D. Jensen and J. Neville (2002). Linkage and autocorrelation cause feature selection bias in relational learning. *Proceedings of the Nineteenth International Conference on Machine Learning (ICML2002)*. Morgan Kaufmann. pp. 259-266

Linkage and Autocorrelation Cause Feature Selection Bias in Relational Learning

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

Abstract

Two common characteristics of relational data sets — *concentrated linkage* and *relational autocorrelation* — can cause learning algorithms to be strongly biased toward certain features, irrespective of their predictive power. We identify these characteristics, define quantitative measures of their severity, and explain how they produce this bias. We show how linkage and autocorrelation affect a representative algorithm for feature selection by applying the algorithm to synthetic data and to data drawn from the Internet Movie Database.

1. Introduction

Recent efforts to learn statistical models from relational data include work on stochastic logic programming (Muggleton 2000), probabilistic relational models (Getoor et al. 1999), and relational Bayesian classifiers (Flach and Lachiche 1999). Relational data representations greatly expand the range and applicability of machine learning, but the greater expressive power of relational representations produces new statistical challenges. Work on relational learning often diverges sharply from traditional learning algorithms that assume data instances are statistically independent. Statistical independence of instances is among the most enduring and deeply buried assumptions of traditional machine learning methods, and it is contradicted by many relational data sets.

In this paper, we focus on how dependence among the values of a class label in relational data can complicate feature selection in methods for machine learning. We define *relational feature selection* and give a simple example of how such a procedure can be biased. We define quantitative measures of concentrated linkage (L) and relational autocorrelation (C'), two common characteristics of relational data sets. We show how high values of L and C' reduce the *effective sample size* of some data sets, introduce additional variance, and lead to feature selection bias. To our knowledge, no current relational learning algorithm accounts for this bias. We show how to estimate the variance of scores and discuss using those estimates to improve feature selection in relational data.

1.1 Relational Data and Statistical Dependence

Figure 1 presents two simple relational data sets. In each set, instances for learning consist of subgraphs containing a unique object x, an object y, and one or more other objects. Objects x contain a class label and objects y contain an attribute that will be used to predict the class label of x. Figure 1a shows a data set where objects x and y have a one-to-one relationship and where the class labels on instances are independent. Figure 1b shows instances where objects x and y have a many-to-one relationship and where the class labels are dependent.[1]

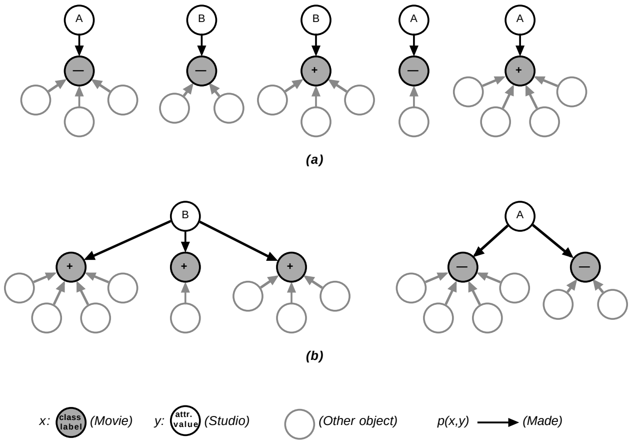

Figure 1: Example relational data sets with independent instances (a) and dependent instances (b).

We will spend the majority of the paper considering data sets similar in structure to Figure 1b where each subgraph consists of multiple relations and each relation may produce dependencies among the instances. For simplicity, our experiments assume that all relations are binary, placing this work somewhere between multiple instance learning and full first-order logic (DeRaedt 1998), although the statistical effects we investigate appear to affect a wider range of relational learning algorithms.

[1] Throughout this paper, we assume that all linkages among instances that could introduce statistical dependence are represented explicitly as links (edges) in the data graph.

1.2 Relational Feature Selection

This paper focuses on *feature selection*, a component of many learning algorithms. We define *feature* as a mapping between raw data and a low-level inference. For example, a feature for a propositional data set about medical patients might be *temperature > 99°F.* In this case, a feature combines an attribute (*temperature)*, an operator, and a value. Typically, many features are combined into a higher-level model such as a tree or rule set. We define *feature selection* as any process that chooses among features, either by identifying the best, selecting some and rejecting others, or merely placing a partial or total ordering over all possible features. This definition is broader than some (e.g., John, Kohavi, and Pfleger 1994), but it emphasizes the central role of feature selection in machine learning algorithms, including algorithms for learning decision trees, classification and association rules, and Bayesian nets. Feature selection is central to any learning algorithm that forms models containing a subset of all possible features.

