Correcting Bias in Statistical Tests for Network Classifier Evaluation

Tao Wang[1], Jennifer Neville[2], Brian Gallagher[3], and Tina Eliassi-Rad[4]

[1] Department of Computer Science,
Purdue University, West Lafayette, IN, USA
[2] Department of Computer Science and Statistics,
Purdue University, West Lafayette, IN, USA
[3] Lawrence Livermore National Laboratory, Livermore, CA, USA
[4] Department of Computer Science,
Rutgers University, Piscataway, NJ, USA

Abstract. It is difficult to directly apply conventional significance tests to compare the performance of network classification models because network data instances are not independent and identically distributed. Recent work [6] has shown that paired t-tests applied to overlapping network samples will result in unacceptably high levels (e.g., up to 50%) of Type I error (i.e., the tests lead to incorrect conclusions that models are different, when they are not). Thus, we need new strategies to accurately evaluate network classifiers. In this paper, we analyze the sources of bias (e.g. dependencies among network data instances) theoretically and propose analytical corrections to standard significance tests to reduce the Type I error rate to more acceptable levels, while maintaining reasonable levels of statistical power to detect true performance differences. We validate the effectiveness of the proposed corrections empirically on both synthetic and real networks.

Keywords: Social network analysis, Network classification.

1 Introduction

A central methodological issue in machine learning research is to compare the empirical performance of two learning algorithms and assess the *significance* of observed performance differences. Generally, to compare two classification algorithms, the available data is repeatedly partitioned (i.e., sampled) into disjoint training and test sets (e.g., using cross-validation). Then the algorithms are used to (1) learn a model from each training set, and (2) apply the learned models to the appropriate test set for prediction. Evaluation of the test set predictions (e.g., using accuracy) results in a *set* of performance measurements, one for each training/test split, for each algorithm. A hypothesis test is often used to assess whether the set of observed scores (for each of the two algorithms) are significantly different—by comparing them to the distribution of scores that would be expected if both sets were drawn from the same underlying distribution (i.e., the null hypothesis that the algorithms perform equivalently).

D. Gunopulos et al. (Eds.): ECML PKDD 2011, Part III, LNAI 6913, pp. 506–521, 2011.

Past work on methodology for accurate algorithm evaluation has mainly focused on data with independent and identically distributed (i.i.d.) instances. Dietterich [2] showed that some statistical tests, in widespread use at the time, had a high probability of Type I error due to sampling procedures that resulted in dependencies among test sets (i.e., they are likely to conclude a significant difference between algorithms when there is none). Owen [7] observed that dependencies among the hypothesis tests greatly affect the variance of the number of false discoveries in which a true null hypothesis was rejected. Other work has shown that the choice of training/test sets can lead to underestimation of variance in the cross-validation estimator of the generalization error [5,1].

However, standard approaches to algorithm evaluation become more challenging in relational learning where the data instances are not independent. In particular, two characteristics of relational learning and collective classification [8] can complicate the application of conventional statistical tests for comparing classification performance: (1) dependence between related instances leads to correlated errors and (2) network structure results in dependence between training and test set samples, which leads to correlated test sets.

Recently Neville et. al [6] conducted an empirical investigation of evaluation bias when learning from non-i.i.d. observations and proposed a novel sampling method called *network cross-validation* (NCV) that can correct for elevated levels of Type I error in network data—but at the expense of decreased statistical power (i.e., legitimate performance differences may not be detected as significant). Note that if a statistical test has biased levels of Type I error, that means many algorithms which appear to be "significantly different" may in fact have equivalent performance; if a statistical test has low statistical power, that means legitimate performance differences between algorithms may not be detected as significant.

In this paper, we consider the problem of *within-network* relational learning, where there are dependencies among data instances and the goal is transductive network learning—models are learned on a partially labeled network and then applied to *collectively* predict the class labels in the remainder of the network (i.e., the unlabeled portion). Within this setting, we demonstrate how the aforementioned network data characteristics contribute to increased Type I error in conventional statistical tests. Our analysis shows that both error correlation and overlapping samples lead to misestimation of the variance that is used in statistical tests. Based on our analysis, we propose an analytical correction to the observed variance which can be used to adjust for the bias and reduce Type I error rates, while maintaining reasonable statistical power. We demonstrate the effectiveness of the correction on both synthetic and real world data, with simulated and real classifiers. Although we evaluate the properties of the corrected significance tests for within-network classification, the findings are also applicable to other learning tasks, where evaluation is conducted with overlapping samples.

2 Network Classifier Evaluation

When comparing the empirical performance of machine learning algorithms, there are two primary methodological decisions: First, the *sampling procedure* dictates how the available data is partitioned into training and test sets for estimation of algorithm performance. Second, the *significance test* takes a set of performance measurements (e.g., accuracy) from the various sampling 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.

Sampling procedures: Given a *single*, fully labeled network S of size m, we consider two sampling procedures to generate training (labeled S_L) and test (unlabeled S_U) sets to evaluate within-network classification algorithms.

The first method is *random resampling* (RS). It involves random draws *without replacement* from the sample population (i.e., S) to generate a training/test split ($S_L \cup S_U = S; S_L \cap S_U = \emptyset$). To produce multiple training/test splits, the method samples repeatedly from the single network S, which results in overlapping test sets (i.e., $|S_{U_i} \cap S_{U_j}| \geq 0$). This method has been used extensively in past work on relational learning algorithms (see the survey in [6] for more detail).

The second method is NCV, a new sampling approach proposed by [6]. NCV samples for k disjoint test sets that will be used for *evaluation* ($S_{U_1} \cup ... \cup S_{U_k} = S; S_{U_1} \cap ... \cap S_{U_k} = \emptyset$). For each test set, the training set is selected from the complement of the network (i.e., $S_{L_i} \subseteq S - S_{U_i}$). When the target training set size is less than the size of the complement, 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, collective inference is run over the full set of unlabeled nodes (i.e., $S - S_{L_i}$), but model performance is only evaluated on the nodes assigned to the test set (S_{U_i}). Since NCV only *evaluates* model performance using disjoint test set instances, it eliminates much of the dependency due to overlapping test sets and will not suffer the same level of bias as RS [6].

