Randomization Tests for Relational Learning

David Jensen, Jennifer Neville and Matthew Rattigan

Knowledge Discovery Laboratory, Department of Computer Science, University of Massachusetts,
140 Governors Drive, Amherst, MA 01003 USA
{jensen | jneville | rattigan} @cs.umass.edu

Abstract

Algorithms for relational learning and propositional learning face different statistical challenges. In contrast to propositional learners, relational learners often make statistical inferences about data that exhibit *linkage* and *autocorrelation*. Recent work has shown that these characteristics of relational data can bias inferences made by relational learners. In this paper, we develop a novel variant of a known statistical procedure — a randomization test — that produces accurate hypothesis tests for relational data. We show that our procedure produces unbiased inferences in situations where more obvious adaptations of existing randomization tests fail.

1 Introduction

Many algorithms for machine learning attempt to distinguish pattern from noise. For example, algorithms for constructing classification trees prune away structure inferred to be useless, algorithms for constructing graphical models remove edges between variables inferred to be conditionally independent, and algorithms for stepwise logistic regression remove variables inferred to produce no significant increase in accuracy. Some learning algorithms, such as simple Bayesian classifiers and neural networks, use all available features, although selective versions of these algorithms have also been developed. In general, models produced by algorithms that select relevant features are preferable for applications where data collection has high costs, where users wish to understand the learned models, or where model runtime must be minimized.

2 Background

2.1 Hypothesis Tests

One widely used method for distinguishing pattern from noise is a statistical hypothesis test. Hypothesis tests compare the value of a statistic (e.g., the correlation of a given feature with the class label) to a *sampling distribu-tion*. A sampling distribution represents the values of the statistic that would be expected under a given *null hypothesis*. A typical null hypothesis is that a feature and a class label are statistically independent, though other null hypotheses are also common. If the value of the statistic exceeds a large percentage of the values in the sampling distribution, the null hypothesis is rejected, and some alternative hypothesis (e.g., that the given feature is correlated with the class label) is accepted. Hypothesis tests have been employed in learning algorithms to limit the size of classification trees [Frank & Witten 1998], select association rules [Megiddo & Srikant 1998], and construct Bayesian networks [Spirtes, Glymour, and Scheines 1993].

For example, consider an extremely simple scenario: a learning algorithm is presented with a small data set consisting of 100 instances. For each instance, we know the value of a binary class label C and a single binary attribute A. The algorithm must determine whether A is independent of C. Inferring non-independence might cause a tree-building algorithm to form a three-node classification tree or a graphical modeling algorithm to create a two-node connected graph rather than a graph with two disconnected nodes.
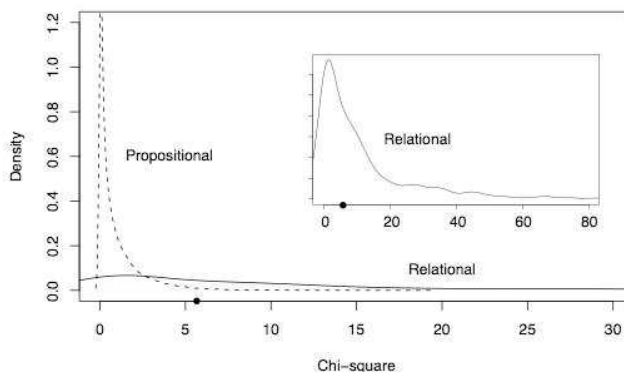

Figure 1: The value of a statistic compared to sampling distributions appropriate for propositional and relational data.

Widely known and computationally simple procedures exist for hypothesis tests in cases such as this. One classical hypothesis test would score the bivariate association

between A and C using a chi-square statistic (producing a value, e.g., 5.91). Exact calculations and approximations of the sampling distribution for chi-square are widely known, and they are parameterized by the number of values in each variable. Such a distribution is shown in Figure 1 (labeled *Propositional*). Based on this distribution, the value of 5.91 appears to exceed the vast majority of the values in the sampling distribution, allowing a learning algorithm to reject the null hypothesis with high confidence. In this case, an algorithm would typically include the variable in an induced model.

