LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-417025

Evaluating Statistical Tests forWithin-Network Classifiers of Relational Data

Jennifer Neville, Brian Gallagher, Tina Eliassi-Rad

September 22, 2009

IEEE International Conference on Data Mining
Miami, FL, United States
December 6, 2009 through December 9, 2009

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

Evaluating Statistical Tests for Within-Network Classifiers of Relational Data

Jennifer Neville[†] Brian Gallagher[*] Tina Eliassi-Rad[*]

[†]Purdue University
neville@purdue.edu

[*]Lawrence Livermore National Laboratory
{bgallagher,eliassi}@llnl.gov

Abstract

Recently a number of modeling techniques have been developed for data mining and machine learning in relational and network domains where the instances are not independent and identically distributed (i.i.d.). These methods specifically exploit the statistical dependencies among instances in order to improve classification accuracy. However, there has been little focus on how these same dependencies affect our ability to draw accurate conclusions about the performance of the models. More specifically, the complex link structure and attribute dependencies in network data violate the assumptions of many conventional statistical tests and make it difficult to use these tests to assess the models in an unbiased manner. In this work, we examine the task of within-network classification and the question of whether two algorithms will learn models which will result in significantly different levels of performance. We show that the commonly-used form of evaluation (paired t-test on overlapping network samples) can result in an unacceptable level of Type I error. Furthermore we show that Type I error increases as (1) the correlation among instances increases and (2) the size of the evaluation set increases (i.e., the proportion of labeled nodes in the network decreases). We propose a method for network cross-validation that combined with paired t-tests produces more acceptable levels of Type I error while still providing reasonable levels of statistical power (i.e., Type II error).

1. Introduction

The seminal work of Dietterich [3] focused on enumerating the types of statistical questions that analysts could ask of the models and algorithms that they develop and/or learn. His work outlined a taxonomy of questions that differentiates between algorithm and model performance, and whether the goal is to estimate accuracy or to choose between models/algorithms. Within this taxonomy, Dietterich formulated the question that is most central to data mining and machine learning research: *Given two learning algorithms A and B and a dataset of size S from a domain D, which algorithm will produce more accurate classifiers when trained on other datasets of size S drawn from D?* This question explicitly formulates the notion of generalization and provides a means to test the notion statistically.

Within this framework, Dietterich analyzed the characteristics of five statistical tests that can be used to assess generalization performance and showed that two of the tests in widespread use (at that time) had a high probability of Type I error (i.e., the tests will likely lead to an erroneous conclusion of algorithm difference when there is none). Overall, Dietterich's work showed that the overlap in training/test sets combined with *imbalanced* samples can lead to higher Type I errors due to biased estimates of mean performance difference between two algorithms. Therefore, a methodology that reduces the overlap between the training and test sets leads to lower Type I errors. Based on this analysis, Dietterich developed a novel 5x2 cross-validation test, which has lower Type I error than the standard cross-validation test but slightly worse statistical power (i.e., higher Type II error).

However, Dietterich's work only considered i.i.d. data where the instances are independent. In this work, we consider the task of comparing algorithm performance on the task of *within-network* relational learning. *Within-network* relational learning aims to generalize within a single relational data graph—models are learned on a partially labeled network and then applied to predict the class labels in the remainder of the network (i.e., the unlabeled portion). In many real world applications, relational learning tasks fall naturally into the within-network classification setting. For example, in the task of research paper classification, new papers to be classified usually have citation links to papers in the past whose topics are known. Similarly, in fraud detection, brokers whose fraud status is yet to be determined might associate with other brokers who have already been identified as fraudulent or not.

Within-network relational learning tasks have two characteristics that can complicate the application of conventional statistical tests for comparing generalization performance. First, the instances in the network are not inde-

pendent. Indeed, relational learning algorithms are specifically trying to exploit the dependencies among instances to improve prediction accuracy. The dependencies among instances, however, tend to result in correlated errors among the instances. These correlated errors can increase the imbalance between network samples and this can lead to increased Type I errors. Second, the size of the training and test sets are dependent and thus as the proportion of labeled data decreases, the size of the test set increases. This results from the fact that the models are learned/applied to a partially labeled network with varying levels of labeled instances and the full set of unlabeled instances are typically used for evaluation. As the size of the unlabeled set increases, the dependencies between samples increases and this can also lead to increased Type I errors.

In this paper, we consider the following question: *Given two learning algorithms A and B and a partially-labeled network from domain D, and with S_L labeled instances and S_U unlabeled instances ($S = S_L + S_U$), which algorithm will produce more accurate classifiers when trained on other partially-labeled networks of size S drawn from D?* We investigate the performance of a number of common statistical tests, using both simulated and real classifiers, and both synthetic and real datasets. The experimental methodology and empirical results for each combination can be found in the following sections:

	Simulated Classifiers	Real Classifiers
Synthetic Data	*Section 4*	*Section 5*
Real Data	*NA*	*Section 6*

Our findings indicate that a commonly-used method of statistical assessment—paired t-tests on repeated samples of randomly selected network samples (labeled training set and unlabeled test set)—results in unacceptably high levels of Type I error. We propose a method for *network cross-validation* that combined with unpaired t-tests produces low levels of Type I error at the expense of reduced statistical power. Combining network cross-validation with paired t-tests is a good compromise, resulting in both acceptable levels of Type I error and reasonable levels of statistical power. The contributions of this work include:

- Formulation of important statistical questions for comparing network classifiers.
- Discussion and demonstration of the challenges of network data for using conventional statistical tests to compare classifiers.
- A proposed solution, *network cross-validation*, which addresses these challenges.
- Empirical evaluation of statistical test characteristics (Type I error/power) on real-world and synthetic data.