We focus here on *relational* feature selection. Relational features are used by models that predict the value of an attribute on particular types of objects (e.g., the box office receipts of movies) based on attributes of related objects (e.g., characteristics of the movie's director, producer, actors, and studio). Relational features are similar to the features described above in that they identify both an attribute and a way of testing the values of the attribute. However, relational features may also identify a particular relation (e.g. *ActedIn(x,y))* that links a single object x (e.g. movie) to a set of other objects Y (e.g. actors). If this is the case, the attribute referenced by the feature may belong to the related objects Y (e.g. age), and the test is conducted on the set of attribute values on the objects in Y. For example, the relational feature:

$$Max(Age(Y)) > 65 \quad \text{where} \quad Movie(x), \quad Y = \{y \mid ActedIn(x,y)\}$$

determines whether the oldest of the actors in movie x is over 65. Throughout this paper, we will use $f(x)$ to refer the value of attribute f for a single object x, and $f(X)$ to refer to the set of values of attribute f for all $x \in X$.

A central characteristic of many relational data sets is that two or more objects of one type (e.g., movies) can both be connected to the same object of another type (e.g., a studio). We expect that this shared linkage represents some statistical associations present in the world. That is, linked objects are not statistically independent.

By examining relational feature selection in general, this work is relevant to nearly any learning algorithm that compares and selects among different relational features, including algorithms for inductive logic programming (Dzeroski & Lavrac 2001) and algorithms for constructing relational versions of commonly used model representations such as rules, trees, and Bayesian networks (Getoor et al. 1999).

1.3 An Example: Bias in Relational Feature Selection

Given that instances in relational data may not be independent, we should examine how such relational structure could affect feature selection. Below we show how relational structure and dependence among values of the class label can bias relational feature selection. To do this, we created data sets about movies and analyzed them with a simple algorithm for relational feature selection. Specifically, we created and analyzed a family of relational data sets whose relational structure was drawn from the Internet Movie Database (www.imdb.com). We gathered a sample of 1383 movies released in the United States between 1995 and 2000. In addition to movies, the data set contained objects representing actors, directors, producers, and studios. The data schema is shown in Figure 2.

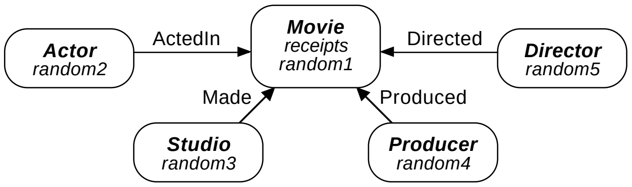

Figure 2: Schema for the movie data sets. Object and link types are shown in roman; attributes are shown in italics.

We created a learning task using a single attribute on movies — opening-weekend box office receipts. We discretized that attribute so that a positive value indicates a movie with more than $2 million in opening weekend receipts (*prob(+)=0.55*). We call this discretized attribute *receipts* and use it as a binary class label. For each of 1000 trials, we also generated a random binary attribute on each of the five types of objects (movies, studios, actors, directors, and producers).

In each trial, we applied an algorithm for relational feature selection, using features formed from the random attributes. The algorithm uses the following relational feature for each attribute $f(x)$ and each attribute value a_f:

$$\begin{gathered} Mode(f(Y)) = a_f \\ \text{where} \quad Movie(x), \quad Y = \{y \mid LinkedTo(x,y) \wedge Type(y) = t\} \end{gathered}$$

which determines whether the modal value of f on the objects of type t linked to x is equal to a_f.

The correlation of each relational feature with the class label was calculated using chi-square and the features were ranked by their chi-square values. In each trial, we identified the object type of the top-ranked feature.

Given that all attributes were created randomly, we would expect an algorithm to select all features with equal probability, since no attribute is useful for predicting *receipts*. However, as shown in the first column of Table 1, features formed from studio objects have a much higher probability of selection than features formed from movies, actors, directors, or producers.

Table 1: Probability of feature selection

Object Type	*Receipts* Class Label	Random Class Label
Studio	0.742	0.214
Director	0.059	0.218
Producer	0.073	0.174
Actor	0.072	0.186
Movie	0.054	0.208

This effect is eliminated if the values of *receipts* are assigned randomly (with the same probability as before) instead of using the actual values of *receipts*. These results are shown in the second column of Table 1. For a random class label, the algorithm has no bias toward studio objects. It behaves in the way we would expect, selecting among features formed from different object types with roughly equal probability. This raises obvious and intriguing questions: Why does the algorithm prefer random features formed from studios, and what does this tell us about relational feature selection in general?