Significance tests: In within-network learning, after a sampling procedure has been chosen to create training/test splits within a network, models are learned from each training set and the learned models are applied for collective inference over the appropriate test set (i.e., unlabeled portion of the network). The predictions on the test set nodes are evaluated to estimate algorithm performance (e.g., accuracy). This results in a set of performance measurements, one for each training/test split, for each algorithm. A significance test is then used to determine whether the observed performance differences are *significantly* different than would be expected if the performance measures were drawn from the same underlying distribution (i.e., the null hypothesis H_0 : the algorithms perform equivalently).

In this work, we considered both paired and unpaired t-tests for assessments of significance. We are interested in two characteristics of these tests: (1) *Type I error*: the probability of rejecting a *true* null hypothesis, and (2) *Power*: the probability of rejecting a *false* null hypothesis (i.e., 1-Type II error). Ideally

the Type I error of a significance test is equal to the chosen significance level α. If a statistical test has biased levels of Type I error (i.e., greater than the significance level α), that implies that many of the conclusions drawn from the test may be incorrect (e.g., algorithms that appear to be different may in fact have equivalent performance). In contrast, if a statistical test has low statistical power, that implies that legitimate performance differences may not be detected as significant.

3 Theoretical Analysis

Here we show theoretically how error correlation and random sampling (i.e., without replacement) from a network affects the variance of average network classification error. To do this, we model the node-level classification errors as Bernoulli random variables and analytically calculate the mean and variance of the average error over repeated samples from the same network. Specifically:

- The input population is a set of m random variables X (i.e., network size=m).
- The population consists of two types of random variables. There are pm random variables of type 1 (i.e., likely errors), which are Bernoulli distributed: $X_i^1 \sim$Bernoulli(q). There are $(1-p)m$ instances of type 0 (i.e., likely correct), which again are Bernoulli distributed: $X_i^0 \sim$Bernoulli($\frac{p}{(1-p)}(1-q)$).
- In the population, there are $|L|$ pairs of "linked" random variables that are correlated. Let ρ be the average correlation between the linked pairs $((X_i, X_j) \in L)$, otherwise we assume that the X_i are independent.
- We sample n random variables $\{X_i\}_{i=1}^n$ without replacement from the population. Since the sampling is without replacement, the random variables X_is are not independent.
- Let $Z_k = \frac{1}{n}\sum_{i=1}^n X_i$ be the average value of the r.v.'s in sample k. We are then interested in the mean and variance of the random variable Z_k, as this corresponds to the estimated error rate of algorithms that is used in statistical tests.

We note that this setup makes two primary assumptions in order to simplify the subsequent analysis. First, we assume that the variance in classification errors throughout the network, across multiple samples, can be represented by the two types of Bernoulli random variables described above. We designed the parameters of the Bernoulli variables to keep the expected value of Z_k equal to p (i.e., the average error), while allowing individual variation of the random variables across multiple samples: $E(Z_k) = E\left(\frac{1}{n}\sum_{i=1}^n X_i\right) = E\left(pX_i^1 + (1-p)X_i^0\right) = pq + (1-p)\frac{p}{(1-p)}(1-q) = p$. Note that if $q = 1$, then the random variables have exactly the same values across all samples (if selected) so this would correspond to sampling from a hypergeometric distribution with pm 1s.

Second, we consider a limited correlation structure in the above model. In particular, we assume (1) uniform correlation among all the linked nodes, and (2) independence among all unlinked nodes in the network. This is a first approximation of the assumptions typical in relational classification models, where the

parameters of directly linked nodes are tied and unlinked nodes are considered conditionally independent.

Since we have assumed independence among unlinked nodes, rather than conditional independence, the validity of our proposed model depends on whether the specified covariance matrix is positive definite. Let σ_i be the standard deviation of X_i, then the entries of the covariance matrix will be:

$$Cov(X_i, X_j) = \begin{cases} \sigma_i^2 & i = j \\ \rho \cdot \sigma_i \sigma_j & (X_i, X_j) \in L \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

In the appendix, we specify the conditions under which this matrix will be positive definite, and thus a valid covariance matrix. In practice, we find that even when the matrix is not positive definite, it is reasonable to use for the purposes of correcting evaluation bias.

Given this setup, we can now show the effect of correlation and sampling without replacement on the variance of Z_k. We state the theorems and their interpretations below and include the proofs in the appendix.

Theorem 1. ***Correlated variables increase the variance of*** Z_k

Let $\mathbf{X}$ *be a population of Bernoulli(p) random variables. Assume that a sample of* n *variables are drawn randomly from the population. Let* ρ *be the average correlation between the* X_i *that are "linked", where the probability of linkage is* $\frac{|L|}{n(n-1)}$[1]*, and assume that otherwise the* X_i *are independent. Then the variance of* Z_k *is:*

$$Var(Z_k) = \frac{1}{n} p(1-p) \left[1 + \rho \frac{|L|}{n} \right] \tag{2}$$

We refer to this variance of the average error, when there is error correlation, as $Var_{corr}(Z_k)$. Note that, other than for very specific graph structures (e.g., bipartite graphs), if relational data are correlated, autocorrelation is positive and ρ will be greater than zero. Thus, as ρ or $|L|$ (i.e., number of correlated pairs) increase, $Var_{corr}(Z_k)$ also increases.