Data Set	**Task**	**Error Corr.**	**Autocorr.**
Enron Email	Executive?	0.18	0.17
Citeseer	Neural Nets?	0.23	0.59
Political Books	Neutral?	0.25	0.22
Cora	Info. Retrieval?	0.28	0.61
Reality Mining	In Study?	0.32	0.79
Reality Mining	Student?	0.52	0.91

Table 1. Error correlation and relational autocorrelation in real-world classification tasks.

2. Error correlation in relational domains

In the previous section we described two sources of Type I error in within-network classification:

1. Inter-instance dependencies lead to correlated errors.
2. Small training sets lead to large test sets, increasing the dependence between samples.

In Section 3.1 we describe how existing resampling procedures create dependent samples. Here, we demonstrate that within-network classifiers produce correlated errors.

To test the conjecture that within-network classifiers produce correlated errors, we experimented with several relational classifiers and real-world classification tasks, using the ϕ coefficient to measure the correlation of 0-1 errors over all pairs of related (i.e., linked) instances. We used a non-learning relational neighbor classifier [13] and a learning link-based classifier [12]. We ran each classifier both with and without collective inference on a number of prediction tasks. Table 1 shows the amount of measured error correlation for each task, averaged over all classifiers and proportions of labeled data $(0.1, 0.3, 0.5, 0.7, 0.9)$. Although we report averages, we should note that all trials (i.e., tasks/classifiers) exhibited some degree of error correlation. We can observe that the level of error correlation is correlated with the level of relational autocorrelation (see e.g., [18]) in the class label. Since autocorrelation has been shown to be essentially ubiquitous in relational data, this suggests that error correlation is widespread as well.

3. Comparing classifiers in network domains

Statistical tests for comparing classifiers generally consist of two parts: (1) The *resampling procedure* dictates how the available data is partitioned into training and test sets for estimation of classifier performance (i.e., *how many times is the classifier trained and tested?*, *which data is used to train the classifier?*, *which data is used to test the classifier?*) and (2) The *significance test* takes the classification results from the resampling trials and makes a determination as to whether observed differences reflect a true difference in classifier performance or whether it is likely to have occurred by chance alone.

3.1. Resampling procedures

Given a fully labeled network of size S, we consider three resampling procedures to generate training (labeled set S_L) and test (unlabeled set S_U) sets to evaluate within-network classification algorithms: *simple random resampling* (RRS), *equal-instance random resampling* (ERS), and *network cross-validation* (NCV). The first two methods have been used extensively in past work on relational learning algorithms (see Section 3.3 for more detail). The third method is a new approach, based on the incremental cross-validation procedure outlined in Cohen [2] for generating learning curves, which will be more robust to Type I error.

Tables 2 and 3 outline the procedures for RRS and ERS, respectively. Both methods involve repeated random draws from the sample population to generate the training/test splits; and, therefore, produce overlapping test sets. However, ERS ensures that each instance in the original sample occurs in exactly the same number of test sets in the collection of resamples.

Table 4 outlines the NCV procedure that eliminates overlap between test sets altogether. The procedure samples for k disjoint test sets that will be used for evaluation. Then for each test set fold, the remaining folds are merged together and the training set of size S_L is randomly sampled from the merged set. When the training set size is less than the size of the merged folds (i.e., $S_L < (k-1)\frac{S}{k}$), this will leave a set of unlabeled nodes that are neither in the test set nor the training set. Since these unlabeled instances will likely be connected to nodes in the test set, we will run collective inference over the full set of unlabeled nodes (the *inference* set), and then only evaluate model performance on the nodes assigned to the test set.

NCV addresses a limitation of standard cross-validation for within-network classification tasks. Namely, standard CV forces us to label $k-1$ of every k instances. So, if $k = 10$, we are forced to experiment with 90% labeled data. The NCV approach accommodates a lower proportion of labeled instances because it samples a smaller labeled set from the $k-1$ non-test folds. However, since NCV applies collective inference to the full unlabeled portion of the network but only *evaluates* the model on the disjoint test set instances, it will not suffer the same problems experienced by resampling due to overlapping test sets.

3.2. Significance tests

Once a sampling procedure is chosen to create training/test splits within a network, the algorithms are learned on each training set and then the models are applied for collective inference over the associated unlabeled portion of the network. The predictions on the test set instances are evaluated to generate an estimate of algorithm performance (e.g., accuracy, AUC, squared loss). The training/test splits

---

input: $network$, $propLabeled$, k
S = total number of instances in $network$
$F = \emptyset$
for $fold$ 1 to k
  $testSet$ = uniform random sample of
    $((1 - propLabeled) * S)$ nodes from $network$
  $trainSet$ = $network - testSet$
  $F = F \cup < trainSet, testSet >$
end for
output: F

Table 2. Simple random resampling procedure.

---

input: $network$, $propLabeled$, k
S = total number of instances in $network$
// Split data into overlapping folds so that each instance occurs
// in the same number of folds.
$testSetSize = (1 - propLabeled) * S$
$numCopies = \frac{k * testSetSize}{S}$
$pool$ = sorted list with $numCopies$ of each instance in network
$testSet[i]$ = {}, for i = 1 to k
for $instance \in pool$
  add $instance$ to smallest $testSet[i]$ that does not already contain it
end for
// create training/test splits
$F = \emptyset$
for $testSet$ 1 to k
  $trainSet$ = $network - testSet$
  $F = F \cup < trainSet, testSet >$
end for
output: F

Table 3. Equal-instance resampling procedure.

results in a set of performance measurements for each algorithm and a significance test is then used to determine whether the observed performance differences are *significantly* different than what would be expected if the performance measures were drawn from the same underlying distribution (i.e., the algorithms perform equivalently).