2. Linkage and Autocorrelation

Our analysis indicates that bias such as that shown in Table 1 occurs when algorithms ignore two common characteristics of relational data — *concentrated linkage* and *relational autocorrelation*. We define these characteristics formally below. Informally, concentrated linkage occurs when many objects are linked to a common neighbor, and relational autocorrelation occurs when the values of a given attribute are highly uniform among objects that share a common neighbor. The example in Figure 1b shows several movies linked to an individual studio and shows that movies made by the same studio have highly correlated class labels.

2.1 Concentrated Linkage

We will define concentrated linkage $L(X,P,Y)$ with respect to two sets of objects X and Y and a set of paths P such that the relation $p(x,y)$ holds. Paths are composed of k links and $k-1$ intervening objects, where $k \geq 1$. Each path represents a series of relations linking an object in X to an object in Y. For example consider the path from linking two movies, m_1 and m_2, made by the same studio. The path is formed from two *Made* links, $Made(m_1,s_1)$ and $Made(m_2,s_1)$. For convenience we treat all links as undirected in order to refer to meaningful sequences of relationships as paths. We assume that paths in P are unique with respect to a given (x,y) pair; if two or more paths between x and y exist in the data, they are collapsed to a single element of P.

Definition: D_{yX} is the degree of the object y with respect to a set of objects X. That is, the number of $x \in X$ such that $p(x,y) \in P$. For example, D_{yX} might measure, for a given studio y, the number of movies (X) it has made. ■

Definition: *Single linkage* of X with respect to Y occurs in a data set whenever, for all $x \in X$ and $y \in Y$:

$$D_{xY} = 1 \quad and \quad D_{yX} \geq 1$$ ■

In these cases, many objects in X (e.g., movies) connect to a single object in Y (e.g., a studio). We use single linkage as an important special case in future discussions.

Definition: The *concentrated linkage $L(x,X,P,Y)$ of an individual object x* (e.g., a movie) that is linked to objects Y (studios) via paths P is:

$$L(x,X,P,Y) = \sum_{\substack{y \ s.t. \\ p(x,y) \in P}} \frac{(D_{yX} - 1)}{D_{yX}} \Big/ {D_{xY}}^2$$ ■

the quantity $(D_{yX} - 1)/D_{yX}$ within the summation is zero when the D_{yX} is one, and asymptotically approaches one as degree grows, and thus is a reasonable indicator of $L(x,X,P,Y)$, given single linkage of x with respect to Y. Because x may be linked to multiple nodes in Y, we define the average across all nodes y_i linked to x, and divide by an additional factor of D_{xY} to rate single linkage more highly than multiple linkage.

Definition: The *concentrated linkage $L(X,P,Y)$ of a set of objects X* (e.g., all movies) that are linked to objects Y is:

$$L(X,P,Y) = \sum_{x \in X} \frac{L(x,X,P,Y)}{|X|}$$ ■

Given particular types of linkage, L can be calculated analytically from the sufficient statistics $|X|$ and $|Y|$. For example, in the case of single linkage of X with respect to Y, $L = (|X|-|Y|)/|X|$. For example, the data set shown in Figure 1b exhibits single linkage, so $L(X,P,Y) = 0.60$. Propositional data also display single linkage, and because $|X|=|Y|$, $L(X,P,Y) = 0$. Calculations of several types of linkage are shown for the movie data in Table 2.

Table 2: Linkage in the movie data

Linkage Type	Value
L(Movie, Made, Studio)	0.91
L(Movie, Directed, Director)	0.23
L(Movie, Produced, Producer)	0.08
L(Movie, ActedIn, Actor)	0.01

In addition to the movie data, we have encountered many other instances of concentrated linkage. For example, while studying relationships among publicly traded companies in the banking and chemical industries, we found that nearly every company in both industries uses one of only seven different accounting firms. In work on fraud in mobile phone networks, we found that 800 numbers, 900 numbers, and some public numbers (e.g., 911) produced concentrated linkage among phones. Concentrated linkage is also common in other widely accessible relational data sets. For example, many articles in the scientific literature are published in a single journal and many basic research articles are cited in single review articles. On the Web, many content pages are linked to single directory pages on sites such as Yahoo.

2.2 Correlation and Autocorrelation

We will define relational correlation *C(X,f,P,Y,g)* with respect to two sets of objects *X* and *Y*, two attributes *f* and *g* on objects in *X* and *Y*, respectively, and a set of paths *P* that connect objects in *X* and *Y*.