Theorem 2. ***Sampling without replacement decreases the variance of*** Z_k

Let $\mathbf{X}$ *be a population of* m *Bernoulli random variables as described above, with* pm X^1 *variables (i.e., type 1) and* $(1-p)m$ X^0 *variables (i.e., type 0), where all the* X_i *are independent. Assume that a sample of* n *variables are drawn randomly from the population. Then the variance of* Z_k *is:*

$$Var(Z_k) = \frac{1}{n} p(1-p) \left[1 - \frac{(n-1)}{(m-1)} \left(\frac{q-p}{1-p} \right)^2 \right] \tag{3}$$

[1] Note that $n(n-1)$ is the number of possible directed edges in a network of n nodes.

We refer to this variance of the average error, when there is overlap between samples due to resampling, as $Var_{rs}(Z_k)$. Note that when $q = p$, the variables correspond to independent Bernoullis across samples and the overall variance reduces to the case when each sample is independent: $Var(Z_k) = \frac{1}{n}p(1-p)$. When $q = 1$, the random variables have exactly the same value across different samples and the variance corresponds to sampling from a Hypergeometric distribution: $Var(Z_k) = \frac{1}{n}p(1-p)\left[\frac{m-n}{m-1}\right]$.

We can now extend the results of Theorem 2, to show the joint effect of correlation and sampling without replacement on the variance of Z_k.

Theorem 3. ***Variance of Z_k with variable correlation and sampling without replacement***

Let $\mathbf{X}$ *be a population of* m *Bernoulli random variables as described above, with* pm X^1 *variables (i.e., type 1) and* $(1-p)m$ X^0 *variables (i.e., type 0). Let* ρ *be the average correlation between the* X_i *that are "linked", where the probability of linkage is* $\frac{|L|}{n(n-1)}$*, and assume otherwise the* X_i *are independent. Assume a sample of* n *variables are drawn randomly from the population. Let* $c = \sqrt{1-2p+pq}$*. Then the variance of* Z_k *is:*

$$Var(Z_k) = \frac{1}{n}p(1-p)\left[1 - \frac{(n-1)}{(m-1)}\left(\frac{q-p}{1-p}\right)^2 + \frac{|L|\rho}{n(m-1)}\left(\frac{1-q}{1-p}\right)\left[pmq - q + 2mc\sqrt{pq} + mc^2 - \frac{c^2}{(1-p)}\right]\right] \quad (4)$$

We refer to this variance of the average error, when there is both overlap between samples and error correlation, as $Var_{obs}(Z_k)$. This is the variance that is observed in networks domains when random sampling is used. Finally, we can use these results to show these two effects combine together to bias conventional statistical tests for network domains.

Theorem 4. ***Sampling without replacement and error correlation increase Type I error***

Let algorithm A and algorithm B have equal error rates of p *on network datasets drawn from the same domain* D*. Let* X_i *be the classification error for node* i *and assume that* $X_{i.A}$ *and* $X_{i.B}$ *(the error made by algorithm A and B respectively) are Bernoulli distributed as described above, i.e., with probability* p*,* $X_{i.A/B}$ *is of type 1 and with probability* $(1-p)$*,* $X_{i.A/B}$ *is of type 0. Let* ρ *be the average correlation between the* X_i, X_j *that are linked (i.e.,* $e_{ij} \in L$*) and assume that otherwise the* X_i *are independent. Assume that* k *test sets, each of size* n*, are drawn from the network of* m *nodes.*

Let $\mathbf{Z^A} = \{Z_1^A, Z_2^A, \ldots, Z_k^A\}$ *and* $\mathbf{Z^B} = \{Z_1^B, Z_2^B, \ldots, Z_k^B\}$ *be the set of average test set errors (*$Z_j = \frac{1}{n}\sum_i X_i$*) for test set* $j = [1, k]$*. Let* $c = \sqrt{1-2p+pq}$*. Then an unpaired t-test over* $\mathbf{Z^A}$ *and* $\mathbf{Z^B}$ *will underestimate the variance of the*

null distribution by: $\Delta = \frac{1}{n}p(1-p)\left[\frac{(n-1)}{(m-1)}\left(\frac{q-p}{1-p}\right)^2 + \rho\frac{|L|}{n}\left[1 - \frac{1}{(m-1)}\left(\frac{1-q}{1-p}\right)\left[pmq - q + 2mc\sqrt{pq} + mc^2 - \frac{c^2}{(1-p)}\right]\right]\right]$.

As ρ (the amount of error correlation) or q (the correlation of node error across samples) increases, the amount of underestimation (i.e., Δ) increases. This increases the probability of a Type I error in the following way. For unpaired tests, the t-statistic is: $\hat{t} = \frac{\bar{Z^A} - \bar{Z^B}}{\sqrt{Var(Z_{A/B})}\cdot\sqrt{\frac{2}{k}}}$. where $\bar{Z^A} = \frac{1}{k}\sum_j Z_j^A$ is the average of $Z_j s$ in $\mathbf{Z^A}$ (averaging the average test set errors made by algorithm A over k tet sets) , $\bar{Z^B} = \frac{1}{k}\sum_j Z_j^B$ is the average of $Z_j s$ in $\mathbf{Z^B}$, and $Var(Z_{A/B})$ is the pooled sample variance. Since $Var_{obs}(Z_k) < Var_{corr}(Z_k)$, the result will be that $\hat{t}_{obs} > \hat{t}_{corr}$ and thus $P(\hat{t}_{obs}|T) < P(\hat{t}_{corr}|T)$, where T is the appropriate t distribution with $dof = 2k-2$. Thus using $Var_{obs}(Z_k)$ instead of $Var_{corr}(Z_k)$, it is more likely that the null hypothesis will be rejected even when it holds, and as such Type I error will increase. This effect will impact paired t-tests in a similar way, as the decrease in observed variance of Z_j^A and Z_j^B will also result in an underestimate of the *difference variance* $Var(Z_j^A - Z_j^B)$, which is used instead of the pooled sample variance.