In this work, we investigated the following three statistical tests: (1) paired t-test, (2) unpaired t-test, and (3) Wilcoxon signed rank test. Both the paired t-test and Wilcoxon signed rank test assume independence of the paired differences between classifiers. We observed no substantive differences due to the use of the Wilcoxon test vs. the t-test. Therefore, we focus on the more commonly used t-test for the remainder of this paper.

3.3. Survey of previous methodology

Over the past 10 years, there has been a great deal of work on classifiers for relational domains. Here we survey

given: $network$, $propLabeled$, k
S = total number of instances in $network$
$F = \emptyset$
Split data into k disjoint folds
for $fold$ 1 to k
  current $fold$ becomes $testSet$
  remaining folds are merged and become $trainPool$
  $trainSet$ = uniform random sample of $(propLabeled * S)$
    nodes from $trainPool$
  $inferenceSet$ = $network - trainSet$
  $F = F \cup < trainSet, testSet, inferenceSet >$
end for
output: F

Table 4. Network cross-validation procedure.

the methodological design of 23 research papers most relevant to our work [1, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. Relevant papers compare the performance of two or more classifiers on a relational classification task. Based on our survey, there are two common evaluation methodologies which emerge:
(1) **Independent set size**. The salient feature of the independent set size methodology is that there is no dependency between training and test set sizes. The problem may be either within-network or across-network classification (i.e., there may or may not be relations between instances in the labeled and unlabeled set). For across-network classification, classifiers are generally trained on a fully labeled training network and then evaluated on a disjoint (partially labeled) test network. In this case, the proportion of labeled data available in the training and test networks may be varied independently. This means that there is no dependence between training and test set sizes. For within-network classification, there is a single network and classifiers are trained on the labeled instances and evaluated on the unlabeled instances. So, the only way to achieve independent set sizes in within-network classification is to fix the proportion of labeled data available in the network. All of the papers in our survey that employ independent set sizes (15 out of the 23) use some form of random resampling or cross-validation for evaluation.
(2) **Dependent set size**. This methodology applies to within-network classification only, where training and test sizes are dependent. Any change in the labeling proportion over the network will affect the number of instances available for both training and testing. All of the papers in our survey that employ dependent set sizes (9 out of the 23) use some sort of random resampling for evaluation (i.e., test sets overlap). Note that standard cross-validation is not possible here since it assumes a fixed proportion of labeled data.

Both of the above methodologies generally address the statistical question that we consider in our work: *Given two learning algorithms A and B and a partially-labeled network from domain D, and with S_L labeled instances and S_U unlabeled instances ($S = S_L + S_U$), which algorithm will produce more accurate classifiers when trained on other partially-labeled networks of size S drawn from D?* However, the independent set size methodology makes a rather strong assumption about the value of S_L. In particular, most studies that employ independent set sizes use 10-fold cross validation, which means that their results only generalize to graphs where 90% of the data is labeled to begin with. This is limiting because: (1) most interesting real-world problems have far less than 90% of data labeled to begin with and (2) many algorithms that perform well at 90% labeled will perform poorly at sparser labelings (e.g., 10%). The dependent set size methodology is more general and powerful since it generalizes over different values of S_L, so we focus on this version in our work.

Resampling Procedure		**Systematic Variation of % Labeled**	
Cross validation	14	No	13
Simple random	8	Yes	10
Controlled random	3	Within-network	8
Snowball sampling	2	Across network	2
Temporal resampling	2		
Statistical Test		**Number of Resampling Folds**	
t-test	10	10	14
StDev/Var/StErr	6	<10	7
None	6	>10	2
Wilcoxon signed rank	2	Unspecified	2
Within vs. Across Network Classification		**Performance Measure**	
Within-network	13	Accuracy	14
Across network	8	AUC	10
Unspecified	6	Precision/Recall/F1	1

Table 5. Experimental characteristics of 23 surveyed papers. Note that section counts do not necessary sum to 23 since papers can fit in >1 category.

Table 5 provides a summary of related work along a number of methodological dimensions. Note that counts in each section do not necessary sum to 23 since papers may fit in more than one category. The majority of the studies in our survey (13/23) make use of resampling procedures that produce overlap between test sets. This includes all resampling procedures except cross validation and temporal sampling. *Temporal sampling* involves training on past instances (e.g., previously published papers with known topics) and evaluating on present instances (e.g., a newly submitted paper with unknown topic). *Controlled random sampling* procedures attempt to control or account for the amount of overlap between test sets (e.g., as in the *equal-instance* resampling procedure described in Section 3.1).

In our survey, within-network classification tasks are

more common than across-network tasks (13 papers vs. 8). However, in a substantial number of cases, it is unclear exactly how the experiments are set up. For example, authors will often say something like: we split the network into training and test sets. It is not clear from this description whether the links are preserved between instances in the training set and instances in the test set. In other cases, authors are explicit regarding whether such links are retained or removed.

The majority of studies in our survey (13/23) do not vary the proportion of labeled data available. Of the studies that do vary the proportion of labeled data, most (8/10) are within-network studies (i.e., *dependent set size*).