Definition: *Relational correlation C(X,f,P,Y,g)* is the correlation between all pairs *(f(x),g(y))* where $x \in X$, $y \in Y$ and $p(x,y) \in P$. ■

Given the pairs of values that these elements define, traditional measures such as information gain, chi-square, and Pearson's contingency coefficient can be used to assess the correlation between values of the attributes *f* and *g* on objects connected by paths in *P*. The range of *C* depends on the measure of correlation used.

We can use the definition of relational correlation *C(X,f,P,Y,g)* to define relational *autocorrelation* as the correlation between the same attribute on distinct objects belonging to the same set.

Definition: *Relational autocorrelation C' is:*

$$C'(X,f,P) \equiv C(X,f,P,X,f) \quad \text{where} \quad \forall p(x_i,x_j) \in P \quad x_i \neq x_j \; ■$$

For example, *C'* could be defined with respect to movie objects, the attribute *receipts* on movies, and paths formed by traversing *Made* links that connect the movies to an intervening studio.

If the underlying measure of correlation varies between zero and one, then *C'=1* indicates that the value of the attribute for a specific node x_i is always equal to all other nodes x_j reachable by a path in *P*. When *C'=0*, values of *f(X)* are independent. Table 3 gives estimates of relational autocorrelation for movie receipts, linked through studios, directors, producers, and actors. For a measure of correlation, Table 3 uses Pearson's corrected contingency coefficient (Sachs 1992), a measure that produces an easily interpreted value between zero and one. Autocorrelation is fairly strong for all object types except actors.

In addition to the movie data, we have encountered many other examples of high relational autocorrelation. For example, in our study of publicly traded companies, we found that when persons served as officers or directors of multiple companies, the companies were often in the same industry. Similarly, in biological data on protein interactions we analyzed for the 2001 ACM SIGKDD Cup Competition, the proteins located in the same place in a cell (e.g., mitochondria or cell wall) had highly autocorrelated functions (e.g., transcription or cell growth). Such autocorrelation has been identified in other domains as well. For example, fraud in mobile phone networks has been found to be highly autocorrelated (Cortes et al. 2001). The topics of authoritative web pages are highly autocorrelated when linked through directory pages that serve as "hubs" (Kleinberg 1999). Similarly, the topics of articles in the scientific literature are often highly autocorrelated when linked through review articles.

Table 3: Autocorrelation in the movie data

Autocorrelation Type	Value
C'(Movie,Receipts,Made|Studio|Made)	0.47
C'(Movie,Receipts,Directed|Director|Directed)	0.65
C'(Movie,Receipts,Produced|Producer|Produced)	0.41
C'(Movie,Receipts,ActedIn|Actor|ActedIn)	0.17

Note: We use a the notation *a*|*x*|*b* to denote paths with links of type *a* and *b* and intervening objects of type *x*.

We have defined relational autocorrelation in a similar way to existing definitions of temporal and spatial autocorrelation (see, for example, Cressie 1993). Autocorrelation in these specialized types of relational data has long been recognized as a source of increased variance. However, the more general types of relational data commonly analyzed by relational learning algorithms pose even more severe challenges because the amount of linkage can be far higher than in temporal or spatial data and because that linkage can vary dramatically among objects.

Relational autocorrelation represents an extremely important type of knowledge about relational data, one that is just beginning to be explored and exploited for learning statistical models of relational data (Neville and Jensen 2000; Slattery and Mitchell 2000). Deterministic models representing the extreme form of relational autocorrelation have been learned for years by ILP systems. By representing and using relational autocorrelation, statistical models can make use of both partially labeled data sets and high-confidence inferences about the class labels of some nodes to increase the confidence with which inferences can be made about nearby nodes.

However, as we show below, relational autocorrelation can also greatly complicate learning of all types of relational models. As we seek to represent and use relational autocorrelation in statistical models of relational data, we will need to adjust for its effects when evaluating more traditional types of features in these models.

2.3 Discussion

The results reported so far for concentrated linkage and relational autocorrelation provide important clues to the behavior reported in Table 1. Figure 3 plots all objects in the movie data in terms of their linkage and autocorrelation with respect to movies, as reported in Tables 2 and 3. The contours in the plot delineate regions where the severity of the bias introduced by linkage and autocorrelation is approximately equal. The contours are a 2-D view of the results reported in Figure 4 (described in detail in section 3.1). Studios objects in the movie data have the highest combination of concentrated linkage and relational autocorrelation. Features that use studios also show the greatest bias in the experiments reported in Table 1. While directors have a higher value of autocorrelation *C′*, their linkage *L* is quite low. As we will show in the next section, when linkage and autocorrelation are both high for a single type of object, they bias learning algorithms toward features formed from objects of that type.

