

Avoiding Bias when Aggregating Relational Data with Degree Disparity

David Jensen
Jennifer Neville
Michael Hay

JENSEN@CS.UMASS.EDU
JNEVILLE@CS.UMASS.EDU
MHAY@CS.UMASS.EDU

Knowledge Discovery Laboratory, Dept. of Computer Science, University of Massachusetts, Amherst, MA 01003 USA

Abstract

A common characteristic of relational data sets leads many relational learning algorithms to discover misleading correlations. This characteristic—*degree disparity*—occurs when entities of one class participate in systematically higher numbers of relations than entities of another class. In such cases, the aggregation functions that are used in many relational learning algorithms (e.g., AVG, MODE, SUM, EXISTS, COUNT, MAX, MIN) 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 significance testing procedures can adjust for the effects of using aggregation functions in the presence of degree disparity.

1. Introduction

A wide variety of algorithms for relational learning have been developed in the past several years. Techniques have been devised for learning probabilistic relational models (Getoor, Friedman, Koller, & Pfeffer 2001), structural logistic regression equations (Popescul, Ungar, Lawrence, & Pennock 2002), relational rules (Knobbe, Siebes, & Marseille 2002), and relational probability trees (Neville, Jensen, Hay, & Friedland 2003). These techniques have been applied to learning tasks such as predicting mutagenicity from molecular structure, classifying web pages, and predicting the box office success of movies.

Despite these algorithmic advances and application successes, important questions remain about the unique challenges of relational learning. One central question is how to appropriately summarize the complex structure of relational data in ways that are useful to a learning algorithm. In contrast to the relatively simple structure of instances in propositional learning, instances in relational data do not necessarily have consistent structure. For example, different molecules have varying numbers of atoms and bonds, different web pages have varying numbers of incoming and outgoing links, and different movies have varying numbers of actors and producers. Some relational learning approaches standardize instances by

"flattening" their relational structure prior to learning (Krogl & Wrobel 2001), and other approaches (e.g., probabilistic relational models) summarize complex relational structure "on the fly" as they learn.

1.1 Feature evaluation

For example, 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.).

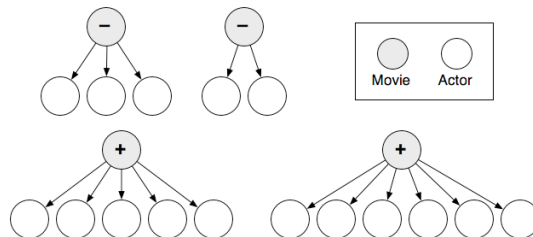


Figure 1: Example fragments of relational data about movies.

Consider the problem of determining 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 example, 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 of a particular actor attribute. MODE and EXISTS are often called *aggregation functions*, and such functions are common to many languages for handling relational data (e.g., SQL), although each language typically implements only a subset of all possible aggregation functions. In this case, the number of possible features, $|features| = kna$, 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). Thus, our example generates 200 binary features.

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 ($\alpha=0.05$, appropriately adjusted for 200 tests). Ten of the 100 MODE features appear useful.

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 IMDb, the *actor degree of movies*—the number of actors associated with any given movie—varies systematically with class label. Successful movies tend to have more actors than unsuccessful movies. Figure 2 shows this tendency graphically. Though subtle, the effect is highly significant ($p < 2.2e-16$), if we compare movies based on whether they gross more than \$2 million in their opening weekend. We call this systematic difference *degree disparity*.

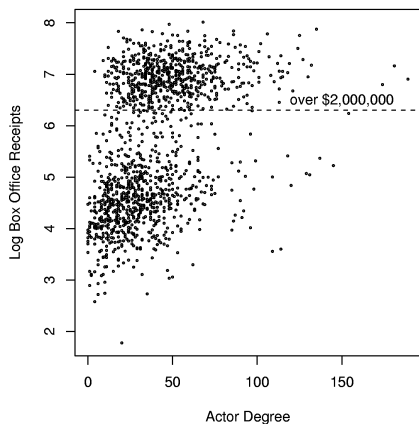


Figure 2: Actor degree varies with box office receipts

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. Similarly, the COUNT of a particular value of a discrete attribute (*gender=female*) will be systematically higher for movies with more actors. 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. All these effects, and the consequences for learned models, are discussed in more detail below.