4 Analytical Correction for Bias

Based on the theoretical analysis in Section 3, we propose an analytical adjustment to correct for the bias due to repeated sampling without replacement. We would like to remove the effects of resampling, and adjust the observed variance $Var_{obs}(Z_k)$ to make it equal to the variance we would expect just due to correlation: $Var_{corr}(Z_k) = \frac{1}{n}p(1-p)[1 + \rho\frac{|L|}{n}]$. To achieve this, we simply add in the correction factor Δ from Theorem 4 above: $Var_{new}(Z_k) = Var_{obs}(Z_k) + \Delta = Var_{corr}(Z_k)$.

Correction for unpaired t-test: The correction can be used in an unpaired t-test in the following manner. We estimate model error (for each model) in the conventional manner, recording average performance over multiple test sets. After computing the variance of the average performances for a particular model (i.e., $Var_{obs}(Z_k)$), we compute the appropriate Δ from above and use it to scale the observed variance. Then the corrected variance $Var_{new}(Z_k)$ is used in place of the observed variance in the standard formulation of the unpaired t-test.

Correction for paired t-test: For the paired t-test, we can use the correction to rescale each observed value before computing the differences and variance. The idea is to compute the standardized value with the original variance (Var_{obs}) and then *unstandardize* using the corrected variance (Var_{new}). Let x^A be an observed error value for algorithm A. Let μ^A be the mean (observed) error for algorithm A. Let $\sigma_{obs}^A = (Var_{obs}^A)^{\frac{1}{2}}$ be the observed standard deviation of the average performance of algorithm A. Let $\sigma_{new}^A = (Var_{new}^A)^{\frac{1}{2}}$ be the corrected standard deviation of algorithm A. Then the adjustment for each measured

performance value x^A is the following: $x_c^A = \left[\left(\frac{x^A - \mu^A}{\sigma_{obs}^A}\right) \cdot \sigma_{new}^A\right] + \mu^A = \left(\frac{\sigma_{new}^A}{\sigma_{obs}^A}\right) x^A + \left(1 - \frac{\sigma_{new}^A}{\sigma_{obs}^A}\right) \mu^A$ The same adjustment is then applied to errors for algorithm B, with appropriate mean and variances. Once all the observed errors are adjusted, we can then compute the paired t-test in the standard way.

The correction Δ requires values for the parameters: $n, m, p, q, \rho, |L|$. We can easily calculate $n, m, |L|$ from the properties of the training/test networks used in a particular evaluation. Also, p, q, ρ can be estimated from the training/test network evaluations. For the experiments below, we use the average misclassification over all instances in a test set for p, the average misclassification for each instance across multiple test sets for q, and for ρ we use the ϕ coefficient to measure the correlation of errors for linked instances in the network (i.e., calculate ϕ coefficient from a contingency table that shows the association of prediction errors of a pair of linked instances). In the following sections we report results for paired tests only. Experiments with unpaired tests yielded qualitatively similar results.

5 Experimental Results

To investigate the effectiveness of our proposed correction with random resampling (RS-C), for significance tests of network classifiers, we conducted experiments with both simulated and real relational classifiers under varying data characteristics, using synthetic data and data from the Internet Movie Database (imdb.com).

We compare the Type I error rates and statistical power of RS, NCV, and RS-C using paired t-tests. In all the experiments, both Type I error rates and statistical power rates were averaged over 500 (simulated) or 50 (synthetic/real) trials. For a given dataset, in each trial we *sample* from the network, either using random sampling (RS) or using network cross-validation (NCV), to create 10 training/test splits (subnetworks). Then we learn classifiers (using two competing algorithms A and B) on the training subnetwork and apply the learned classifiers on its corresponding test subnetwork to measure its performance (e.g. average error rate). To compare performance, we conducted significant tests ($\alpha = 0.05$) to either accept or reject the null hypothesis that the performance of algorithm A and B are equivalent. When the experiments are designed so that two learned classifiers have equivalent error rates, any rejection of the null hypothesis corresponds to a Type I error (i.e., false positive identification of a difference when it does not exist). However, when the two classifiers perform differently, a rejection of the null hypothesis represents the *statistical power* of the test (i.e., true positive identification of a difference when it exists). We calculate and report the proportion of trials for which the null hypothesis was rejected (i.e. Type I error or power in its corresponding experimental setup).

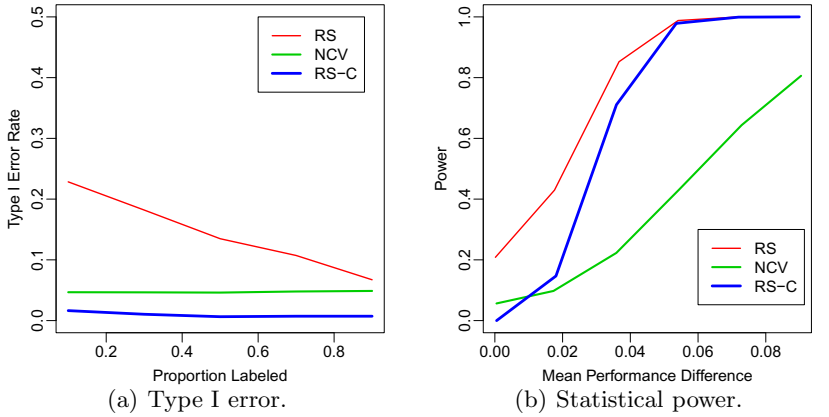
5.1 Experiments with Simulated Classifiers

Here we replicate the experiments of [6] to analyze test characteristics with simulated classifiers. We simulate the correlated errors observed in real network

classifiers by dividing data instances into disjoint groups and assigning "classification errors" such that errors are correlated among the instances within a group. We simulate two group-based classifiers A and B, ensuring that A and B have the same error rate (p) while still making different kinds of errors (i.e., A misclassifies different groups from B). Each trial utilizes datasets with default parameters $m = 300$, $p = 0.1$, and $q = 0.9$.