2.2 Relational Data

However, the propositional distribution in Figure 1 is derived under the assumption that the 100 data instances are independent and identically distributed. If the instances are drawn from a *relational* data set, the data instances may not be independent. For example, consider the two data sets shown schematically in Figure 2. The data in Figure 2a consists of pairs of objects $<U,W>$, where the pairs of objects in one instance are distinct from the objects in every other instance. For example, instances might be formed of users (U) and their workstations (W). Such a data set can be represented propositionally and the structure of the data produces no obvious dependence among instances. In contrast, the data in Figure 2b require a relational representation because they consist of pairs of objects where some objects are shared (e.g., users (U) and workgroup servers (W)). Relational data sets with this latter structure are said to have high *linkage*. In addition, many relational data sets exhibit high consistency among the class labels of pairs of objects U connected through objects W. For example, the users of a single workgroup server may all have the same research area. Such consistency in the value of a variable is referred to as *autocorrelation*.
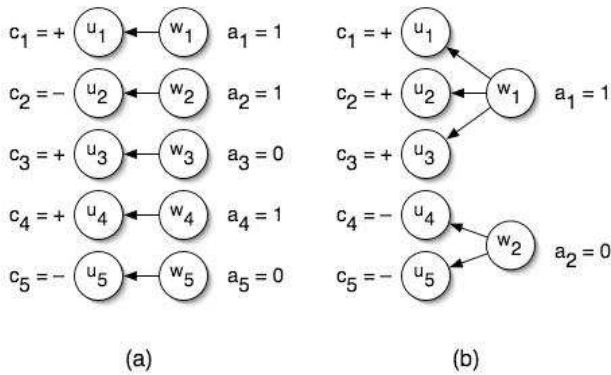

Figure 2: Propositional and relational data fragments.

Learning from relational data is becoming an increasingly common task for machine learning and data mining algorithms [Dzeroski & Lavrac 2001]. Methods now exist for learning relational versions of Bayesian networks, Bayesian classifiers, logical rules, and logistic regression

equations. All of these methods face the basic task of distinguishing pattern from noise in relational data.

Returning to our example, if we generate many relational data sets with high linkage and autocorrelation and score the relationship between $A=\{a_1,a_2,...a_n\}$ and $C=\{c_1,c_2,...c_n\}$ using a chi-square statistic, we obtain a sampling distribution such as the one shown in Figure 1 (labeled *Relational*). Clearly the propositional distribution is a poor approximation to the sampling distribution appropriate for the relational data. Our prior inference — that we could reject the null hypothesis of independence — is now reversed. Given the new sampling distribution, we cannot reject the null hypothesis. As we will show later, if we are to accurately test hypotheses in relational data with linkage and autocorrelation we cannot use hypothesis tests developed for propositional data.

In this paper, we describe and evaluate an alternative procedure — a novel form of *randomization test* — that provides accurate hypothesis tests for relational data. This procedure modifies existing techniques for randomization tests that have been widely applied to propositional data over the past two decades. However, a novel version of this procedure is needed for relational data. We show that our procedure produces unbiased inferences in situations where a more obvious adaptation of procedures for propositional data will fail.

2.3 Linkage and Autocorrelation

New tests are needed because of the statistical dependencies among data instances that arise in relational data. These dependencies violate the assumptions of conventional hypothesis tests. The dependencies can be characterized by the *linkage* and *autocorrelation* present in a given relational data set. This paper extends work reported last year by Jensen and Neville [2002] that identified these two characteristics of relational data and demonstrated that they can introduce substantial bias into the feature selection of relational learners.

