technique in relational learning, some approaches use other techniques for handling the varying structure of relational instances. For example, Lachiche and Flach (2002) discuss the use of set-valued probability estimators, and Emde and Wettschereck (1996) use an instance-based approach to learning that calculates a similarity measure between cases represented in first-order logic. However, approaches that eschew explicit aggregation are relatively rare. The dominant approach appears to be aggregating values either in pre-processing or as part of the actual learning procedure.

3. Degree Disparity

What is degree disparity? For purposes of this paper, we define degree disparity as "systematic variation in the distribution of the degree with respect to the target variable." For example, actor degree disparity exists with respect to the box office receipts of movies if successful movies tend to have more (or fewer) actors than unsuccessful movies. Indeed, as shown in Figure 2 above, this is true for the IMDb. The effects of aggregation and degree disparity are quite general and could affect many learning tasks. However, for simplicity, this paper focuses on their effects with respect to classification.

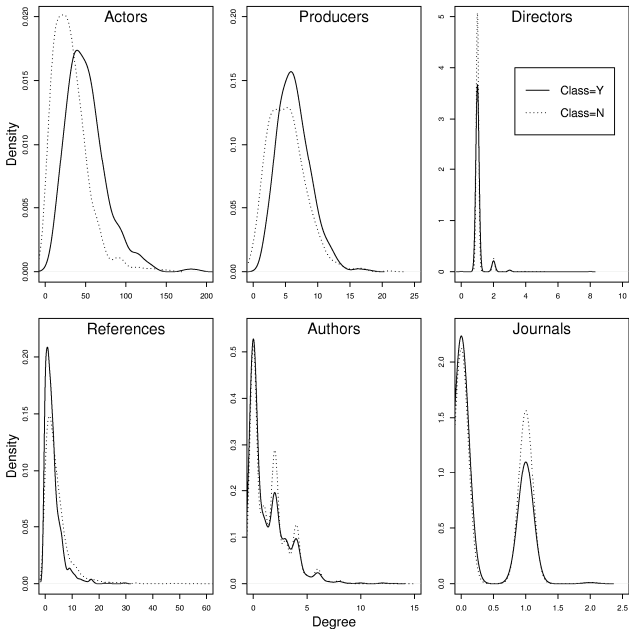

Figure 4. degree disparity in IMDb and Cora data sets

Degree disparity is a common characteristic of relational data. Figure 4 shows the degree distributions for two relational data sets commonly used for evaluating relational learning algorithms. The first data set is drawn from the IMDb. We gathered a sample of 1382 movies released from January 1996 to September 2001. The data set consisted of all movies released in the United States during that time period with opening weekend receipt information. Other time periods and geographic regions have much sparser attribute information. In addition to movies, the data set contains objects representing actors, directors, producers, and studios. In total, the data set contains approximately 46,000 objects and 68,000 links. The learning task was to predict movie opening-weekend box office receipts. We discretized the attribute so that a positive label indicates a movie that garnered more than $2 million in opening-weekend receipts ($P(+)=0.45$). Figure 4 (top) shows degree disparity of the three types of entities. We tested those differences using the Kolmogorov-Smirnoff (K-S) distance, a measure of the maximum difference between two cumulative probability distributions (Sachs, 1982). K-S distance has a known sampling distribution parameterized by sample size, and thus can be used to test whether two degree distributions are drawn from the same parent distribution. The degree disparity with respect to actors and producers is statistically significant ($p<0.0001$); the degree disparity with respect to directors is not significant. Although the degree disparity for actors and producers appears small, it has large effects, as we showed in the introduction and will show later.

The second data set is drawn from Cora, a database of computer science research papers extracted automatically from the web using machine learning techniques (McCallum, Nigam, Rennie & Seymore, 1999). We se-

lected the set of 4330 papers with topic "Artificial Intelligence/Machine Learning" along with their associated authors, journals, books, publishers, institutions and cited papers. The resulting collection contains approximately 11,500 objects and 26,000 links. Machine learning papers are further subdivided into seven categories (e.g., "Theory", "Reinforcement learning"). The prediction task was to identify whether a paper's topic is Neural Networks ($P(+)=0.32$). The degree disparity for all three types of entities—references, authors, and journals—are statistically significant ($p<0.0001$). Although the degree disparity for references, authors, and journals appears remarkably small, it can have large effects, as we will show in later sections.