Figure 1(a) shows the effects of varying the proportion of labeled data for training. In these experiments, algorithms A and B have equal error rates of $p = 0.1$ so rejecting the null hypothesis corresponds to a Type I error. For RS, the Type I error rate increases as *propLabeled* decreases. This result is expected since the degree of overlap between test sets increases as the number of unlabeled instances increases. Since NCV disallows overlapping test sets by design, it is not susceptible to this problem, achieving uniformly low Type I error rates. The corrected test, RS-C, exhibits a further reduction in type I error over NCV since it accounts for error correlation as well as test set overlap.

Figure 1(b) shows the statistical power of the tests when the difference in error rates between A and B is varied (*propLabeled* $= 0.3$). In this case, since the algorithm error rates are different, a rejection of the null hypothesis corresponds to a true positive. RS has the highest statistical power overall, but when its high Type I error rates are taken into account, RS has little practical utility. RS-C, on the other hand, is able to maintain low Type I error while achieving a reasonable amount of statistical power. For example when there is a 4% difference in error rates, RS-C will be able to detect it 80% of the time. NCV has substantially lower statistical power—it will only be able to detect a 4% difference 20% of the time.



(a) Type I error. (b) Statistical power.

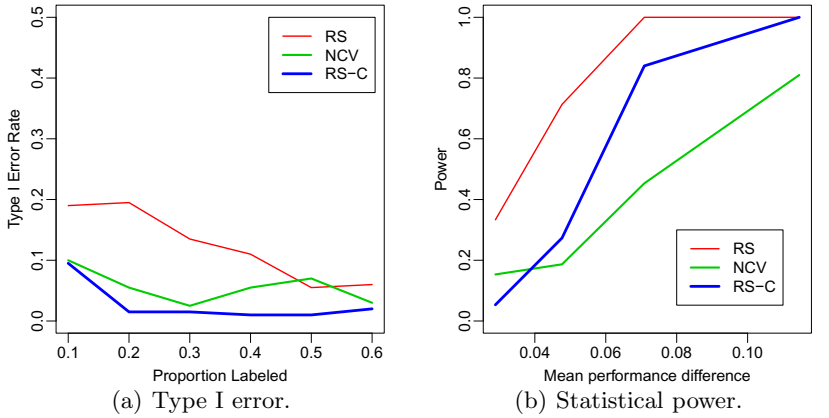
Fig. 1. Type I error and power experiments on synthetic data with simulated classifiers. (Left) Type I error as proportion of labeled data increases. (Right) Statistical power as the difference between classifiers increases.

5.2 Experiments with Real Classifiers

To further investigate RS-C, we compare the collective classification models used in [6]: weighted-vote relational neighbor (wvRN) [4] and network-only Bayes classifier (nBC) [4]. For both models, we use relaxation labeling for collective inference. To estimate Type I error, we handicap the *better* performing model (wvRN) until the performance difference between the models is negligible (i.e., ≤ 0.005). This is achieved by randomly selecting $b\%$ of the wvRN's predictions and perturbing those probabilities toward the opposite class. We searched for a value of b that resulted in a performance difference of ≤ 0.005 between the two models on a separate set of *calibration* networks. To estimate statistical power, we handicap the *worse* performing model (nBC) to increase the performance difference between the two models. We used $b = [0.025, 0.075, 0.15, 0.3]$ and measured the resulting performance difference, which is reported in Figure 2(b) and 3(b).

Results on synthetic data: In this set of experiments, we use synthetic datasets as described in [6]. The generated networks have size $m = 300$ with average autocorrelation= 0.40 and class prior $P(+) = 0.70$. The data is designed so that wvRN and nBC will make classification errors on *different* nodes. To measure Type I error rates and power of the statistical tests, we used four synthetic networks (in addition a set of 50 calibration networks).

Figure 2(a) plots the Type I error rates for three statistical tests. Notably, the level of Type I error exhibited by RS-C is significantly lower than that of RS ($> 50\%$ reduction in error). RS-C Type I error is also slightly lower than that of NCV. Figure 2(b) plots the power of each statistical test on networks with 30% labeled nodes. Here we observe, that RS-C again achieves much higher power than NCV. This is due to its use of larger test sets sizes—after correcting



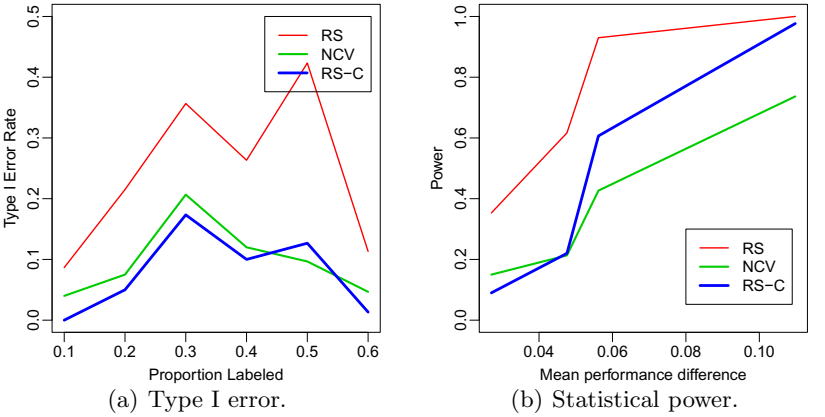
(a) Type I error. (b) Statistical power.

Fig. 2. Type I error and power experiments on synthetic data with real classifiers. (Left) Type I error as proportion of labeled data increases. (Right) Statistical power as the difference between classifiers increases.

for overlap, the *effective* sample size is still larger than the disjoint sets used in NCV. For example, on a network of 300 nodes with 30% labeled nodes, RS-C uses test set sample of 210 nodes while NCV only use a test set of 30 nodes (because of 10 cross validation). Note that the test set of 210 nodes in RS-C are not independent sample. The overlap correction will adjust its sample size downward, but the effective sample size of RS-C will still be larger than 30.