The quantitative measure of linkage L indicates the degree to which many objects in U are connected through a small number of objects W. Given particular types of relational structure in a data set, L can be calculated analytically from the sufficient statistics $|U|$ and $|W|$. Perhaps the simplest type of linkage is that shown in Figure 2b, where a many-to-one relation holds between U and W. In this case, $L = (|U|-|W|)/|U|$, so for Figure 2b, $L(U,P,W) = 0.6$.

The quantitative measure of autocorrelation C' indicates the degree to which a variable on U is correlated with the values of the same variable on other objects U connected through paths P. For example, for the data in Figure 2b, paths in P could run from objects in U through objects in W to other objects in U. Four such paths exist (u_1u_2, u_1u_3, u_2u_3, and u_4u_5), and all objects U connected by such paths have the same value of the class label, thus the data in Figure 2b has perfect autocorrelation. Additional details on measuring linkage and autocorrelation can be found in Jensen and Neville [2002].

Empirically, we have observed high levels of linkage and autocorrelation in many relational data sets. Figure 3 shows the levels of linkage and autocorrelation in two common data sets used for relational learning. Figure 3a shows results for data drawn from the Internet Movie Database, a public resource on movies (*www.imdb.com*). Each point represents the linkage of movies with respect to the specified object type (e.g., studios) and the autocorrelation of a binary class label on movies (whether the movie's box office receipts totaled more than $2 million for its opening weekend) with respect to the specified object type. For example, the point at the top of the figure indicates that linkage of movies through studios is high, and autocorrelation of movie receipts through studios is moderately high. Figure 3b shows results for data drawn from Cora, a database of technical papers constructed and processed automatically using machine learning techniques [McCallum, Nigam, Rennie & Seymore 1999]. Points represent linkage of papers with respect to other object types and the autocorrelation of paper topic (e.g., neural networks) with respect to those other object types.
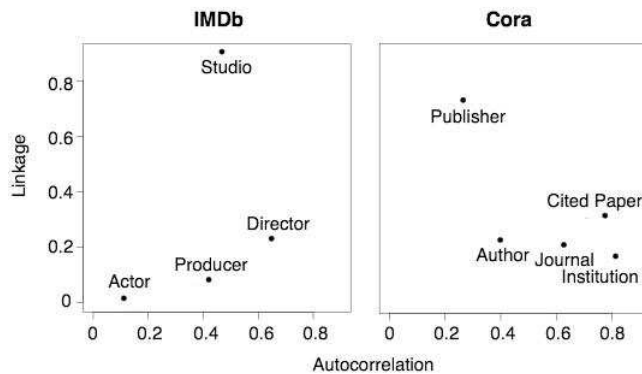

Figure 3: Linkage and autocorrelation for IMDb (movies and their receipts) and Cora (papers and their topics).

High levels of linkage and autocorrelation introduce dependencies in relational data sets that can act to reduce the "effective" sample size of those sets. Consider the data shown in Figure 2b. Do these instances provide five independent instances $<u,v>$ that indicate the correlation between A and C? Alternatively, given the apparently perfect autocorrelation and high linkage, is this set equivalent to only two instances (one for each user)? The experiments in Jensen & Neville [2002] indicate the latter. As a result, different relational features can have different effective sample sizes. Accurate statistical tests for these features can require widely differing sampling distributions. This implies that two features can have identical scores, but one feature can be highly significant while the other fails its hypothesis test.

2.4 Randomization Tests

Due to the biases that linkage and autocorrelation introduce into conventional tests, we explored alternative

forms of hypothesis tests. A randomization test is a type of computationally intensive statistical test [Edgington 1980, Noreen 1989, Good 1994]. Other types include resampling and Monte Carlo procedures.[1] Each of these tests involves generating many replicates of an actual data set — typically called *pseudosamples* — and using the psuedosamples to estimate a 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*.