The most common number of resampling folds used in the surveyed papers is 10. As Dietterich notes in his original study, the probability of Type I error for random resampling procedures increases with the number of resampled folds [3]. Our simulation experiments confirm this finding, but we do not replicate the result here.

Half of the studies in our survey do not make use of an explicit significance test. However, of these, about half do report standard deviation, variance, or standard error (*StDev/Var/StErr* in Table 5). Finally, both accuracy and AUC are common measures of classifier performance, with precision/recall-based measures being much less common.

4. Evaluating sources of Type I error for within-network classification

Type I errors occur when a statistical test incorrectly rejects the null hypothesis (i.e, the test concludes that there is a significant difference between two classifiers when there is none). We run simulation experiments in order to assess the contributions of two key factors in Type I errors for within-network classification: (1) correlation of errors among related data instances and (2) dependence between samples. We also assess the potential of various statistical tests to produce Type I errors in a within-network classification setting.

Our method preserves the basic structure of Dietterich's [3], but introduces a group-based model to more easily represent sets of related instances and vary the degree of error correlation among them. We also ran Dietterich's original procedure with qualitatively similar results.

4.1. Methodology

As we have seen, real classifiers exhibit correlated errors on sets of related instances. To simulate this behavior, we divide all data instances into disjoint groups such that classification errors are more likely to be correlated on instances within a group than on instances from different groups.

```
for simulation 1 to 10
    for trial 1 to 1000
        (NCV*) Create sample and split into 10 disjoint folds
        for propLabeled ∈ (0.1, 0.3, 0.5, 0.7, 0.9)
            (RS*) Create sample and resample 30 folds
            for each fold
                run groupBasedClassification(fold)
            end for each
            Apply significance test to all folds in trial/propLabeled
            Accept or reject null hypothesis
            Measure error correlation for this trial/propLabeled
        end for
    end for
    Calculate null hypotheses rejection rate for simulation
    Calculate mean error correlation for this simulation
end for
Calculate final mean rejection rate and error correlation
```

NCV^*: *Performed for NCV (see Table 4) only.*
RS^*: *Performed for RRS and ERS (see Tables 2-3) only.*

Table 6. Simulation algorithm. See Table 7 for $groupBasedClassification$ procedure.

Table 6 outlines our simulation algorithm for both the resampling procedures and the network cross-validation procedure. The two procedures differ only in that: (1) they use different resampling algorithms to create their test sets, and (2) NCV uses the same samples and folds across all proportions of labeled data, whereas the resampling procedures choose a different sample and random split for each trial and proportion labeled.

We simulate drawing a network sample s from an underlying population by creating 300 instances and assigning one of 10 groups to each instance uniformly at random (a skewed group-size distribution produced qualitatively similar results). We then resample s and run a simulated classification experiment on each resampled train/test split (see Table 7). For each trial, we apply a significance test to either accept or reject the null hypothesis. We calculate the proportion of trials for which the null hypothesis was rejected. Since the simulation is designed so each classifier has the same error rate in the underlying population, any rejections of the null hypothesis represent Type I errors. In addition, we measure the degree of error correlation for each trial. We use the ϕ coefficient to measure the pairwise correlation of 0-1 errors among all instances in the same group, averaged over all groups.

4.2. Results

For all experiments, we present average Type I error rates for various statistical tests over 10 simulations of 1000 trials each, on data samples of size 300 instances. Unless

```
groupBasedClassification(fold)
    NG = total number of groups
    M = round(NG * P(err))
    P(MG) = P(err) + errCorr * (1 - P(err))
    P(MG') = P(err) + (1-P(MG))/(1-P(err))
    // a real classifier would normally train here, but there is
    // no training phase since we are simulating classification
    // choose groups to misclassify
    MG_A = random set of M groups from 1 to NG/2
    MG_B = random set of M groups from NG/2 + 1 to NG
    for each instance i ∈ fold
        // simulate application of classifier A
        if group(i) ∈ MG_A
            i misclassified by classifier A with P(MG)
        else
            i misclassified by classifier A with P(MG')
        end if
        // simulate application of classifier B
        if group(i) ∈ MG_B
            i misclassified by classifier B with P(MG)
        else
            i misclassified by classifier B with P(MG')
        end if
    end for each
```

Table 7. Group-based classifier simulation algorithm. This method ensures that classifiers A and B have the same error rate, while still making different kinds of errors (i.e., A misclassifies different groups from B). M is the number of groups chosen for misclassification by each classifier. $P(MG)$ is the misclassification probability for instances of the chosen groups (MG_A/MG_B) and $P(MG')$ is the misclassification probability for instances of all other groups. $P(err)$ is the overall error rate of both classifiers A and B and $errCorr$ controls the degree of error correlation among instances within MG_A/MG_B.

otherwise noted, our default experimental parameters are: classifier error rate $P(err) = 0.1$, error correlation parameter $errCorr = 0.9$, and proportion of labeled instances $propLabeled = 0.9$. The $errCorr$ parameter determines the likelihood of misclassifying instances in the chosen misclassification group vs. instances in other groups.

Figure 1(a) shows the effects of varying the proportion of labeled data available (0.1, 0.3, 0.5, 0.7, 0.9). For both resampling procedures, the Type I error rate increases as $propLabeled$ decreases. This result is expected since the degree of overlap between test sets increases as the test sets become larger due to the larger number of unlabeled instances. Since NCV disallows overlapping test sets by design, it is not susceptible to this problem, achieving low Type I error rates across the range of $propLabeled$ values.