4. Apparent Correlation

Given degree disparity, the use of aggregation functions can lead to correlation between the aggregated feature and the class label even if the individual attribute values are independent of the class label. This is true regardless of which of a large class of aggregation functions are used—COUNT, EXISTS, SUM, MAX, MIN, AVG, MODE—although the amount of correlation depends on the aggregation function employed, the extent of degree disparity, and the distribution of the attribute being aggregated.

Such correlation reflects degree disparity alone, and it can have strong negative effects on model learning. First, this type of correlation produces models that are easily misunderstood as representing correlation between the attribute values themselves and the class label. At the very least, correlation due to degree disparity introduces an added level of indirection into a user's understanding of an induced model. Second, correlation due to degree disparity can vastly increase the number of apparently useful features, making induced models much more complex. This added complexity makes models correspondingly much less understandable and much less computationally efficient to use. For many techniques, particularly graphical models such as PRMs, the identification of conditional independence among attributes is a central goal, because it improves both interpretability and computational efficiency. Both these goals are impaired by added complexity. In addition, the large number of surrogate features for degree will cause some types of models to spread the credit for the predictive ability of degree across a large number of other features, making it appear that many features are weakly predictive rather than the truth—that a single structural feature (degree) is strongly predictive.

4.1 Apparent Correlation in Theory

The effects of degree disparity are relatively straightforward to prove for certain, restricted classes of attribute distributions. In the interests of brevity, we omit detailed proofs, but provide informal sketches for three types of aggregation functions.

The probability that a given discrete value EXISTS changes strongly with degree. For example, if we assume that the genders of all actors in a given movie are mutually independent, then the probability of a given number s of female actors is determined by the binomial distribution (Sachs, 1982). That is, the probability distribution of the random variable S is $b(s;t,p)$, where t is the total number of actors in a movie and p is the probability that a given actor will be female. The cumulative binomial distribution increases monotonically with increasing t. Similarly, aggregated features using AVG can be influenced by degree. Based on Bernoulli's theorem (or the weak law of large numbers), for a given distribution with mean μ, the probability that the average value of a set of independent draws from that distribution will exceed a given threshold x, where $x > \mu$, decreases as sample size increases. Finally, The probability of achieving a particular MAX or MIN also varies with the number of items t (Jensen & Cohen, 2000).

4.2 Apparent Correlation in Practice

Do apparent correlations between aggregated attributes and a class label happen in practice? Specifically: 1) Will actually observed levels of degree disparity produce significant correlations in attributes whose values are otherwise uncorrelated with the class label; and 2) Will those correlations exceed the correlations of simple features based on degree as well as other features unaffected by degree disparity? Below, we present evidence for positive answers to both questions.

To illustrate the bias caused by degree disparity, we took the existing relational structure of the IMDb data and generated attributes whose values were uncorrelated with the class label. On the data set of 1382 movies, we added a pair of attributes (one discrete and one continuous) to each object related to a movie (actors, directors, and producers). The attributes' values were uniformly distributed, and independent of the class label.

We generated 300 such data sets and recorded the chi-square scores for each aggregated feature. Figure 5 shows the distributions of these scores. The top plot shows the distribution of scores for features formed from the two random attributes on actors. The bias is highest for the aggregation functions SUM and EXISTS and the bias tends to decrease as degree disparity decreases. As shown in Figure 4, actors have high degree disparity, producers moderate disparity and directors have no significant degree disparity.

To test the effect of degree disparity on feature selection, we ranked all features and then applied both a conventional chi-square test and a randomization test (described in Section 5) to assess the statistical significance of the association between the given feature and the class label ($\alpha<0.05$, adjusted for multiple comparisons). Figure 6 shows ranked scores for all features deemed significant based on the conventional test. Each bar corresponds to a

feature, and its length indicates the chi-square score of the feature. Dark shading indicates that the feature was also deemed significant using a randomization test. The two methods produce very different results. In the IMDb data, a randomization test eliminates the top-ranked feature, and in Cora, it eliminates the vast majority of features.

