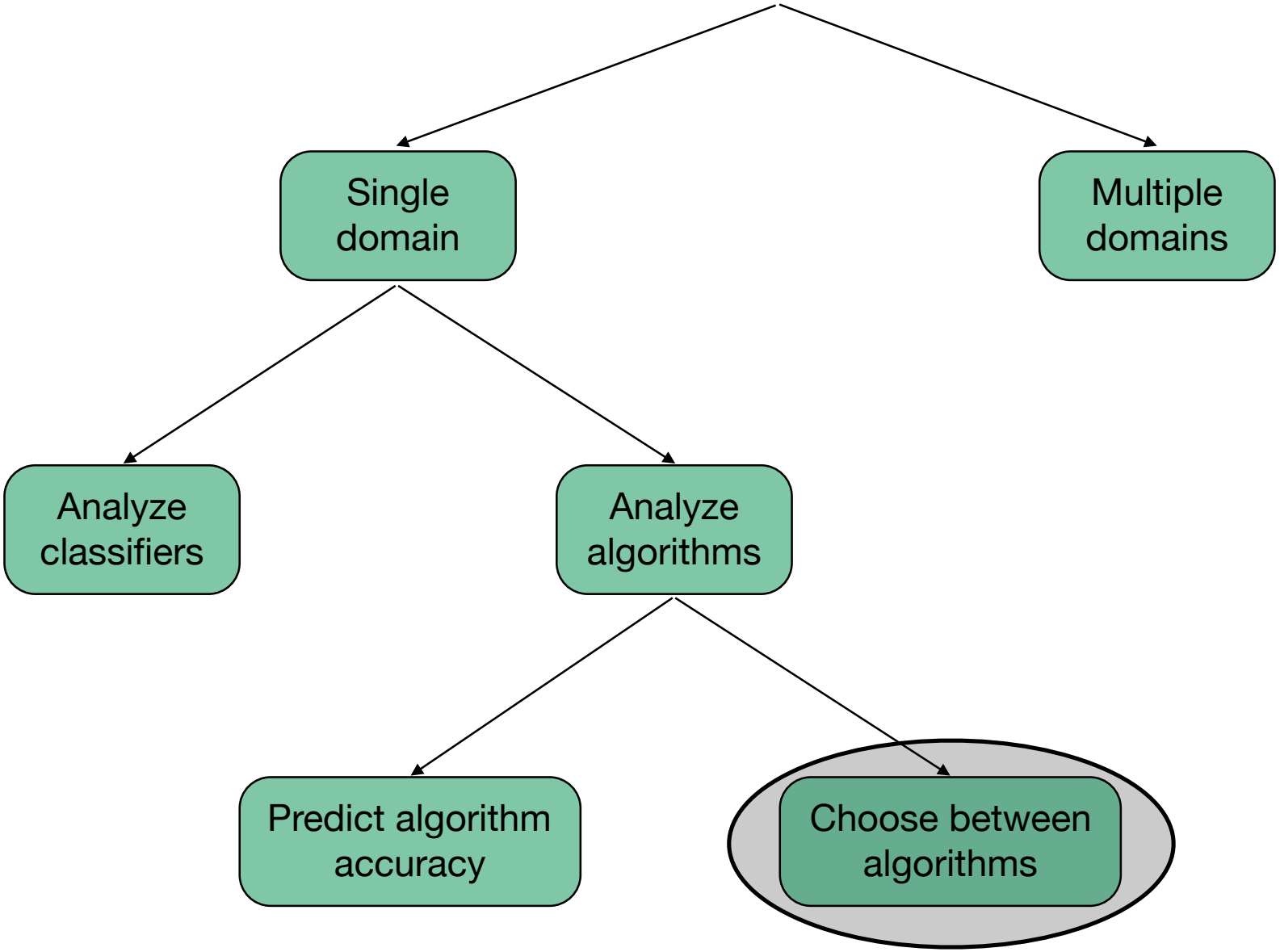
Evaluating Statistical Tests for Within-Network Classifiers of Relational Data

Jennifer Neville, *Purdue University*
Brian Gallagher and Tina Eliassi-Rad, *Lawrence Livermore National Laboratory*