Figure 1(b) shows the effects on measured error correlation and Type I error rate as we vary $P(err) = [0.1, 0.2, 0.3, 0.4]$ and $errCorr = [0, 0.2, 0.4, ..., 1.0]$. We can observe that the Type I error rate of the resampling procedures increase as the error correlation increases. This is not surprising since increased error correlation is expected to lead to increased imbalance between samples. In addition, we note that our experiments showed, for a fixed value of $P(err)$ ($errCorr$), both the measured type I error rate and the measured error correlation increased monotonically with $errCorr$ ($P(err)$). Overall, NCV is less affected by imbalanced samples since test sets do not overlap; so it exhibits much lower levels of Type I error. The Type I error rates of equal-instance resampling (ERS) are lower than simple random resampling, however since the improvement is not sufficient to make it competitive with NCV, we do not consider ERS further.

5. Evaluating suitability of statistical tests on real classifiers

This section describes our investigation of the characteristics of statistical tests when comparing real relational learning algorithms. We consider two collective inference models and compare their performance on synthetic data. The synthetic data generation enables the simulation of multiple draws of networks from the same distribution. We evaluate the Type I error and power of statistical tests, as the performance of the two models is varied.

5.1. Models

In order to run experiments at the scale needed to assess Type I error and power rates, we choose to investigate two simple and efficient collective models described in Macskassy and Provost [14].

The first model is the weighted-vote relational neighbor (wvRN), which estimates class label probabilities by assuming the existence of homophily. Given the unlabeled nodes in a network $v_i \in V_U$, wvRN estimates $P(y_i|N_i)$ as the average of the class probabilities of the instances in N_i (v_i's neighbors): $P(y_i = +|N_i) = \frac{1}{Z}\sum_{v_j \in N_i} P(y_j = +|N_j)$, where Z is a normalizing constant.

The second model is the network-only Bayes classifier (nBC), which estimates class label probabilities for v_i with a multinomial naive Bayesian model, based on the classes of v_i's neighbors N_i: $P(y_i = +|N_i) = \frac{P(N_i|+)P(+)}{P(N_i)} \propto [\frac{1}{Z}\prod_{v_j \in N_i} P(y_j = \tilde{y}_j|y_i = +)]P(+)$, where Z is a normalizing constant and $\tilde{y}_j$ is the class observed at node v_j.

The wvRN model does not require any learning. To estimate the parameters of the nBC model, we use maximum likelihood estimation over the labeled part of the network. For collective inference, we use relaxation labeling with both models.

5.2. Data

The synthetic datasets are generated with a latent group model (LGM) [17]. Each network is generated with 300 nodes (instances). The nodes are generated as members of (hidden) groups and group membership determines the binary class label values and link existence for each node. The average group size is 10 and there are two types of groups: A and B. The network is skewed towards A groups, $P(A) = 0.75$. Members of A groups are more likely to have a positive class label, $P(+|A) = 0.9$. Members of A groups also have higher intra-group linkage with $P_A(e_{ij} = 1|i \in g_k^A \wedge j \in g_k^A) = 0.6$ and lower inter-group linkage with $P_A(e_{ij} = 1|i \in g_k^A \wedge j \notin g_k^A) = 0.003$. Members of B groups are more likely to have a negative class label, $P(+|B) = 0.1$. Members of B groups have relatively lower intra-group linkage with $P_B(e_{ij} = 1|i \in g_k^B \wedge j \in g_k^B) = 0.4$ and higher inter-group linkage with $P_B(e_{ij} = 1|i \in g_k^B \wedge j \notin g_k^B) = 0.013$. The resulting networks have an average autocorrelation of 0.40 and a class prior of $P(+) = 0.70$.

Our choice of data generation parameters was designed to create networks where the wvRN and nBC would make *different* classification errors. Many of the nodes in type B groups have more links to nodes in type A groups so the networks does not fully meet the assumption of homophily which underlies the wvRN model. The nBC should thus more accurately *learn* how to classify the type B nodes, while the wvRN will likely be more accurate on the type A nodes. However, since wvRN does not *learn* the concept of homophily, it will not experience variance due to small labeled sets.

5.3. Methodology

To estimate Type I error, we need two models with equal performance on partially-labeled networks of the same size, drawn from the same domain. To achieve this, we measured the average accuracy of the wvRN and the nBC models on the synthetic data and handicapped the better model (wvRN) until the performance difference of the models was ≤ 0.005. The better performing model was handicapped by randomly selecting $c\%$ of it's predictions and perturbing those probabilities toward the opposite class. To set c, we generated 50 networks for use as a *calibration* set. Each of the 50 networks was sampled into 10-fold network cross-validation sets, resulting in 500 training/test set splits on which we measured average accuracy of each model. Using this calibration set, we searched for a value of c that resulted in a performance difference of ≤ 0.005 between the two models.

To estimate power, we need to vary the performance difference between the two models. To achieve this, we perturbed the predictions of the *worse* performing model (nBC) to increase the mean difference in performance between the two models. For the power experiments, we used perturbation rates of $c = [0.025, 0.075, 0.15, 0.3]$.