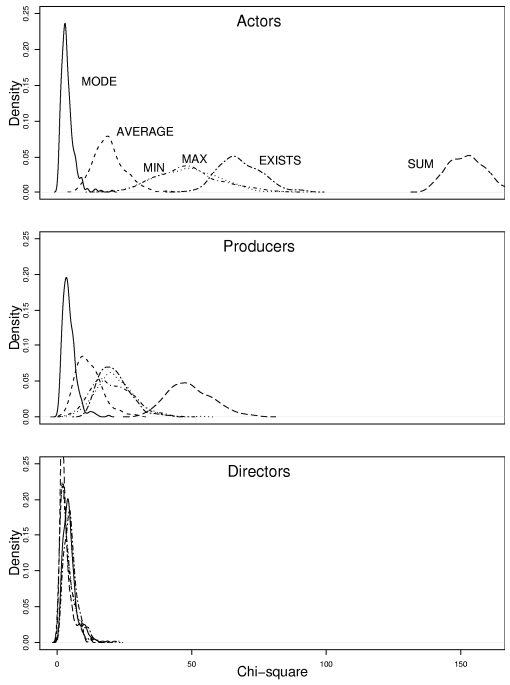

Figure 5. Simulation results for different types of attributes

5. Hypothesis Tests

We have devised two alternatives to traditional hypothesis tests that can adjust for the effects of degree disparity.

5.1 Traditional Tests

Relatively simple adjustments can be made to standard hypothesis tests that account for the effects of degree disparity. The introduction contained one example of this type of test—a modification of a standard chi-square test.

The chi-square statistic is the summation of normalized squared deviations from expected values. That is:

$$\sum_i \frac{(o_i - e_i)^2}{e_i}$$

where o_i is the actual value and e_i is the expected value. Given a value of this statistic, we can compare it to a known sampling distribution.

For example, the contingency table shown in Figure 7a summarizes the relationship between a feature value x and a class label y, where $x \in \{T,F\}$ and $y \in \{+,-\}$. Based on Figure 7a, we can calculate the expected values for each cell under the assumptions that the class label and feature value of each instance are independent and that the data instances are independent. Given actual counts (Figure 7a) and expected counts (7b), we can calculate the probability of actual counts at least as extreme as those observed under the null hypothesis of independence (p = 0.003).

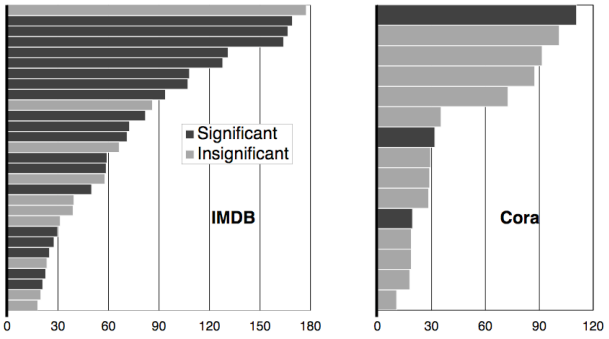

Figure 6. Histograms of ranked scores of features in two relational data sets. Bar length indicates the raw chi-square score. Shading indicates whether the feature is significant.

	+	−
T	11	3
F	4	12

(a)

	+	−
T	7	7
F	8	8

(b)

	+	−
T	10.7	3.3
F	4.3	11.7

(c)

Figure 7. An example contingency table (a), expected cell counts (b), and expected cell counts with degree disparity (c).

As we showed in Section 4, degree disparity can introduce dependence between class labels and feature values, thus violating the first of these assumptions. However, given a particular empirical distribution of degree for each class, we can calculate the expected feature values, given only the dependence introduced by degree disparity. For example, we can calculate expected values for the feature *COUNT(actor.gender=female)>2* with respect to movies. The overall distribution of actor.gender in our sample of movies is 66% male and 34% female. To calculate the table of expected values, we assume that each attribute value is independent of any other, and use the cumulative binomial distribution to determine the probability distribution over the possible attribute values for each movie. For a movie with 10 actors, the probability distribution for the feature values {T,F} is {0.716,0.284}; for a movie with 5 actors, the distribution is {0.220,0.780}. By summing the fractional counts across all instances, we can obtain a table such as the one in Figure 7c. Given these expected values, the probability of obtaining a table such as 7a (or a more extreme table), under the null hypothesis of attribute value independence, is large (p = 0.813). This method was used to calculate the corrected distribution in Figure 3.

This approach to producing a chi-square score "factors out" degree disparity. It is theoretically justified, computationally efficient, and often simple in practice. However, it assumes that each value being aggregated is independent, and that attribute values are independent of degree. Both assumptions are violated in practice. In addition, it is difficult to calculate for some combinations of aggregation function and attribute distribution.