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 ("observed"), 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.

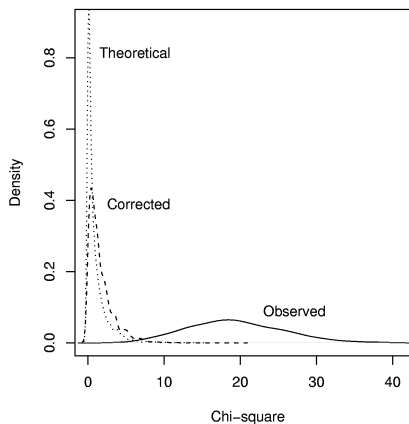


Figure 3: Theoretical, corrected, and observed distributions of the chi-square statistic given the actor degree disparity in the IMDb data.

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 biases 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. 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 preprocess data to "flatten" or "propositionalize" it prior to applying a conventional non-relational learning algorithm. This preprocessing often uses aggregation functions. For example, Kregel and Wrobel (2001) use AVG, 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 use MODE, AVG, MEDIAN, MAX, MIN, and SIZE. Probabilistic relational models (PRMs) learn a form of graphical model that represents the joint probability distribution over individual and aggregated values of particular attributes. Aggregation is central to this model. 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 makes heavy use of aggregation functions to create dichotomous divisions within an RPT. We use COUNT, PROPORTION, MIN, MAX, SUM, AVG, MODE, and EXISTS.

Other approaches to learning in relational data make heavy use of EXISTS. Some of these approaches are based on inductive logic programming (ILP). Others, however, are relational adaptations of known techniques for propositional learning. For example, Popescul, Ungar, Lawrence, and Pennock (2002) adapt logistic regression to the problem of relational learning, using EXISTS to create ILP-like features that serve as independent variables in the regression equation. Blockeel and De Raedt (1998), use EXISTS to create ILP-like 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 dominant technique in relational learning, some techniques use other approaches to handling the varying structure of relational instances. For example, Lachiche and Flach (2002) discuss the use of set-valued probability estimators. Rather than estimating conditional probability distributions (CPDs) for the aggregated values of some attribute, they discuss estimating CPDs over sets of values. An alternate approach is taken by Neville, Jensen, Gallagher, and Fairgrieve (2003) to learn relational Bayesian classifiers. They form CPDs for individual values and then combine probabilities under an assumption of independence. However, approaches that eschew 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 will define degree disparity as "systematic variation in the distribution of the degree for one entity type with respect to the class label on another entity type." 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 actually true.

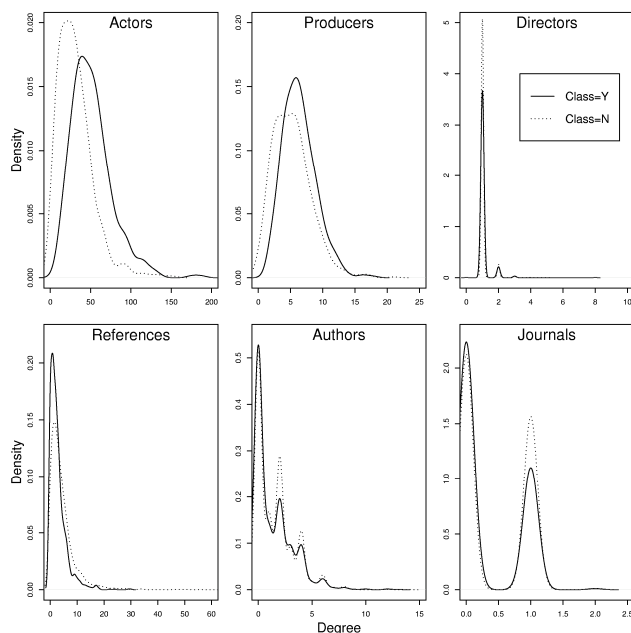


Figure 4: degree disparity in IMDb and Cora data sets

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 Internet Movie Database (www.imdb.com). We collected a sample of 1382 movies released in the United States between 1995 and 2000. In addition to movies, the data set contains objects representing actors, directors, producers. In total, this sample contains approximately 46,000 objects and 68,000 links. We discretized movie opening-weekend box office receipts so that a positive class label indicates a movie earned 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. The degree disparity for two types of entities—actors and producers—are statistically significant ($p<0.0001$); the degree disparity for the third type of entity—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 selected a set of 4330 machine-learning papers along with associated authors, cited papers, and journals. The resulting collection contains 11,500 objects and 26,000 links. Figure 4 (bottom) shows degree disparity for three types of entities in a collection of machine learning papers. The class label indicates whether a particular paper was assigned the topic of "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 has large effects, as we will show in later sections.