5.4. Results

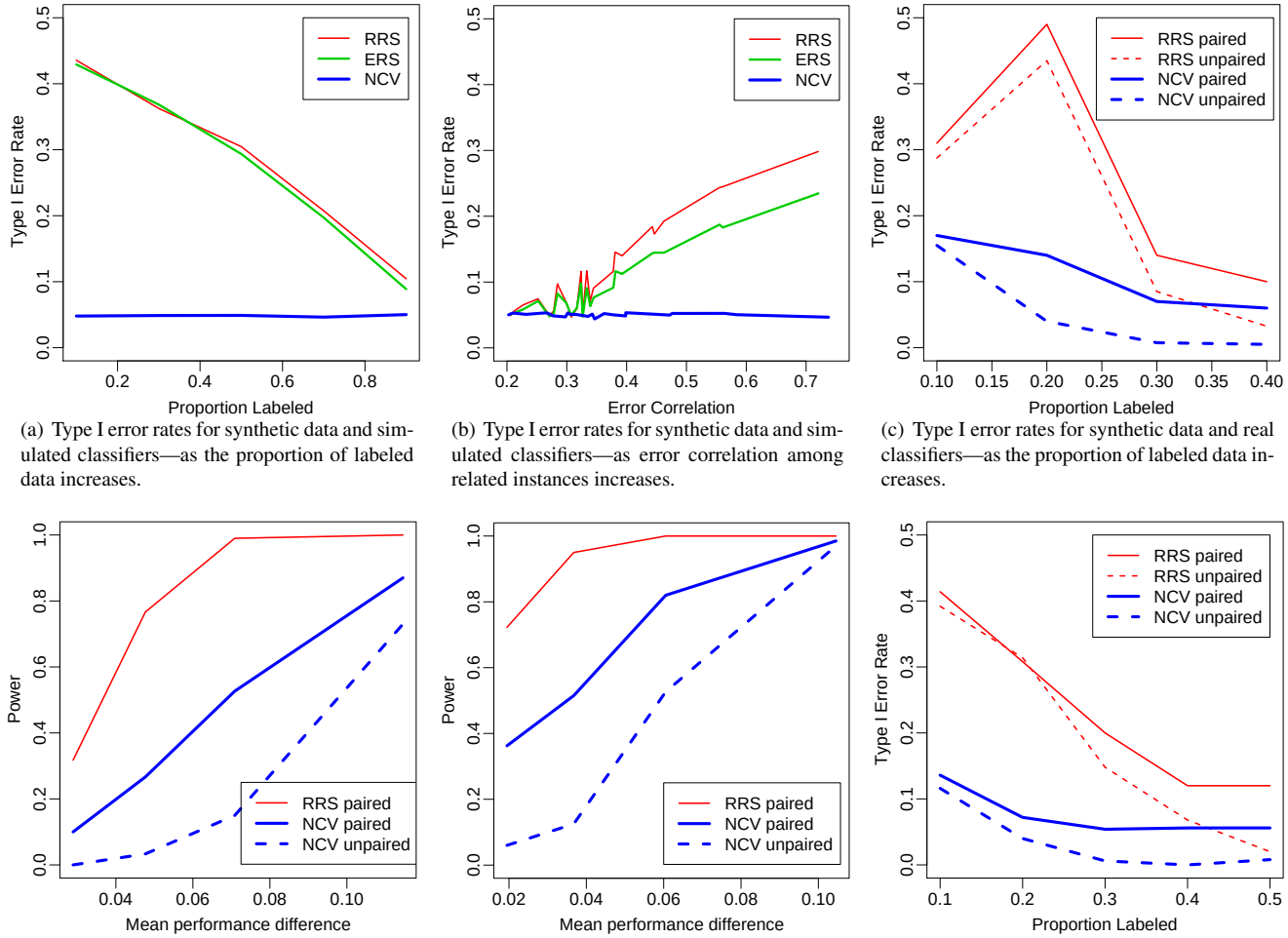
To measure Type I error rates and power of the statistical tests, we used four synthetic networks (in addition to the calibration set). On each network, we considered four levels of labeling: [0.1, 0.2, 0.3, 0.4]. At each level of labeling, we sampled the network 10 times, either by repeated sampling or cross-validation. On each of the ten samples, we learned the nBC model on the labeled portion of the network and then we applied both models to the unlabeled portion of the network with the perturbation rate c. We measured the accuracy of each model on the ten test sets and then assessed the difference in performance using a t-test.

Figure 1(c) plots the Type I error rates for four combinations of sampling method (RS, NCV) and statistical test (paired and unpaired t-test). Type I error rates for each dataset are measured over 100 trials and averaged. Recall that we use the calibration set to choose a value for c that makes the average performance of the wvRN and nBC equal at the given level of labeling. Thus any trial in which the t-test assesses the observed performance difference as *significant* corresponds to a Type I error.

All the tests have high Type I error rates with 10% of instances labeled. This error generally decreases as the amount of labeled data increases (and thus the size of the test set decreases). Since we are using relatively simple classifiers, as the number of labeled data increase model performance does indeed converge and the two models make similar classification errors at labeling rates greater than 40%. In reality, when comparing relational models with different representations and different complexity, we expect Type I error to occur at all levels of labeling.

Notably, the repeated sampling approach experiences as high as 50% Type I errors (at 20% labeling). This means that *half* the time the method is concluding a significant performance difference between the two models, when in fact there is none. For data mining and machine learning researchers that are investigating the tradeoffs between learning algorithms, this is an unacceptable level of error. On the same data, the network cross-validation procedure error rate is only 15% with the paired t-test—a 70% reduction in error. Clearly, the network cross-validation approach results in a more accurate comparison of model performance.

Note that in the simulated classifier experiments (see Figures 1(a)-1(b)), NCV across the proportions labeled is equivalent to 10-fold CV at 90% labeled, since performance in the simulated classifier experiments does not depend on: (1) the number of labeled neighbors available during inference, and (2) the number of instances available during train-



(a) Type I error rates for synthetic data and simulated classifiers—as the proportion of labeled data increases.