5.2 Randomization Tests

Randomization tests provide an alternative method for hypothesis testing under the assumption of degree disparity. A randomization test (also called a *permutation test*) is a type of computationally intensive statistical test (Edgington, 1980). Randomization tests generate many data sets—called pseudosamples—and use the scores derived from these pseudosamples to estimate a sampling distribution. Each pseudosample is generated by randomly permuting the values of one or more variables in the original data. Each unique permutation of the values corresponds to a unique pseudosample. A score is then calculated for each pseudosample, and the distribution of these randomized scores approximates the sampling distribution for the score calculated from the actual data.

To construct pseudosamples in relational data with degree disparity, we permute the assignment of attribute values to entities across the entire data set prior to aggregation. Thus, each entity in a pseudosample (e.g., an actor) will be assigned a random attribute value (e.g., gender) drawn without replacement from the multiset of all such values in the real data. Then, the values are aggregated (e.g., *MODE(actor.gender)*) and the association between the aggregated feature (e.g., *MODE(actor.gender)=F*) and the class label in the pseudosample is scored using a conventional chi-square statistic. Note that this calculation is made without the adjustments discussed in Section 5.1. The chi-square statistic is calculated as if degree disparity does not introduce any correlation between the feature values and the class labels.

The set of scores—one per pseudosample—approximates the sampling distribution of chi-square under the null hypothesis, given the amount of degree disparity present in the actual data. In contrast, the procedure discussed in Section 5.1 alters how the chi-square statistic itself is calculated, adjusting the value of the statistic so that a known sampling distribution can be used to test the statistical significance of the resulting value.

As with the previous approach, this approach to hypothesis testing "factors out" degree disparity. Like the adjusted chi-square calculation, randomization tests are both theoretically justified and practically simple. However, randomization tests are computationally intensive, typically generating and evaluating hundreds of pseudosamples. While this only introduces a constant factor increase in computation time, the practical impact can be large, particularly if the hypothesis test constitutes an inner loop of a learning procedure. What countervailing benefits offset the disadvantage of added computation?

Randomization tests can be used to adjust for a much broader range of statistical effects than the modified chi-square calculation presented in Section 5.1. For example, we have developed randomization tests to adjust for the effects of autocorrelated class labels in relational data (Jensen & Neville, 2002). Autocorrelation violates the other assumption of the traditional chi-square test mentioned in the previous section; autocorrelation means that individual instances are not independent. In addition, we have developed randomization tests to adjust for the effects of other biases in learning algorithms (Jensen & Cohen, 2000). The same randomization test can be used to adjust for all of these effects simultaneously, so it is preferable in cases where all effects are present.

6. Experiments

To examine the practical effects of degree disparity and the effectiveness of randomization tests in adjusting for those effects, we applied an algorithm for learning relational probability trees (Neville et al., 2003).

6.1 Learning algorithm

Relational Probability Trees (RPTs) extend probability estimation trees (Provost & Domingos, 2000) to a relational setting. The RPT algorithm constructs a probability estimation tree that predicts a target class label given: 1) the attributes of the target object; 2) the aggregated attributes of other objects and links in the relational neighborhood of the target object; and 3) graph attributes that characterize the structure of relations (e.g., degree). We selected RPTs for experimentation because they select a subset of all features and because the recursive partitioning paradigm presents a set of simple univariate hypothesis tests rather than more complex multivariate tests.

The RPT learning algorithm searches over a space of binary relational features. The algorithm considers the attributes of different related object or link types and multiple methods of aggregating the values of those attributes, creating binary features from the aggregated values. For example, the algorithm considers features such as *AVG(actor.age)>25* for numeric attributes such as *actor.age*, and features such as *MODE(actor.gender)=Male* for nominal attributes such as *actor.gender*. The algorithm also searches over degree features that count the number of items in each relation (e.g., *DEGREE(actor)>6*). The algorithm uses Bonferroni-adjusted chi-square tests of significance to select features (Jensen & Cohen, 2000). All the experiments reported in this paper used a Bonferroni-adjusted α value of 0.05 as the stopping criteria.

In order to separate the effects of the randomization tests from the rest of the RPT learning algorithm we included a conventional tree learner in the evaluation. Following the approach of Krogel and Wrobel (2001), we generated

propositional data sets containing all the binary features considered by the RPT and supplied these data to C4.5. All experiments reported in this paper used the Weka implementation of C4.5 (Witten & Frank, 1999).