Degree disparity can reflect a wide variety of different influences. For example, the degree disparity observed in Cora may indicate that researchers in neural networks cite a smaller number of papers, or that the Cora database contains more conference papers in neural networks (as opposed to journal articles which cite a larger number of papers), or that the Cora system had greater difficulty correctly parsing citations in neural network papers. One can also imagine degree disparity that is far more extreme than these cases. For example, the "publication degree" of researchers with respect to their seniority will exhibit degree disparity merely because of the cumulative nature of a researcher's publication list. Similarly, the "incoming-link degree" of web pages with respect to their popularity will exhibit degree disparity due to the pages' popularity itself.

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 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 cumulative binomial distribution. That is, $|female\ actors| = b(s,t,p)$, where t is the total number of actors and p is the probability that a given actor will be female. The cumulative binomial distribution increases monotonically with increasing t . 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 t , where $t > \mu$, decreases as the sample size increases. Finally, The probability of achieving a particular MAX or MIN also varies with degree. In a different context, Jensen & Cohen (2000) show that MAX depends strongly on the number of values aggregated.

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, and in a later section with more extensive experiments, we present evidence for positive answers to both questions.

To illustrate the bias caused by degree disparity, we took the existing link structure of the IMDB data set and generated attributes whose values were uncorrelated with the class label. On a 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 a total of 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.

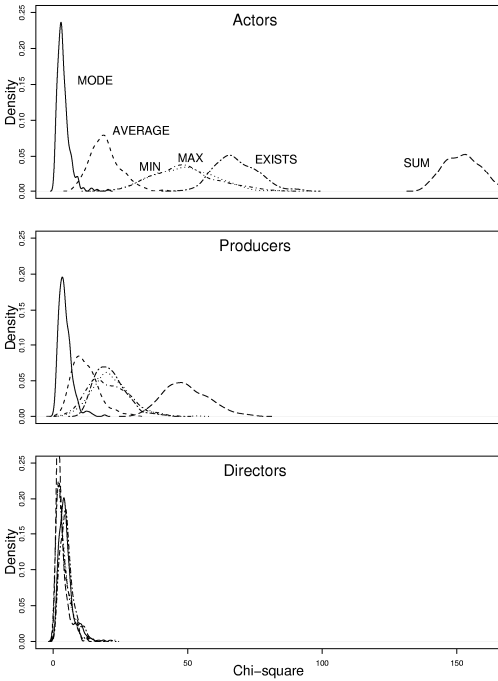


Figure 5: Simulation results for different types of attributes

To test the effect of degree disparity on feature selection, we compared features using both a conventional chi-square test and a randomization test (described in section 5). This experimental protocol allowed us to examine decisions made on all features, rather than only the top-ranked feature. Figure 6 shows ranked scores for all features. Dark shading indicates a significant association between the given feature and the class label ($p < 0.05$, adjusted for multiple comparisons), using a randomization test. In the absence of an adjusted test, features would be deemed significant if they had a score exceeding 9.5 for Cora and 10.8 for IMDB.

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 dis-

parity. The introduction contained one example of this type of test—a modification of a standard chi-square test.