(b) Type I error rates for synthetic data and simulated classifiers—as error correlation among related instances increases.

(c) Type I error rates for synthetic data and real classifiers—as the proportion of labeled data increases.

(d) Statistical power as the performance difference between classifiers increases (300 node network).

(e) Statistical power as the performance difference between classifiers increases (600 node network).

(f) Type I error rates for real data (Add Health) and real classifiers—as the proportion of labeled data increases.

Figure 1. Experimental results.

ing. These additional dependencies contribute to the higher Type I error observed for NCV with real classifiers.

Although the Type I error of NCV combined with a paired t-test is much lower than resampling, it is still higher than the generally accepted level of 5%. To investigate this behavior, we examined the estimates used in the t-test calculation: (1) the estimate of the mean performance difference between the two models: $\mu_{diff} = \mu_{A_{acc}} - \mu_{B_{acc}}$ and (2) the estimate of the variance of the differences: $Var_{diff} = Var(\{A_{acc} - B_{acc}\}_i)$. The error correlation increases the variance of the estimated μ_{diff}, but the estimates are not biased. On the other hand, the error correlation decreases the variance of the estimated differences, resulting in a biased underestimate of Var_{diff}. We considered the unpaired t-test to adjust for this bias. The unpaired t-test uses a *pooled* estimate of variance over the performance measurements $\{A_{acc}\}$ and $\{B_{acc}\}$, instead of an estimate of variance of the differences. The pooled estimate of variance is higher than the variance of the differences, so it can offset the bias in variance estimation due to error correlation. This is indeed the case—combining NCV with unpaired t-tests results in Type I error rates of less than 0.05 (when proportion labeled is greater than 10%).

Figure 1(d) plots the power of each statistical test as we varied the average performance difference between wvRN and nBC. More specifically, we used $c = [0.025, 0.075, 0.15, 0.3]$ and measured the average performance difference between the two models on the calibration set. This gives us the mean performance difference that is plotted on the x-axis. Then for each of the four evaluation networks, we sampled the network 10 times and learned/applied/evaluated the models as described above. Again the results are measured over 100 trials. Since the two models *do* perform differently, any trial in which the t-test *does not* conclude that the observed performance difference is significant corresponds to a Type II error. Power is defined as the proportion of trials in which the t-test correctly concludes that the two models are different. We only

plot the results for proportion labeled of 30%. The results for other levels of labeling are qualitatively similar.

The power results illustrate an additional challenge in evaluating the performance difference between the models. Even when there is 5% difference in the mean performance of the two algorithms, it is sobering to note that repeated sampling can detect this difference less than 80% of the time. Network cross-validation is significantly worse—the paired t-test can detect the difference less than 30% of the time and the unpaired t-test less than 5% of the time. This may be due to the difference in test set size used by the two approaches. Recall that repeated sampling uses *all* the unlabeled data for evaluation, so at 30% labeling this corresponds to 210 nodes. On the other hand network cross validation uses only 10% of the nodes for evaluation (i.e., 30 nodes) regardless of the level of labeling in the network.

To explore this issue, we increased the dataset size to 600 and measured the power of each approach again. Figure 1(e) graphs the resulting power rates. In general, power of any test will be increased as you increase the sample size. However, here we can see that the gains for network cross-validation are relatively larger than for repeated sampling. It is difficult to compare across the two sets of results due to different mean performance of the models. However, if we interpolate between the results in Figure 1(e) to assess the power at 5% mean performance difference and compare to Figure 1(d) at 5%, we can see that doubling the dataset size reduced the error of repeated sampling by 10% but the network cross-validation was reduced by 45% (paired t-test) and 70% (unpaired), respectively.

6. Evaluating suitability of statistical tests on real-world data

This section describes our investigation of the characteristics of statistical tests when comparing relational learning algorithms on real-world network data. To confirm the behavior we observed in the synthetic datasets, we compared the performance of wvRN and nBC on data from the National Longitudinal Study of Adolescent Health [10].

The Adolescent Health (Add Health) data consists of survey information from 144 middle and high schools, collected in 1994-1995. The survey questions queried for the students' social networks along with myriad behavioral and academic attributes. In this paper, we consider the social networks of six schools with similar autocorrelation and link patterns. The classification task is to predict whether the student *smokes* based on the behavior of their friends in the social network. The six schools we selected have sizes ranging from 300-700 nodes, average degree of 7-8, and autocorrelation in the range [0.25,0.35].

To assess the Type I error characteristics of the models, we used a procedure similar to the one described in Section 5. Each trial considers one school network as the evaluation set, then we calibrate the models on the remaining 5 school networks under the assumption that these networks were drawn from the same distribution. We sampled each of these five networks 10 times into 10-fold NCV sets, producing a calibration set of 500 training/test splits. As described previously, we searched for a value of c that resulted in a performance difference of ≤ 0.005 between the two models. We considered five levels of labeling: [0.1,0.2,0.3,0.4,0.5], calibrated the models at each level of labeling, and measured the Type I error on the held out network. The models converge in performance at 50% labeling (i.e., $c=0$) so we do not consider labeling rates $> 50\%$.

Figure 1(f) shows Type I error for each combination of sampling method and statistical test, measured over 50 trials. As expected, the statistical tests exhibit similar behavior on the Add Health data and the synthetic data. Again resampling produces unacceptable levels of Type I error (up to 40%) and network cross-validation has more reasonable error rates. Overall error decreases as the proportion of labeled data increases to 50% (i.e., test set size decreases). Recall, however, that we are investigating simple models that are nearly equivalent on a restricted task involving only the class label and no other attribute/link features. In practice, as the complexity of models and concepts increase, Type I errors are likely to occur at all levels of labeling.