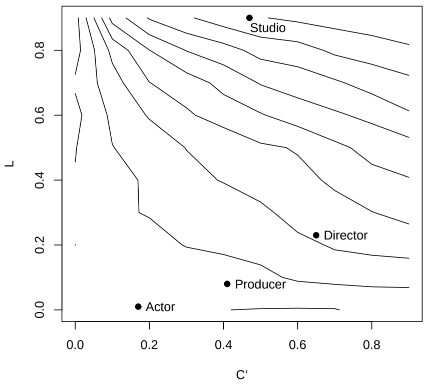

Figure 3: Relational autocorrelation vs. concentrated linkage

3. Effects of Linkage and Autocorrelation

Linkage and autocorrelation cause feature selection bias in a two-step chain of causality. First, linkage and autocorrelation combine to reduce the *effective sample size* of a data set, thus increasing the variance of scores estimated using that set. Relational data sets with high linkage and autocorrelation contain less information than an equivalently sized set of independent data. This reduction in effective sample size increases the variance of parameter estimates made with the data. Just as small data samples can lead to inaccurate estimates of the scores used to select features, concentrated linkage and autocorrelation can cause the scores of some features to have high variance. Second, increased variance of score distributions increases the probability that features formed from objects with high linkage and autocorrelation will be selected as the best feature, even when these features are random.

3.1 Decreased Effective Sample Size

Below, we prove a special case of linkage and autocorrelation decreasing effective sample size, for data exhibiting single linkage and in which $C'=1$ and $L\geq 0$. Then we explore a wider array of values for C' and L via simulation. Specifically, in data sets exhibiting single linkage, and where $L\geq 0$ and $C'=1$, the variance of scores estimated from relational features depends on $|Y|$ (the number of objects with an attribute value) rather than $|X|$ (the number of objects with a class label). For example, in the movie data, if autocorrelation of movie *receipts* through studios were perfect ($C'=1$), then the variance of scores for predicting *receipts* with a relational feature formed from studios (e.g., location) would depend on the number of studios in the sample rather than the number of movies.

Theorem: Given a relational data sample with objects X, objects Y, paths P, a class label $f(x)$, and an attribute $g(y)$, where $C'=1$, $D_{xY}=1$, and $D_{yX}\geq 1$, the sampling distribution for the scoring function $S(f,g)$ will have the same variance as it would for a data set with $|Y|$ independent instances.

Proof sketch: Consider the set of independent instances shown in Figure 1a, where $L=0$ (and, thus, $|X| = |Y|$). If we alter the data set so that $L>0$ (and, thus, $|Y|<|X|$), but retain the constraints that $C'=1$, $D_{xY}=1$, and $D_{yX}\geq 1$, then additional objects X will be added, and their corresponding values of $f(x)$ will match the value of other objects X already connected to a given Y. Such alterations increase the number of objects $|X|$, but they do not alter the number of possible arrangements of values of f and g. That number remains the same, because the value of $f(x)$ for additional objects X linked to a given Y is completely determined by the existing value of $f(x)$, given that $C'=1$. Each sample for which $L>0$, $C'=1$, and $D_{xY}=1$ corresponds directly to a sample for which $L=0$ though the latter sample contains fewer objects X. The number of ways of assigning values of f and g to objects is identical, the probability of each of these corresponding data sets remains equal, and the sampling distribution of any scoring function will also be identical. ■

In the independent case, the effective sample size $N = |X| = |Y|$. In the case where $L>0$, the effective sample size $N = |Y|<|X|$. In less extreme cases, where $0<C'<1$, the effective sample size lies somewhere between $|X|$ and $|Y|$. In addition, forms of linkage where $D_{xY}>1$ complicate estimates in ways we do not consider formally in this paper, although our experimental results below give some indications of the effect.

Simulation can demonstrate the effect of varying values of C' and L. We generated data sets with 1000 objects X with varying degrees of concentrated linkage to, and relational autocorrelation through, another set of objects Y ($|Y|$ varies between 1000 (L=0) and 100 (L=0.9)). We generated a binary class label on X and a binary attribute on Y, both drawn from a uniform distribution. At each level of linkage and autocorrelation, we generated 10,000 sets, calculated the chi-square score between the attribute and class label for each set, and then estimated the 95% confidence threshold for the resulting distribution. Because chi-square increases proportionally with sample size for a given level of association, we can find the effective sample size by dividing the 95% critical value of chi-square for independent data (3.84) by the 95th percentile of the distribution of simulated scores and multiply by the sample size (1000). The results are summarized in Figure 4.