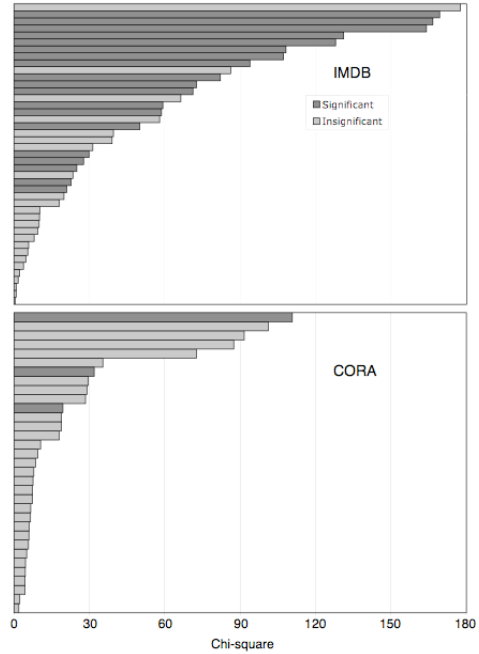


Figure 6: Histograms of ranked scores of features in two relational data sets. Bar length indicates the raw chi-square score. Shading indicates whether the feature is significant.

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.

	+	-
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 8: An example contingency table (a), expected cell counts (b), and expected cell counts with degree disparity (c).

For example, consider the contingency table shown in Figure 8a. The table summarizes the relationship between a class label and a feature value, where each class label is an element of $\{+, -\}$ and each feature value is an element of $\{T, F\}$. Based on figure 8a, we can calculate the expected values for each cell under the assumptions that: 1) the class label and feature value of each instance are independent; and 2) data instances are independent. Given the actual (figure 8a) and expected (figure 8b) counts, we can calculate the probability of the actual counts, under the null hypothesis of independence ($p = 0.003$).

As we showed in section 4, degree disparity can introduce a dependence between class labels and the feature values, thus violating the first of these assumptions. However, given a particular empirical distribution of degrees for each class, we can calculate the expected feature values, given only the dependence introduced by degree disparity. For example, consider the problem of calculating expected values for the feature `COUNT(gender=female)>2` for movie actors. The overall distribution of gender in our sample of movies is 66% male and 34% female. To calculate the table of expected values, we will assume that each attribute value (i.e., $\{M|F\}$) 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 $\{0.220,0.780\}$. The probability distributions for each instance provide fractional counts to the cells in a contingency table. By summing the fractional counts for all instances in a particular data set, we can obtain a table such as the one in Figure 8c. Here, we assume that movies with positive labels have 10 actors, but those with negative labels have only five actors. With these revised expected values, the probability of obtaining a table such as 8a, under the null hypothesis of attribute value independence, is much larger ($p = 0.813$). This was the method 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 that 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 (Jensen, Neville, & Rattigan 2003) provide an alternative method for hypothesis testing under the assumption of degree disparity. A randomization test is a type of computationally intensive statistical test (Edgington 1980). Other types include resampling and Monte Carlo procedures. Each of these tests involves generating many replicates of an actual data set — typically called pseudosamples — and using the pseudosamples to estimate a sampling distribution. In the case of a randomization test, pseudosamples are generated by randomly reordering (or permuting) the values of one or more variables in an actual data set. 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 is used to estimate a sampling distribution for the score calculated from the actual data. Randomization tests are also called *permutation tests*.

To construct pseudosamples in data with degree disparity, we randomize the attribute values prior to aggregation. Then, the values are aggregated and the association between the aggregated feature and the class label is assessed using a conventional chi-square calculation. Note that this calculation is made without the sorts of adjustments discussed in section 5.1 above. The chi-square statistic is calculated as if degree disparity does not introduce any correlation between the feature values and the class labels.

The empirical sampling distribution produced by the randomization procedure approximates the distribution that would be expected 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. It is typical for a randomization test to generate and evaluate 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 is included in some inner loop of a learning procedure. What countervailing benefit could 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 autocorrelation in relational data (Jensen, Neville, & Rattigan 2003). 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 attribute selection errors (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 to three relational data sets.

6.1 Model

Relational Probability Trees (RPTs) extend standard probability estimation trees (Provost & Domingos 2000)

to a relational setting. The RPT algorithm constructs a probability estimation tree to predict the target class label given (1) the attributes of the target objects, (2) the attributes of other objects and links in the relational neighborhood of the target objects, and (3) graph attributes specifying the structure of relations.

The RPT learning algorithm searches over a space of binary relational features. The algorithm considers the attributes of different 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 splits such as $AVG(\text{age}) > 25$ for numeric attributes such as actor-age, and splits such as $MODE(\text{gender}) = \text{Male}$ for nominal attributes such as actor-gender. The algorithm also searches over degree attributes that count the number of items in each relation (e.g., $DEGREE(\text{actor}) > 6$).