7. Conclusions and discussion

In this paper we examined the characteristics of statistical tests for comparing *within-network* classification algorithms. We presented three resampling procedures and three significance tests; performed experiments on both real and synthetic data using real and simulated classifiers.

Our analysis shows that a commonly-used form of evaluation in relational learning (paired t-tests on overlapping network samples) can result in unacceptably high levels of Type I error (as high as 50%). High Type I error indicates that many algorithm differences will be judged incorrectly as significant when in fact performance is equivalent. Although for efficiency reasons we considered relatively simple relational models for this work, our findings apply to evaluations of more complex relational models as well—since any relational model that attempts to exploit relational autocorrelation is likely to produce correlated errors.

Furthermore, we demonstrated that Type I error increases as (1) the correlation among instances increases and (2) the size of the evaluation set increases (i.e., the proportion of labeled nodes in the network decreases).

Although we investigated the properties of significance tests for within-network classification, the findings are also applicable to across-network tasks and other forms of hypothesis testing (e.g., standard error bars will be underes-

timated). The extent of the effect will depend on the level of observed autocorrelation (which will cause error correlation), as well as the amount of overlap between samples.

We proposed a method for *network cross-validation* that reduces the overlap between samples. We note that although the method creates disjoint test sets, the predictions for those test set instances will be influenced by other predictions in the unlabeled *inference* set (due to the collective inference process). This means that there is still some dependency between test sets (since the inference sets overlap) which could increase the Type I error of NCV.

Our empirical evaluation shows that NCV combined with *unpaired* t-tests results in low levels of Type I error. However, this low error is achieved at the expense of statistical power (i.e., Type II error). NCV combined with *paired* t-tests produces more acceptable levels of Type I error while still providing reasonable levels of statistical power.

Promising research directions include: (1) using patterns (such as communities) in relational data to split train/test data (e.g., stratified by community, or biased by community); (2) an investigation of non-random labeling patterns and their impact on error correlation for different collective inference methods; and (3) investigating how characteristics of relational data affect the power of statistical tests (i.e., Type II error).

Acknowledgments

We thank Rongjing Xiang for her assistance in experimental implementation. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. This was also supported by DARPA and NSF under contract numbers NBCH1080005 and SES-0823313.

References

[1] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD'98*, pages 307–318, 1998.

[2] P. Cohen. *Empirical Methods for Artificial Intelligence*. MIT Press, 1995.

[3] T. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.

[4] B. Gallagher and T. Eliassi-Rad. An examination of experimental methodology for classifiers of relational data. In *ICDM'07 Workshops*, pages 411–416, 2007.

[5] B. Gallagher and T. Eliassi-Rad. Leveraging label-independent features for classification in sparsely labeled networks: An empirical study. In *SNA-KDD'08*, August 2008.

[6] B. Gallagher, H. Tong, T. Eliassi-Rad, and C. Faloutsos. Using ghost edges for classification in sparsely labeled networks. In *KDD'08*, pages 256–264, 2008.

[7] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. In *ICML'01*, pages 170–177, 2001.

[8] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models with link uncertainty. *JMLR*, 3:679–707, 2002.

[9] L. Getoor, E. Segal, B. Taskar, and D. Koller. Probabilistic models of text and link structure for hypertext classification. In *IJCAI'01 Workshop on Text Learning: Beyond Supervision*, pages 170–177, 2001.

[10] K. Harris. The national longitudinal study of adolescent health (Add Health), waves I & II, 1994-1996; wave III, 2001-2002 [machine-readable data file and documentation]. *Chapel Hill, NC: Carolina Population Center, University of North Carolina at Chapel Hill*, 2008.

[11] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *KDD'04*, pages 593–598, 2004.

[12] Q. Lu and L. Getoor. Link-based classification. In *ICML'03*, pages 496–503, 2003.

[13] S. Macskassy and F. Provost. A simple relational classifier. In *KDD'03 Workshop on Multi-Relational Data Mining*, pages 64–76, 2003.

[14] S. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *JMLR*, 8:935–983, 2007.

[15] S. A. Macskassy. Classification in networked data: A toolkit and a univariate case study. In *AAAI'07*, pages 590–595, 2007.

[16] L. McDowell, K. Gupta, and D. Aha. Cautious inference in collective classification. In *AAAI'07*, pages 596–601, 2007.

[17] J. Neville and D. Jensen. Leveraging relational autocorrelation with latent group models. In *ICDM'05*, pages 322–329, 2005.

[18] J. Neville and D. Jensen. Relational dependency networks. *JMLR*, 8:653–692, 2007.

[19] J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *KDD'03*, pages 625–630, 2003.

[20] J. Neville, D. Jensen, and B. Gallagher. Simple estimators for relational Bayesian classifers. In *ICDM'03*, pages 609–612, 2003.

[21] C. Perlich and F. Provost. Acora: Distribution-based aggregation for relational learning from identifier attributes. *MLJ*, 62(1/2):65–105, 2006.

[22] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

[23] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI'02*, pages 485–492, 2002.

[24] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *IJCAI'01*, pages 870–878, 2001.

[25] Z. Xu, K. Kersting, and V. Tresp. Multi-relational learning with gaussian processes. In *IJCAI'09*, pages 1309–1314, 2009.

[26] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML'03*, pages 912–919, 2003.