For example, consider the problem of forming a sampling distribution for the problem given in the introduction. A sampling distribution for the chi-square score for A and C can be obtained via a randomization test. First, we generate all possible pseudosamples, where each pseudosample contains a unique permutation of the values of C. Second, we calculate a score for each pseudosample, using the same estimator used on the actual data (here, chi-square). The set of all scores constitutes a sampling distribution for X. Finally, we determine the percentage of scores in the sampling distribution that equal or exceed the actual score x_{actual}. This value, $p(X_{random} \geq x_{actual})$ approximates $p(X_i \geq x_{actual} | H_0)$, the probability that any particular value drawn from the sampling distribution will be at least as large as x_{actual}.

In contrast to conventional hypothesis tests, randomization tests make a relatively small number of assumptions about the data. For example, randomization tests make no assumptions about the form of the distributions from which variable values are drawn. In addition, they can be used to form sampling distributions for estimators whose precise statistical properties are not known.

The primary disadvantage of randomization tests is their computational cost. In their pure form, all possible pseudosamples are used to create the sampling distribution. Even moderately large sample sizes cause the number of pseudosamples to become extremely large. For a sample of N instances and a class label with a uniform binary class, the number of possible permutations is $N!/(2(N/2)!)$. For a mere twenty instances, 184,756 unique psueodsamples exist. Fortunately, only a fraction of all possible pseudosamples is necessary to obtain a good estimate of the sampling distribution for many purposes. For example, 1000 pseudosamples is adequate to obtain a reasonable estimate of the 5% or 10% critical values of a sampling distribution. Another optimization is to sequentially generate pseudosamples and employ an "early stopping" criteria. If, for example, a test will use 1000 psuedosamples to establish whether a given score is

---

[1] The method used to produce Figure 1 was a form of Monte Carlo procedure.

significant at the 5% level, and the first 50 of the pseudosamples all produce scores above the given score, then it clearly cannot be significant. We employ both of these optimizations in the work reported below.

3 Randomization Tests for Relational Data

We evaluated three alternatives for creating a randomization test for relational data. First, we considered *propositionalizing* the relational data and then running a conventional randomization test on the resulting propositional data. The first part of this approach is often called "flattening" a relational data set. For example, a set of relational data on papers, journals, and authors could be propositionalized by creating one instance per paper. This would duplicate the journals (because a one-to-many relation exists between journals and papers) and aggregate authors (because a one-to-many relation exists between papes and authors). Propositionalizing converts a relational task to a propositional one, with some loss of information about the relational structure of the data.

In this case, however, the lost information is essential to an accurate hypothesis test. Consider the two data sets in Figure 2. They would be identical after propositionalizing, as would their inferred sampling distributions. Propositionalizing destroys the information about both linkage and autocorrelation, making it impossible for an hypothesis test to adjust for the effects of these two characteristics of relational data.

Second, we considered retaining the relational structure of the data and *randomizing class labels*. For example, in a data set of papers, journals, and authors, we would randomize the class label on papers (e.g., paper topic) and retain all other elements of the original data. This is the most obvious adaptation of randomization tests to relational data. Randomizing class labels retains the linkage of the original sample in each pseudosample, but it destroys the autocorrelation among class labels. As a result, the pseudosamples have an effective sample size that is equivalent to a propositionalized data set, and the resulting sampling distribution will be equivalent to that estimated from propositionalized pseudosamples.

Finally, we considered retaining the relational structure of the data and *randomizing attribute vectors* associated with each object type. For example, in the case of papers, journals, and authors, this approach would randomize the attribute vectors of papers (e.g., length and type), journals (e.g., cost and circulation), and of authors (e.g., age and gender). This is equivalent to randomizing the linkage between the object containing the class label and other object types (although this interpretation leaves unexplained how to handle non-class attributes that are intrinsic to the object with the class label (e.g., paper type)).

Using this approach, pseudosamples retain the linkage present in the original sample and the autocorrelation among the class labels. Randomizing attribute vectors destroys the correlation between the attributes and the class in pseudosamples, thus making them appropriately conform to the null hypothesis. In addition, this approach destroys any autocorrelation among the attribute values, but we have yet to identify biases caused by this side effect.