In Figure 4, effective sample size drops monotonically with increases in C' and L. At extreme values of linkage and autocorrelation, effective sample size is reduced by almost an order of magnitude.

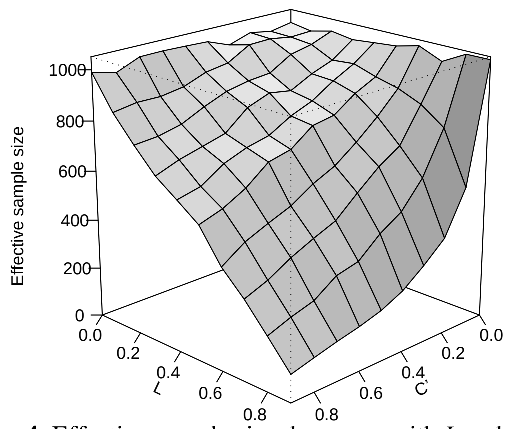

Figure 4: Effective sample size decreases with *L* and *C*'

We used similar means to gauge the effects of linkage and autocorrelation on the movie data, and estimated effective sample sizes for each object type. We examined the distribution of 200 random relational features formed using each object type. Autocorrelation and linkage should have no effect on movies (because movies are not linked directly to other movies), and the score distributions for features formed from movies approximately match the distribution of chi-square expected under the assumption of independent instances. The other distributions, however, differ substantially from this expectation. Table 4 shows the effective sample sizes obtained by minimizing the sum of the absolute difference at all percentiles of the empirical and theoretical chi-square distributions (other measures of similarity produced similar results).[2] In all cases, the assumed sample size would be equal to, or larger than, the number of movies (1383).

Table 4: Effective sample size in the movie data

Object Type	Scaling Factor	Effective Sample Size
Studio	0.026	36
Director	0.842	1164
Producer	0.615	851
Actor	0.702	971
Movie	1.000	1383

Figure 5 shows the distributions of scores obtained from testing the relational features used to construct Table 1. The dotted line shows the score distribution for studios. The solid lines show the overlapping distributions for movies, actors, directors, and producers. These latter distributions have quite similar variance, but the variance for features on studios is much higher. For these experiments, we used a chi-square statistic augmented with a sign, to indicate which diagonal of the contingency table contained the most mass (see Sachs 1992). In this way, we obtained a symmetric scoring function and we were able to tell if the utility of a given feature changed sign between data sets.

Evidence in Table 1, Figure 3, Table 4 and Figure 5, all points to common conclusions. Studios have the highest combination of linkage and autocorrelation, and the distributions of scores for features formed from studios display the highest variance. This variance reduces the effective sample size, and causes feature selection algorithms to be biased in favor of these features.

3.2 Feature Selection Bias

Given that the scores of some features are estimated with higher variance than others, why should this lead to a bias in favor of these attributes? Recent work on the statistical effects of multiple comparison procedures on score distributions (Jensen & Cohen 2000) provides an explanation.

Features are usually formed by a local search over possible parameters of the feature. For example, forming the feature mentioned earlier — *Max(Age(Y)) > 65* — could involve local search over many possible aggregation functions (*Max, Min, Average*), operators (*>, <, =*), and values (*[0,100]*). This local search is usually done prior to feature selection, so only the best feature from each feature "family" is compared.

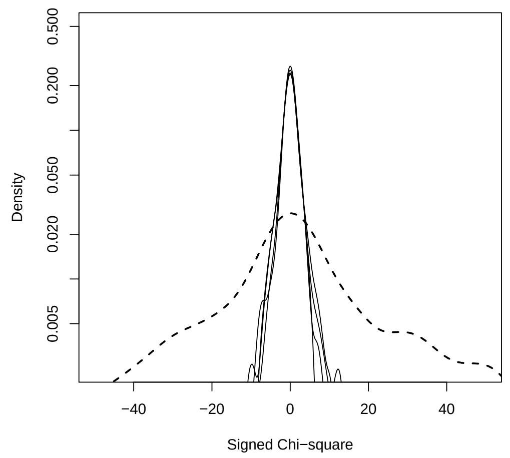

Figure 5: Distributions of random scores

Jensen and Cohen (2000) prove that, if the score of each member of a feature family is estimated with some variance, then the estimated score of the best member (the maximum score) will be a biased estimator of the feature's true score. In addition, that bias increases as the variance of the score distributions increases. Thus, the estimated score of features formed from objects with high linkage and autocorrelation (e.g., studios) will be more biased than those formed from objects with low linkage and autocorrelation (e.g., actors).