6.2 Classification tasks

Our first task (RANDOM) uses a subset of the IMDb data described in Section 3. Due to limitations of our randomization procedure, which can only randomize among data sets with non-zero degree, we selected the set of 1364 movies with at least one actor, director, studio and producer. We created a classification task for the RPTs where the only feature correlated with the class label was the degree of the objects in the relational data. Recall that movies with a positive class label tend to have higher degree with respect to actors and producers (there is no significant difference in director degree). On each actor, director, and producer object we added 10 random attributes (5 discrete and 5 continuous). Discrete attributes were drawn from a uniform distribution of ten values; continuous attribute values were drawn from a uniform distribution of integer values in the range [1,10]. The model considered 3 degree features, one for each type of object linked to the movie.

The second task (IMDB) also used the IMDb data described above, but used both the structure and the attributes in the original data. RPT models were built to predict movie success based on 14 attributes, such as movie genre and actor age. There were two continuous and two discrete attributes on each non-target entity type (actors, directors, and producers). Movies had two attributes (genre and year). The model also considered 3 degree features, one for each type of object linked to the movie.

The third task (CORA) used a subset of the Cora data described in Section 3 where the class label indicates whether a paper's topic is "neural networks." We selected the set of 1511 papers with at least one author, reference and journal. The RPT models had 12 attributes available for classification, including a cited paper's high-level topic (e.g. Artificial Intelligence) and an author's number of publications. There were equal proportions of discrete and continuous attributes on each non-target object.

For each of the three tasks, we built trees using three methods: the RPT algorithm with randomization tests (RTs), the RPT algorithm with only conventional significance tests (CTs), and the C4.5 algorithm. To examine the effect of degree disparity on the types of features selected, we recorded the number of nodes in the tree that used features based only on relational structure, which we called *degree features*, as well as recording the overall number of nodes. We weighted each count based on the proportion of training instances which travel through a given node. We also measured tree accuracy and area under the ROC curve (AUC). The experiments used two-tailed, paired t-tests to assess the significance of the results obtained from ten-fold cross-validation trials.

6.3 Results

As shown in Figure 8, CTs and RTs produced trees with equivalent performance with respect to accuracy and AUC across all data sets. C4.5's trees were significantly less accurate on RANDOM and equivalent on CORA. On IMDB, C4.5 trees were more accurate assuming equal misclassification costs (traditional accuracy), but less accurate when the entire area under the ROC curve is considered.

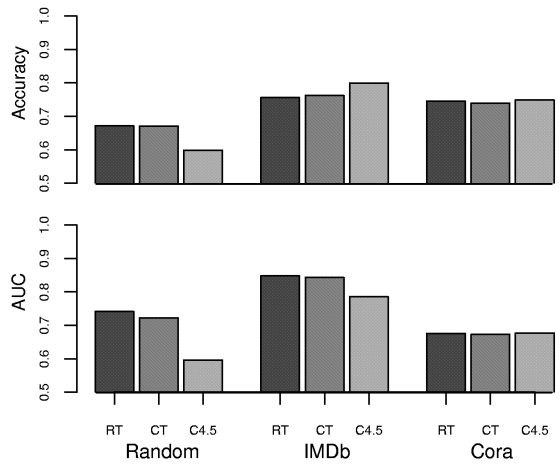

Figure 8. Tree accuracy and AUC.

Despite similar accuracy, trees built by the different methods have radically different structure. Figure 9 summarizes the features used in trees built with conventional tests and randomization tests. Each bar expresses both the size of the tree and the weighted proportion of degree features. In all data sets, RTs and C4.5 add much more non-degree structure than CTs.

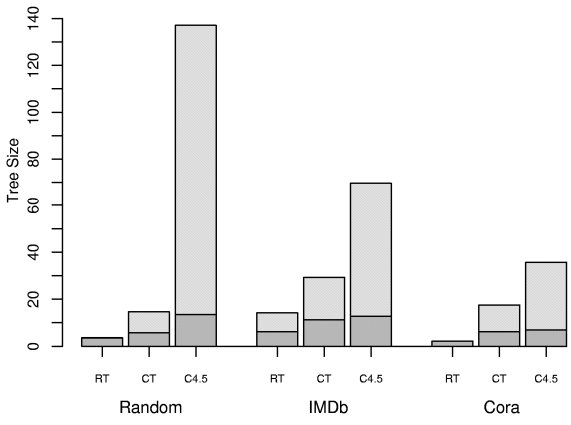

Figure 9. Tree size and weighted proportion of degree features.