To evaluate the performance of randomizing attribute vectors, we compared it to classical hypothesis tests rather than propositionalizing or randomizing class labels. As we discuss above, these latter approaches are equivalent, but less exact than, classical hypothesis tests. Thus, a classical hypothesis test provides an upper bound on the performance of propositionalizing or randomizing class labels.

We ran experiments and simulations to examine the degree of divergence between the two alternative approaches to hypothesis testing in relational data.
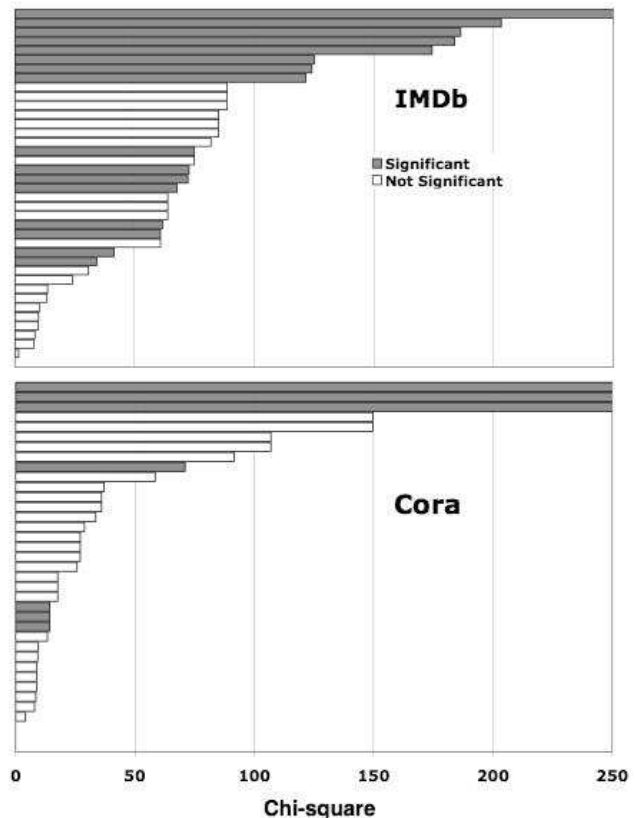

Figure 4: Relative rankings and hypothesis test results on relational features for IMDb and Cora.