[2] The effective sample size estimated for studios is almost certainly too small, given that it is less than the total number of studios in the sample (128). The sampling distribution obtained for random attributes on studios had more density in the tails than the theoretical distribution for chi-square, and thus our similarity measures may not be adequate. We are exploring alternative methods for estimating effective sample size.

Results from the movie data clearly indicate high variance in estimated scores. Figure 7 shows score distributions based on multiple training/test splits of the movie data, where one set was used to select the best feature from each feature family, and the other set was used to obtain an unbiased score estimate. The scores vary widely, but features formed from studios have the highest variance. This experiment also indicates the competing pressures on feature selection algorithms in the face of high variance. Some random features on studios have variance sufficiently high to allow them to exceed the scores of weakly useful features on other objects. However, some non-random attributes on studios appear to form useful features, and any method for discounting high-variance features should not discard these.

4. Estimating Score Variance by Resampling

The first step toward correcting for high variance is to obtain accurate estimates of variance for each feature. In this section, we describe and test an approach to estimating score variance by bootstrap resampling.

4.1 Bootstrap Resampling

Bootstrap resampling is a technique for estimating characteristics of the sampling distribution of a given parameter by generating multiple samples by drawing, with replacement, from the original data as if it were the population (Noreen 1989). Each generated sample is called a *pseudosample* and contains as many instances as the original data set. Some instances in the original data set will occur multiple times in a given pseudosample, and others will not occur at all. Resampling can be used to estimate the variance of a parameter by estimating the parameter on hundreds of pseudosamples, and then finding the variance of the resulting distribution of scores.

To estimate the variance of a given score distribution using resampling, we draw links randomly and with replacement from all links in P until the number of links in the pseudosample is equal to the number in the original data. For example, to estimate variance for a relational attribute formed from studios, we would sample paths formed from *Made* links. Then we create objects based on the endpoints of the paths in the pseudosample. For example, we would create movie and studio objects based on the movies and studios that form the endpoints of the *Made* links in our pseudosample.

In most cases, we create a single object in response to many paths in the pseudosample with the same endpoint. For example, we would generally link many movies to a single studio object we have created for the pseudosample. In some cases, however, the degree of the resulting object in the pseudosample exceeds the degree of any similar object in the original data. In this case, we create an additional version of that object to keep linkage similar between the original sample and the pseudosamples. For example, movies in our data have single linkage to studios, thus we create extra movies in pseudosamples when the same link between movies and studios is sampled twice. The distribution of the scores calculated from many pseudosamples forms a distribution from which the variance can be estimated.

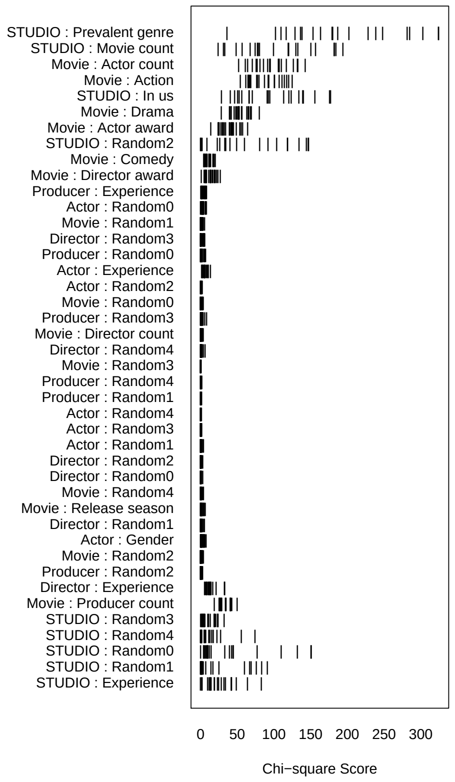

Figure 7: Scores of movie features and variance estimates

We evaluated the quality of pseudosamples by comparing their linkage, autocorrelation, and attribute distributions to the original sample. The measured quantities remain stable over all the pseudosamples and closely resemble the values in the original sample.

4.2 Using Resampled Estimates

While resampling can be used to estimate the variance of scores for particular features, the use of those estimates to improve feature selection remains an open problem. Figure 6 demonstrates the broad outlines of the problem. Given a set of scores for features and estimates of their sampling distributions, which features should be selected? In Figure 6, score B is clearly preferable to A, because it has both higher expected value and lower variance. However, scores B and C are not easily ranked because C has a higher expected value but also a higher variance.