Results on real data: In the second set of experiments, we use data from the Internet Movie Database (IMDB). We collected a sample of 1,543 movies released in the United States between 2003 and 2007, with their associated producers and studios. We create six disjoint network samples using stratified sampling by studios. Within each partition, we created links among movies with a common producer. The resulting networks have an average size of 257 nodes and the movies have average degree of 16. The classification task is to predict whether the movie will make more than $\$60mil$ in total box office receipts. The average autocorrelation in these networks is 0.35.

Figure 3(a) and 3(b) show the Type I error and statistical power for each test respectively. The relative performance of the statistical tests is similar across the synthetic data and the real network data. RS-C has Type I error rates comparable to NCV and significantly lower than RS. Again RS-C has much higher power than NCV for detecting the algorithm differences in real network data.



(a) Type I error. (b) Statistical power.

Fig. 3. Type I error and power experiments on real data (IMDB) with real classifiers. (Left) Type I error as proportion of labeled data increases. (Right) Statistical power as the difference between classifiers increases.

6 Conclusion

We investigated two biases present in statistical tests for within-network classification algorithms: (1) correlated errors among related instances and (2) overlap between samples. These biases increase the Type I error to unacceptably high-levels. To adjust for these biases, we developed analytical corrections to the

empirical estimates of variance. Experiments on real and synthetic data, using real and simulated classifiers demonstrate that our corrections reduce the Type I error while maintaining good statistical power. Compared to the network cross-validation, our corrections result in a significant increase in statistical power.

Acknowledgments. We thank Yao Zhu and S.V.N. Vishwanathan for helpful discussions. This research was supported by NSF Science & Technology Center grant CCF-0939370 and NSF under contract number SES-0823313. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344 (LLNL-CONF-485754).

References

1. Bengio, Y., Grandvalet, Y.: No unbiased estimator of the variance of k-fold cross-validation. Journal of Machine Learning Research 5, 1089–1105 (2004)
2. Dietterich, T.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10, 1895–1923 (1998)
3. Franklin, J.N.: Matrix Theory. Dover Publications, Mineola (1993)
4. Macskassy, S., Provost, F.: Classification in networked data: A toolkit and a univariate case study. Journal of Machine Learning Research 8, 935–983 (2007)
5. Nadeau, C., Bengio, Y.: Inference for the generalization error. Machine Learning Journal 52(3), 239–281 (2003)
6. Neville, J., Gallagher, B., Eliassi-Rad, T., Wang, T.: Correcting evaluation bias of relational classifiers with network cross validation. Knowledge and Information Systems, 1–25 (2011)
7. Owen, A.B.: Variance of the number of false discoveries. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67, 411–426 (2005)
8. Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. AI Magazine 29(3), 93–106 (2008)

Appendix

Conditions for Covariance Matrix Validity

The covariance matrix, denoted as $\mathbf{\Sigma}$, can be specified in matrix form as:

$$\mathbf{\Sigma} := \rho(\sigma\sigma^T).* \mathbf{A} + \text{diag}(\sigma .* \sigma) \tag{5}$$

where $\sigma = [\sigma_1, \ldots, \sigma_i, \ldots, \sigma_n]$, $\mathbf{A}_{ij} = 1$ if instance i and j are linked and 0 otherwise, diag has the usual semantics, and $.*$ is the pointwise product.

To show the conditions under which the specified covariance matrix is valid, it is enough to show when $\mathbf{\Sigma}$ is positive definite.

Lemma 1. *Let $\nu_{\min}$ denote the minimum eigenvalue of matrix $\mathbf{H} = (\sigma\sigma^T).* \mathbf{A}$, and $\psi_{\min}$ denote the minimum eigenvalue of matrix $\mathbf{P} = \text{diag}(\sigma .* \sigma)$. If ρ satisfies:*

$$\rho \begin{cases} < -\frac{\psi_{\min}}{\nu_{\min}} & \text{if } \nu_{\min} > 0 \\ > -\frac{\psi_{\min}}{\nu_{\min}} & \text{if } \nu_{\min} < 0, \end{cases} \tag{6}$$

then the covariance matrix $\mathbf{\Sigma}$ defined above is positive definite.

Proof. To ensure that $\mathbf{\Sigma}$ *is positive definite it is sufficient to show that* $\lambda_{\min} > 0$*, where* $\lambda_{\min}$ *denotes the minimum eigenvalue of* $\mathbf{\Sigma}$*. By Weyl's inequality [3] we have* $\rho\nu_{\min} + \psi_{\min} \leq \lambda_{\min}$*, from which it directly follows that* $\lambda_{\min} > 0$ *whenever* (6) *is satisfied.*

Even though Lemma 1 gives admissible values of ρ to ensure that the covariance matrix is positive definite, we observe empirically that other values of ρ also yield good analytical corrections in practice. In other words, even if the covariance matrix underlying the correction is not positive definite, our adjustment method is still able to correct for the evaluation bias and correctly assess significant algorithm differences.