3.1 Does Relational Randomization Change Learning on Real Data?

Our first experiment amounts to a "sanity check": Does randomizing attribute vectors result in substantially different inferences than a classical test? We applied both approaches to the relational learning tasks on the data described earlier (IMDb and Cora). On IMDb, we examined how features formed from different intrinsic and
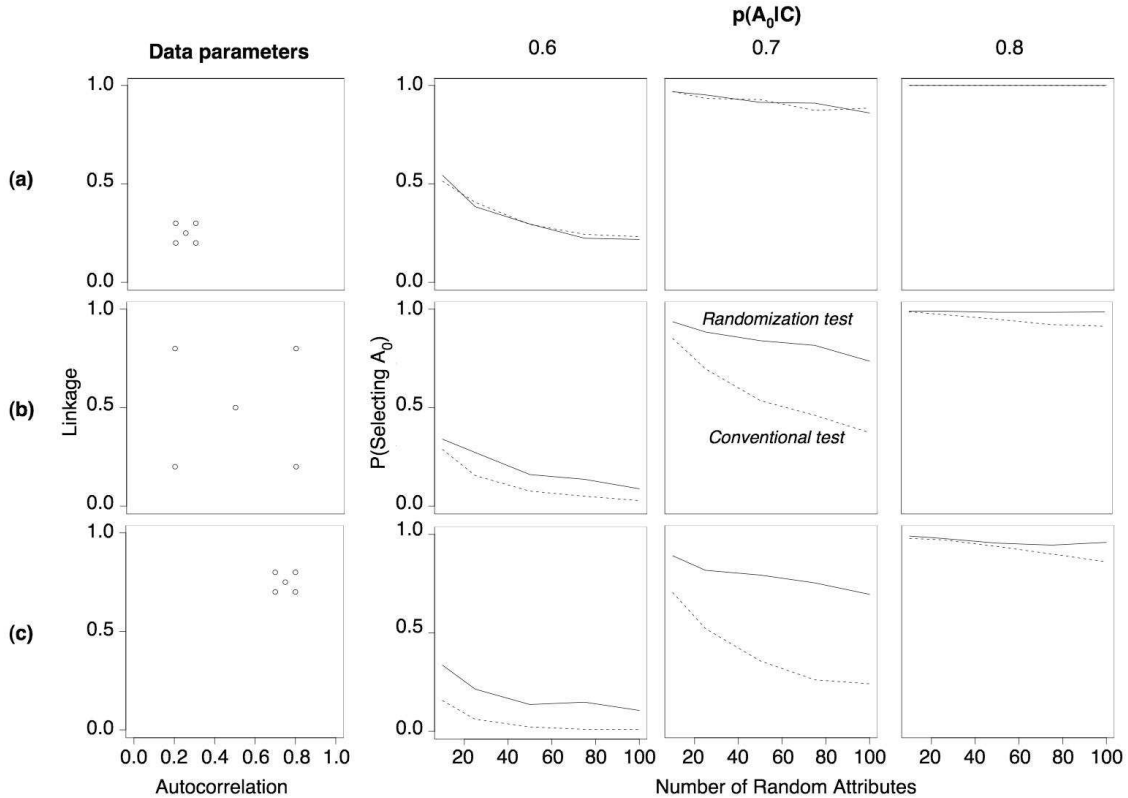
Figure 5: The probability of selecting a predictive attribute varies with the correlation of that attribute with the class label, the number of random attributes, the linkage and autocorrelation of the data, and the type of hypothesis test used.

relational attributes on movies, studios, producers, directors, and actors predicted movies success, as measured by the *receipts* class label described earlier. On Cora, we examined how features formed from attributes on papers, journals, and authors predicted paper topic. Rather than learn full models, we evaluated individual features using both approaches. This experimental protocol allowed us to examine the decisions made on all_ features, rather than only the top-ranked feature.

Figure 4 shows results for each data set. The length of each horizontal bar indicates the chi-square score of a feature. Features are ranked by score. The shading of each bar indicates whether a hypothesis test using randomized attribute vectors accepted or rejected the null hypothesis of independence at the 10% level. Dark shading indicates a significant association between the given feature and the class label. In the absence of the randomization test, features in each figure would be deemed significant if they had a score exceeding 9.1, a Bonferroni-corrected critical value for α=0.10.

While all the top-ranked attributes in these data sets would be deemed significant under either approach, some highly ranked features are dropped by the randomization test. The specific features that are dropped is instructive. For example, the top-ranked features that are *not* significant in the IMDb data are all formed from attributes on studios, the object type with the highest combination of linkage and autocorrelation in the movie data. Similarly, all features in the Cora data using attributes on publishers

and journals are dropped, despite their high raw chi-square scores.

3.2 When Do Linkage and Autocorrelation Affect Results?

Our second experiment examines the accuracy of feature ranking under varying conditions of linkage and autocorrelation. Specifically, we examine data sets that contain a single attribute (A_0) correlated with the class label (C) and k noise attributes that are generated independently of the class label. Ideally, hypothesis tests should eliminate features formed from noise attributes, allowing A_0 to be ranked first.

We ran a Monte Carlo simulation that created data sets with specified distributions of linkage and autocorrelation. For each distribution, we generated data sets with 500 objects U containing a binary class label C and a binary attribute A_0. In addition, each data set contained five other sets of objects W_1, W_2, W_3, W_4, and W_5. On each object set, we created between 1 and 20 random binary attributes. Thus, between 6 and 101 attributes were available to form features to predict C. Only one of these features, however (the feature formed from A_0), ever has any genuine predictive power.