The empirical results support our earlier conjectures. First, aggregation functions can cause misleading correlations in the presence of degree disparity. For example, in RANDOM, where only degree disparity of "actor" and "producer" objects are predictive, more than 60% of the

features selected by CTs, and more than 90% of the features selected by C4.5, were derived from random attributes that serve as surrogates for degree. Second, the trees from RANDOM show that aggregation functions can add complexity. Trees built with CTs and C4.5 were, on average, four times and 40 times larger, respectively, than trees built with randomization tests. Finally, randomization tests can adjust for the effects of degree disparity. In three different data sets, randomization tests result in trees with similar accuracy that are vastly smaller and contain a much larger proportion of degree features.

7. Conclusions and Future Work

Understanding the effects of degree disparity should affect the design of almost all approaches to relational learning, including algorithms for learning logic programs, probabilistic relational models, and structural logistic regression equations. However, to our knowledge, no learning algorithm for these models adjusts for the effects of degree disparity. This issue is not faced by other fields that consider autocorrelation (e.g., temporal or spatial analysis) because these fields generally consider problems with uniform degree.

Much interesting work remains to be done. First, we have largely ignored the issue of autocorrelation among attribute values (though we do adjust for autocorrelation among class labels). Autocorrelation among attribute values could have strong effects on hypothesis tests, and we intend to explore new approaches to randomization that can also adjust for attribute autocorrelation. Second, the effects of degree disparity highlight potential problems of inference in incompletely sampled relational data. We intend to explore how to improve the accuracy of learning through the use of metadata on sampling rates and potentially missing data.

Acknowledgements

Helpful comments and assistance were provided by Lisa Friedland, Matthew Rattigan, three anonymous reviewers, and our ICML area chair. This research is supported by DARPA and NSF under contract numbers F30602-01-2-0566 and EIA9983215, respectively. 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 DARPA, NSF, or the U.S. Government.

References

Blockeel, H., and L. De Raedt (1998). Top-down induction of first-order logical decision trees. *Artificial Intelligence* 101:285-297.

Edgington, E. (1980). *Randomization Tests*. New York: Marcel Dekker.

Emde, W. and D. Wettschereck (1996). Relational instance based learning. In *Proc. 13th Int. Conf. on Machine Learning*. Morgan Kaufmann. 122-130.

Getoor, L., N. Friedman, D. Koller, A. Pfeffer (2001). Learning probabilistic relational models. In *Relational Data Mining*, S. Dzeroski and N. Lavrac (Eds.). Springer-Verlag.

Jensen, D. and P. Cohen (2000). Multiple comparisons in induction algorithms. *Machine Learning* 38(3):309-338.

Jensen, D. and J. Neville (2002). Linkage and autocorrelation cause feature selection bias in relational learning. In *Proceedings of the 19th International Conference on Machine Learning*. Morgan Kaufmann. 259-266.

Lachiche, N. and P. Flach (2002). 1BC2: A true first-order Bayesian classifier. In *Proc. of the 12th Int. Conf. on Inductive Logic Programming*. Springer-Verlag.

Knobbe, A., A. Siebes, and B. Marseille (2002). Involving aggregate functions in multi-relational search. In *Proc. of the 6th European Conf. on Principles of Data Mining & Knowledge Discovery*. Springer-Verlag.

Kramer, S. (1996). Structural regression trees. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*. 812-819.

Krogel, M. and S. Wrobel (2001). Transformation-based learning using multirelational aggregation. In *Proc. of the 11th Int. Conf. on Inductive Logic Programming*. Springer-Verlag. 142-155.

McCallum, A., K. Nigam, J. Rennie, & K. Seymore (1999). A machine learning approach to building domain-specific search engines. In *Proceedings of the 16th Int. Joint Conf. on Artificial Intelligence*. 662-667.

Neville, J., D. Jensen, L. Friedland and M. Hay (2003). Learning relational probability trees. To appear in the *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining.*

Popescul, A., L. Ungar, S. Lawrence and D. Pennock (2002). Towards structural logistic regression: Combining relational and statistical learning. In *Proceedings of the SIGKDD 2002 Workshop on Multi-Relational Data Mining*. 130-141.

Provost, F. and P. Domingos. (2000). Well-trained PETs: Improving probability estimation trees. CDER Working Paper #00-04-IS, Stern School of Business, NYU.

Sachs, L. (1982). *Applied Statistics: A Handbook of Techniques*. Springer-Verlag.

Witten, I. and E. Frank (1999). *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, San Francisco.