Proof of Theorem 1

Proof.

$$Var(Z_k) = Var\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) \tag{7}$$

$$= \frac{1}{n^2}\left(\sum_{i=1}^{n} Var(X_i) + \sum_{i=1}^{n}\sum_{j\neq i}^{n} Cov(X_i, X_j)\right) \tag{8}$$

$$= \frac{1}{n^2}\left(n \cdot p(1-p) + |L|\rho \cdot p(1-p)\right) \tag{9}$$

$$= \frac{1}{n} p(1-p)\left[1 + \rho\frac{|L|}{n}\right] \tag{10}$$

Proof of Theorem 2

Proof. First we consider the joint probability of two instances, based on sampling without replacement:

$$\begin{aligned}
&P(X_i{=}1 \wedge X_j{=}1) \\
&= P(X_i \in X^1 \wedge X_i = 1)P(X_j \in X^1 \wedge X_j = 1 | X_i \in X^1) + \\
&\quad P(X_i \in X^1 \wedge X_i = 1)P(X_j \in X^0 \wedge X_j = 1 | X_i \in X^1) + \\
&\quad P(X_i \in X^0 \wedge X_i = 1)P(X_j \in X^1 \wedge X_j = 1 | X_i \in X^0) + \\
&\quad P(X_i \in X^0 \wedge X_i = 1)P(X_j \in X^0 \wedge X_j = 1 | X_i \in X^0) \qquad (11) \\
&= \left[\left(\frac{pm}{m}q\right)\left(\frac{pm-1}{m-1}q\right)\right] + \\
&\left[\left(\frac{pm}{m}q\right)\left(\frac{(1-p)m}{m-1}\frac{p}{1-p}(1-q)\right)\right] + \\
&\left[\left(\frac{(1-p)m}{m}\frac{p}{1-p}(1-q)\right)\left(\frac{pm}{m-1}q\right)\right] + \\
&\left[\left(\frac{(1-p)m}{m}\frac{p}{1-p}(1-q)\right)\left(\frac{(1-p)m-1}{m-1}\frac{p}{1-p}(1-q)\right)\right] \qquad (12)
\end{aligned}$$

$$= \frac{p}{(m-1)}\left[pm - q^2 - \frac{p(1-q)^2}{(1-p)}\right] \tag{13}$$

Now consider the covariance of two instances, based on sampling without replacement:

$$\begin{aligned} &Cov(X_i, X_j) \\ &= E(X_i X_j) - E(X_i)E(X_j) && (14)\\ &= P(X_i = 1 \wedge X_j = 1) - p \cdot p && (15)\\ &= \frac{p}{(m-1)}\left[pm - q^2 - \frac{p(1-q)^2}{(1-p)}\right] - p^2 && (16)\\ &= -\frac{p(1-p)}{(m-1)}\left[\frac{(q-p)^2}{(1-p)^2}\right] && (17) \end{aligned}$$

With the covariance, we can compute the overall variance based on sampling without replacement:

$$\begin{aligned} Var(Z_k) &= Var\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) && (18)\\ &= \frac{1}{n^2}\left[\sum_{i=1}^{n} Var\left(X_i\right) + \sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} Cov(X_i, X_j)\right] && (19)\\ &= \frac{1}{n}\left[p(1-p) - (n-1)\frac{p(1-p)}{(m-1)}\left[\frac{(q-p)^2}{(1-p)^2}\right]\right] && (20)\\ &= \frac{1}{n}p(1-p)\left[1 - \frac{(n-1)}{(m-1)}\left(\frac{q-p}{1-p}\right)^2\right] && (21) \end{aligned}$$

Proof of Theorem 3

Proof. To combine the covariance based on error correlation with the covariance based on overlap, we need to determine the effect of the correlation on the conditional probability of a linked instance, i.e., $P(X_j = 1 | X_i = 1, e_{ij} \in L)$. We can derive this from the relationship between correlation and covariance:

$$Cov(X_i, X_j | e_{ij} \in L) = Corr(X_i, X_j | e_{ij}) Var(X_i)^{\frac{1}{2}} Var(X_j)^{\frac{1}{2}} \tag{22}$$

$$E(X_i X_j | e_{ij} \in L) - E(X_i)E(X_j) = \rho \cdot Var(X_i)^{\frac{1}{2}} Var(X_j)^{\frac{1}{2}} \tag{23}$$

$$P(X_j | X_i, e_{ij} \in L) = E(X_j) + \frac{\rho \cdot Var(X_i)^{\frac{1}{2}} Var(X_j)^{\frac{1}{2}}}{E(X_i)} \tag{24}$$

We can then enumerate the conditional probabilities for each of the four possible worlds for (X_i, X_j):

$$P(X_j^1 | X_i^1) = E(X_j^1) + \frac{\rho Var(X_i^1)^{\frac{1}{2}} Var(X_j^1)^{\frac{1}{2}}}{E(X_i^1)} = q + \rho(1-q) \tag{25}$$

$$P(X_j^0|X_i^1) = E(X_j^0) + \frac{\rho Var(X_i^1)^{\frac{1}{2}} Var(X_j^0)^{\frac{1}{2}}}{E(X_i^1)} \tag{26}$$

$$= \frac{p(1-q)}{1-p} + \rho\frac{(1-q)}{(1-p)}\sqrt{\frac{p(1-2p+pq)}{q}} \tag{27}$$

$$P(X_j^1|X_i^0) = E(X_j^1) + \frac{\rho Var(X_i^0)^{\frac{1}{2}} Var(X_j^1)^{\frac{1}{2}}}{E(X_i^0)} \tag{28}$$

$$= q + \rho\sqrt{\frac{q(1-2p+pq)}{p}} \tag{29}$$

$$P(X_j^0|X_i^0) = E(X_j^0) + \frac{\rho Var(X_i^0)^{\frac{1}{2}} Var(X_j^0)^{\frac{1}{2}}}{E(X_i^0)} \tag{30}$$

$$= \frac{p(1-q)}{1-p} + \rho\left(\frac{1-2p+pq}{1-p}\right) \tag{31}$$

Now we can incorporate these conditional probabilities into the calculation of $P(X_i, X_j)$ and $Cov(X_i, X_j)$, incorporating both correlation and sampling without replacement. Let $c = \sqrt{1-2p+pq}$, then:

$$P(X_i = 1 \wedge X_j = 1) \tag{32}$$

$$= \left[\left(\frac{pm}{m}q\right)\left(\frac{pm-1}{m-1}[q + \frac{|L|}{n(n-1)}\rho(1-q)]\right)\right] +$$
$$\left[\left(\frac{pm}{m}q\right)\left(\frac{(1-p)m}{m-1}\left[\frac{p}{1-p}(1-q) + \frac{|L|}{n(n-1)}\rho\frac{(1-q)}{(1-p)}c\sqrt{\frac{p}{q}}\right]\right)\right] +$$
$$\left[\left(\frac{(1-p)m}{m}\frac{p(1-q)}{1-p}\right)\left(\frac{pm}{m-1}\left[q + \frac{|L|}{n(n-1)}\rho c\sqrt{\frac{q}{p}}\right]\right)\right] +$$
$$\left[\left(\frac{(1-p)m}{m}\frac{p(1-q)}{1-p}\right)\left(\frac{(1-p)m-1}{m-1}\left[\frac{p(1-q)}{1-p} + \frac{|L|}{n(n-1)}\rho\left(\frac{c^2}{1-p}\right)\right]\right)\right] \tag{33}$$

$$= \frac{p}{(m-1)}\left[pm - q^2 - \frac{p(1-q)^2}{(1-p)}\right] +$$
$$\frac{|L|}{n(n-1)}\left(\frac{pq(1-q)\rho}{m-1}\right)\left[pm - 1 + 2mc\sqrt{\frac{p}{q}} + \frac{mc^2}{q} - \frac{c^2}{q(1-p)}\right] \tag{34}$$

$$Cov(X_i, X_j) = E(X_i X_j) - E(X_i)E(X_j) \tag{35}$$

$$= P(X_i = 1 \wedge X_j = 1) - p \cdot p \tag{36}$$

$$= \frac{p}{(m-1)}\left[pm - q^2 - \frac{p(1-q)^2}{(1-p)}\right] - p^2 +$$
$$\frac{|L|}{n(n-1)}\left(\frac{pq(1-q)\rho}{m-1}\right)\left[pm - 1 + 2mc\sqrt{\frac{p}{q}} + \frac{mc^2}{q} - \frac{c^2}{q(1-p)}\right] \tag{37}$$

$$= \frac{p(1-p)}{(m-1)}\left[-\left(\frac{q-p}{1-p}\right)^2 + \frac{|L|\rho}{n(n-1)}\left(\frac{1-q}{1-p}[pmq - q + 2mc\sqrt{pq} + mc^2 - \frac{c^2}{(1-p)}\right]\right]$$

Now we can compute the overall variance of Z_k, with correlation as well as sampling without replacement:

$$Var(Z_k) = Var\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{1}{n^2}\left[\sum_{i=1}^{n} Var\left(X_i\right) + \sum_{i=1}^{n}\sum_{j=1,j\neq i}^{n} Cov(X_i, X_j)\right] \quad (38)$$

$$= \frac{1}{n}\left[p(1-p) - \frac{n(n-1)}{n}\frac{p(1-p)}{(m-1)}\left[\left(\frac{q-p}{1-p}\right)^2 - \right.\right.$$

$$\left.\left.\frac{|L|\rho}{n(n-1)}\left(\frac{1-q}{1-p}\right)\left[pmq - q + 2mc\sqrt{pq} + mc^2 - \frac{c^2}{(1-p)}\right]\right]\right] \quad (39)$$

$$= \frac{1}{n}p(1-p)\left[1 - \frac{(n-1)}{(m-1)}\left(\frac{q-p}{1-p}\right)^2 + \right.$$

$$\left.\frac{|L|\rho}{n(m-1)}\left(\frac{1-q}{1-p}\right)\left[pmq - q + 2mc\sqrt{pq} + mc^2 - \frac{c^2}{(1-p)}\right]\right] \quad (40)$$

Proof of Theorem 4

Proof. The unpaired t-test uses the average (i.e., *pooled*) variance of Z^A and Z^B for the null distribution. Since the error distribution of A and B are equal, the average is equal to the variance of a single algorithm. When the nodes are repeatedly sampled without replacement, we know from Theorem 3 that the observed variance of Z_k will be the following: $Var_{obs}(Z_k) = \frac{1}{n}p(1-p)\left[1 - \frac{(n-1)}{(m-1)}\left(\frac{q-p}{1-p}\right)^2 + \frac{|L|\rho}{n(m-1)}\left(\frac{1-q}{1-p}\right)\left[pmq - q\ + 2mc\sqrt{pq} + mc^2 - \frac{c^2}{(1-p)}\right]\right.$, where $c = \sqrt{1-2p+pq}$. However, when there is error correlation ρ among the instances in the data, from Theorem 1 we know that the variance of Z_k with independent samples is the following: $Var_{corr}(Z_k) = \frac{1}{n}p(1-p)\left[1 + \rho\frac{|L|}{n}\right]$. Since the t-test assumes independent samples, the variance of the null distribution should correspond to the variance without repeated sampling $Var_{corr}(Z_k)$. If the observed variance $Var_{obs}(Z_k)$ is used in the t-test, it will result in an underestimate of Δ:

$$\Delta = Var_{corr}(Z_k) - Var_{obs}(Z_k) \quad (41)$$

$$= \frac{1}{n}p(1-p)\left[1 + \rho\frac{|L|}{n}\right] - \frac{1}{n}p(1-p)\left[1 - \frac{(n-1)}{(m-1)}\left(\frac{q-p}{1-p}\right)^2\right] +$$

$$\left[\frac{|L|\rho}{n(m-1)}\left(\frac{1-q}{1-p}\right)\left[pmq - q + 2mc\sqrt{pq} + mc^2 - \frac{c^2}{(1-p)}\right]\right] \quad (42)$$

$$= \frac{1}{n}p(1-p)\left[\frac{(n-1)}{(m-1)}\left(\frac{q-p}{1-p}\right)^2 + \right.$$

$$\left.\rho\frac{|L|}{n}\left[1 - \frac{1}{(m-1)}\left(\frac{1-q}{1-p}\right)\left[pmq - q + 2mc\sqrt{pq} + mc^2 - \frac{c^2}{(1-p)}\right]\right]\right] \quad (43)$$