For each data set, we scored the correlation between C and features formed from each attribute A_i using chi-square. We used two approaches to determine the top-ranked feature. The first approach ranked exclusively by chi-square. The second approach ran randomization tests

by randomizing attribute vectors and then selected the top-ranked significant feature (in the unlikely event that no features were significant, the top-ranked insignificant feature was selected). Experimental outcomes were measured by the proportion of 500 trials in which A_0 was selected.

Figure 5 shows the results of nine sets of experiments. Each set of experiments uses a different level of correlation between C and A_0 and a different distribution of linkage and autocorrelation. The levels of correlation are shown across the top of the chart ($p(A_0=1|C=+) = p(A_0=1|C=+) = p(A_0|C) \in \{0.6, 0.7, 0.8\}$). The distributions of linkage and autocorrelation are shown by graphs along the leftmost column of the figure. Within each set of experiments, we varied the number of noise attributes k.

The results show that randomization tests increase the probability of selecting the correct attribute A_0 when that attribute is moderately predictive ($p(A_0|C) = 0.7$) and when at least some of the objects in the data set have high linkage and autocorrelation with respect to the class label and its corresponding object type (labeled b and c). The methods are nearly indistinguishable when the A_0 is either a very poor or very good predictor ($0.6 \leq p(A_0|C) \leq 0.8\}$). Similarly, the methods are indistinguishable and when linkage and autocorrelation are relatively low (e.g., a).

4   Conclusions and Future Work

A particular type of randomization test — *randomizing attribute vectors* — can adjust for the strong biases introduced by linkage and autocorrelation. These biases are present in typical relational data sets such as IMDb and Cora, and they can affect the ability of learning algorithms to correctly rank different features. Conventional hypothesis tests or more obvious adaptations of conventional randomization tests to relational data cannot adjust for the biases.

Intriguingly, these sorts of hypothesis tests may be necessary for learning in data sets that are usually presented in propositional form. If a propositional data set is drawn from a domain that is inherently relational, and that domains exhibits strong linkage and autocorrelation, then pervasive bias may occur if these original properties of the domain are not reflected in the propositional data. This raises the spectre that hidden biases may lurk in many analyzes of supposedly propositional data, and the topic deserves additional study.

In addition, we have studied only one simple method for generating randomized relational data — randomizing attribute vectors. If a generative model for linkage and autocorrelation could be devised, it would provide a more universal method for creating randomized data sets. Such a model should generate both graph structure and attributes on many types of objects and links. This prospect is challenging, but appears within reach.

References

[Dzeroski & Lavrac 2001] S. Dzeroski and N. Lavrac (editors) (2001). Relational Data Mining. Berlin: Springer.

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

[Frank & Witten 1998] E. Frank and I. Witten (1998). Using a permutation test for attribute selection in decision trees. In Shavlik, J., ed., *Machine Learning: Proceedings of the Fifteenth International Conference*, Madison, Wisconsin. Morgan Kaufmann Publishers, San Francisco, CA, 152-160.

[Good 1994] P. Good (1994). *Permutation Tests: A Practical Guide for Testing Hypotheses*. Springer-Verlag.

[Jensen & Neville 2002] 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.

[McCallum, Nigam, Rennie & Seymore 2000] Andrew McCallum, Kamal Nigam, Jason Rennie, Kristie Seymore (2000). Automating the contruction of Internet portals with machine learning. *Information Retrieval Journal* 3:127-163. Kluwer.

[Megiddo & Srikant 1998] N. Megiddo and R. Srikant (1998). Discovering predictive association rules. *Knowledge Discovery and Data Mining (KDD-98)*. 274-278.

[Noreen 1989] E. Noreen (1989). *Computer-Intensive Methods for Testing Hypotheses*. New York: Wiley.

[Spirtes, Glymour, and Scheines 1993] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*.