The RPT algorithm is a standard recursive partitioning algorithm which uses Bonferroni-adjusted chi-square tests of significance to select features. All the experiments reported in this paper used a Bonferroni-adjusted α value of 0.05 as the stopping criteria. For an extended description of the algorithm see (Neville et al. 2003).

6.2 Classification tasks

Our first task uses the IMDb data set described in section 3 where the class label indicates a movie earned more than \$2 million in opening-weekend receipts. 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 structure. 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 also used the IMDb dataset, 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 used the Cora data set described in section 3 where the class label indicates whether a paper’s topic is “neural networks.” In addition to papers, the data contained objects representing authors, cited papers and journals. The RPT models had 12 attributes available for classification, including attributes such as 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 RPT models that used conventional significance tests (CTs) to evaluate feature splits and RPT models that used randomization tests (RTs) to measure significance. As a measure of the effect of degree disparity, we recorded the number of non-degree features and the number of degree features selected as nodes in the tree. We weighted each feature depending on the proportion of training instances which travel through that node. We also measured tree accuracy and area under the ROC curve (AUC). The experiments used ten-fold cross-validation.

6.3 Results

Trees built with conventional tests and the randomization tests had equivalent performance with respect to accuracy and AUC. However, the tree had radically different structure. Figure 8 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 attributes, averaged over ten trials.

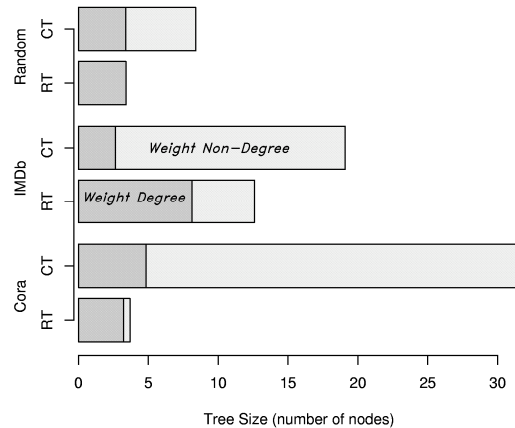


Figure 8: Tree size and weighted proportion of degree features.

The results support the claims made earlier. First, aggregation functions can cause misleading correlations in the presence of degree disparity. This claim is supported by the results from the Random data set, where the only predictive structure is the degree disparity of “actor” and “producer” objects. By our measure of weighted proportion, approximately 2/3 of the trees built with conventional tests (CT) consisted of features derived from random attributes that served as surrogates for degree. Second, the trees from the Random data set show that aggregation functions can add complexity. Trees built with conventional tests were, on average, twice the size of the 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 comparably accurate trees that are significantly smaller and contain a much larger proportion of degree features.

7. Conclusions and Future Work

Many current techniques for relational learning use aggregation functions. However, if those functions are applied without adjusting for the underlying structure of relational data, they can produce misleading models. 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 propositionalizing relational data, as well as other approaches that ignore the correlation between attribute values and relational structure.

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 consistent degree.

Much interesting work remains to be done. First, we have largely ignored the issue of autocorrelation among attribute values. This type of autocorrelation 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 and Matthew Rattigan. 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.

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 *Proc. of the 19th Int. Conf. on Machine Learning*. Morgan Kaufmann. 259-266.

Jensen, D., J. Neville and M. Rattigan (2003). Randomization tests for relational learning. Submitted to the *18th Int. Joint Conf. on Artificial Intelligence*.

Lachiche, N. and P. Flach (2002). IBC2: 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 *Proc. of the 13th National Conf. on Artificial Intelligence*.

Kroegel, 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 *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence*.

Neville, J., D. Jensen, L. Friedland and M. Hay (2003). Learning relational probability trees. Submitted to the *9th Int. Conf. on Knowledge Discovery & Data Mining*.

Neville, J., D. Jensen, B. Gallagher and R. Fairgrieve (2003). Simple estimators for relational Bayesian classifiers. Submitted to the *18th Int. Joint Conf. on Artificial Intelligence*.

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