We have tried two obvious ranking schemes without success. In the first, we ranked features based on their lower confidence limits (e.g., 5%). In the second, we grouped feature distributions into equivalence classes based on estimates of *prob(A>B)* for pairs of distributions. Features within equivalence classes were ranked based on variance. We evaluated these ranking schemes on randomly drawn subsets of movies and compared their rankings to the ranking on the full data set. We also conducted extensive simulations. Neither scheme significantly improved feature rankings. This issue of comparing distributions with unequal variance is a longstanding problem in statistics, and we are continuing to explore alternatives for improving feature selection.

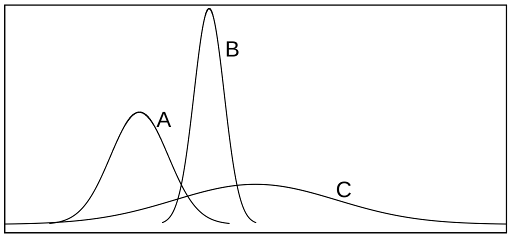

Figure 6: Example score distributions

Conclusions

Based on our work to date, substantial bias is likely to afflict relational learning algorithms that engage in feature selection in data sets with high linkage and autocorrelation. Some learning tasks avoid these conditions, either because the data consist of disconnected subgraphs (e.g., molecules) or because the data otherwise lack high linkage or autocorrelation. However, we have discovered high linkage and autocorrelation in data sets drawn from many domains, including web page analysis, fraud detection, and citation analysis. General-purpose algorithms for relational learning will need to address this source of bias.

We attribute the poor performance of our alternative ranking schemes to the fact that, given only the data, the individual score of a given feature remains the best estimate of its utility. Bootstrap resampling appears to provides accurate approximations to score distributions, but those distributions only indicate that the true score could be substantially lower or higher than the estimated score, with roughly equal probability. We suspect that future research to avoid feature selection bias will have to consider additional information, such as prior estimates of the true score. Given such information, priors on the true scores would receive different updates for the same set of data, because of the differing effective sample sizes for each feature. This may provide a better route to using the distribution information than our experiments to date.

Regardless of the shape of the eventual solution, the bias associated with linkage and autocorrelation indicates the importance of maintaining relational data representations, rather than propositionalizing data. Maintaining a relational data representation makes it possible to assess the statistical effects of linkage and autocorrelation, and to adjust for the resulting bias. In addition, as noted in section 2.2, maintaining relational representations allows inference procedures to exploit relational autocorrelation to improve the predictive accuracy of models.

Acknowledgments

Foster Provost and Hannah Blau provided valuable comments. Ross Fairgrieve prepared the movie data for analysis. The data used in this paper were drawn from the Internet Movie Database (www.imdb.com). This research is supported by The Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL), Air Force Material Command, USAF, under agreements F30602-00-2-0597 and F30602-01-2-0566.

References

Cortes, C., D. Pregibon, and C. Volinsky (2001). Communities of Interest. *Proceedings of the Fourth International Symposium on Intelligent Data Analysis*.

Cressie, N. (1993). *Statistics for Spatial Data*. Wiley.

Dzeroski, S. and N. Lavrac, (Eds.) (2001). *Relational Data Mining*. Berlin: Springer.

De Raedt, L. (1998). Attribute-Value Learning versus Inductive Logic Programming: The Missing Links. *ILP '98*. 1-8. Springer

Flach, P. and N. Lachiche (1999). 1BC: a first-order Bayesian classifier. *ILP'99*. 92-103. Springer.

Getoor, L., N. Friedman, D. Koller, and A. Pfeffer (1999). Learning probabilistic relational models. *IJCAI'99*. 1300-1309.

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

John, G., R. Kohavi, and K. Pfleger (1994). Irrelevant features and the subset selection problem. *ICML'94*. 121-129.

Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM* 46:604-632,

Muggleton, S. (2000). Learning Stochastic Logic Programs. *AAAI Workshop on Learning Statistical Models from Relational Data*, 36-41.

Noreen, E. (1989). *Computer Intensive Methods for Testing Hypotheses*. Wiley.

Neville, J. and D. Jensen (2000). Iterative Classification in Relational Data. *AAAI Workshop on Learning Statistical Models from Relational Data*, 42-49.

Sachs, L. (1982). *Applied Statistics.* Springer-Verlag.

Slattery, S., and Mitchell, T. (2000). Discovering test set regularities in relational domains. *ICML 2000. 895-902.*