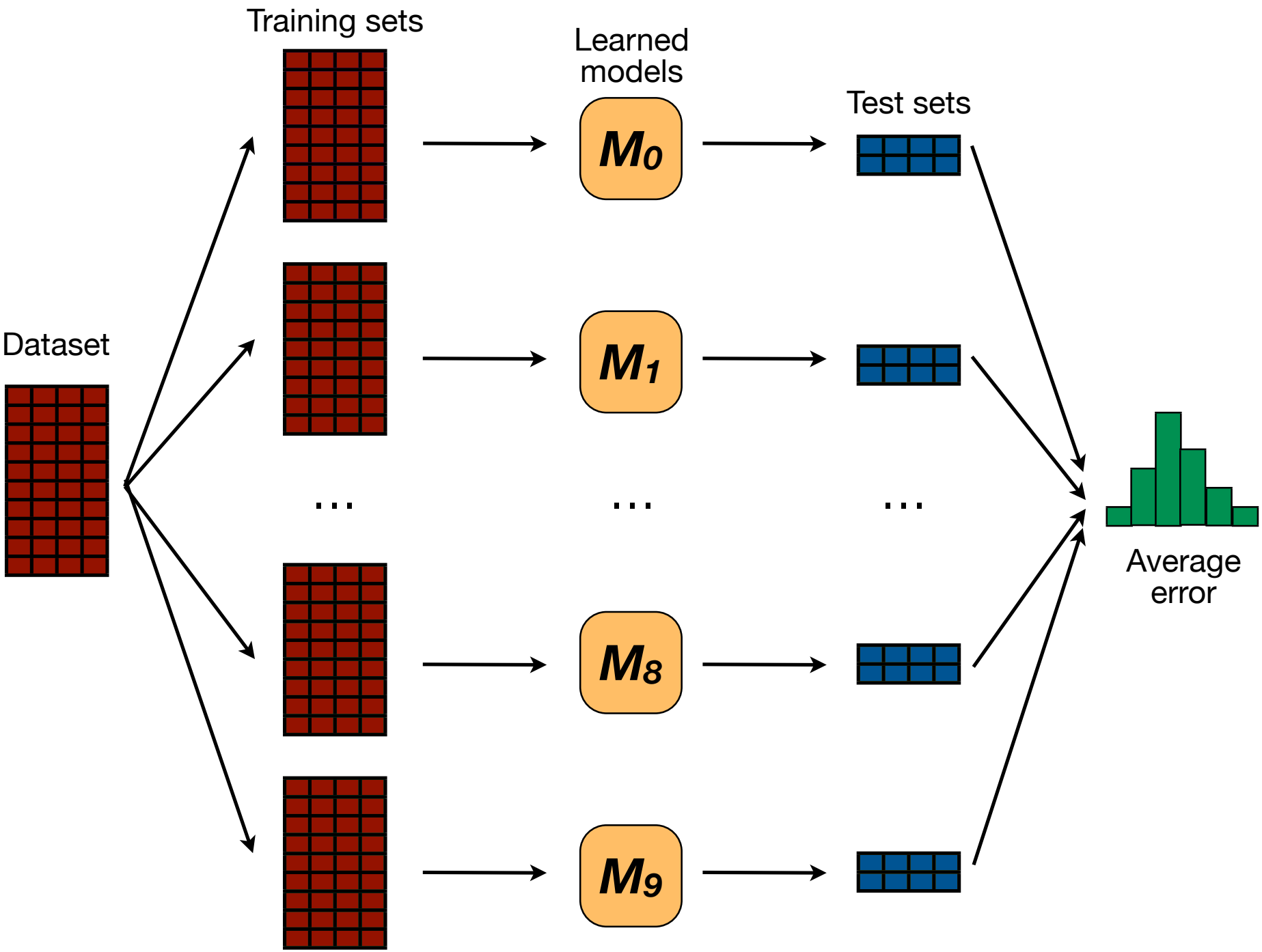
ICDM 2009

Statistical questions in machine learning *(Dietterich '98)*



Given two learning algorithms A and B and a dataset of size S from a domain D...

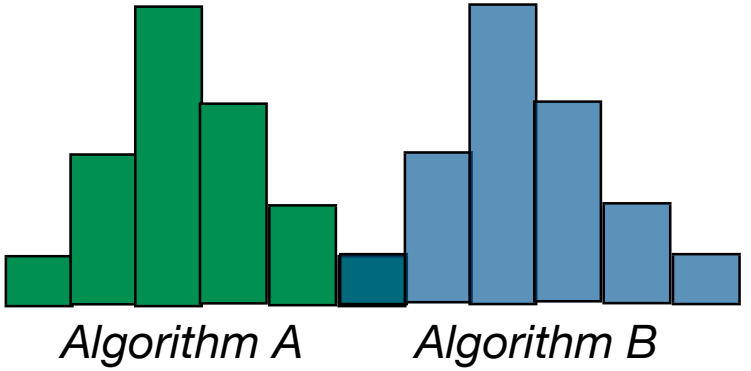
which ***algorithm*** *will produce more accurate classifiers when* ***trained*** *on other* ***datasets*** *of size S drawn from D?*


Training sets
Learned models
Test sets
Dataset
M_0
M_1
M_8
M_9
Average error

Comparison of algorithm performance

- Is observed performance difference **significantly more** than would be expected by random chance?

- Hypothesis testing
  - Use two-sample **t-test**
  - Null hypothesis (H_0): Algorithm performance rates are drawn from the same distribution



- Types of errors:
  - Type I error: Reject the null hypothesis when it is true (false positive)
  - Type II error: Accept the null when it is false (false negative)

Our findings

- We show that commonly used statistical tests can result in **unacceptably high levels of Type I error** for network classifiers
  - This means that many algorithm differences will be judged incorrectly as significant when in fact performance is equivalent
- Broad set of empirical experiments validate findings
  - Synthetic data, simulated classifiers
  - Synthetic data, real classifiers
  - Real data, real classifiers
- Proposed solution: **Network cross-validation**
  - Lowers probability of Type I error (but at expense of decreased power)

Our focus:

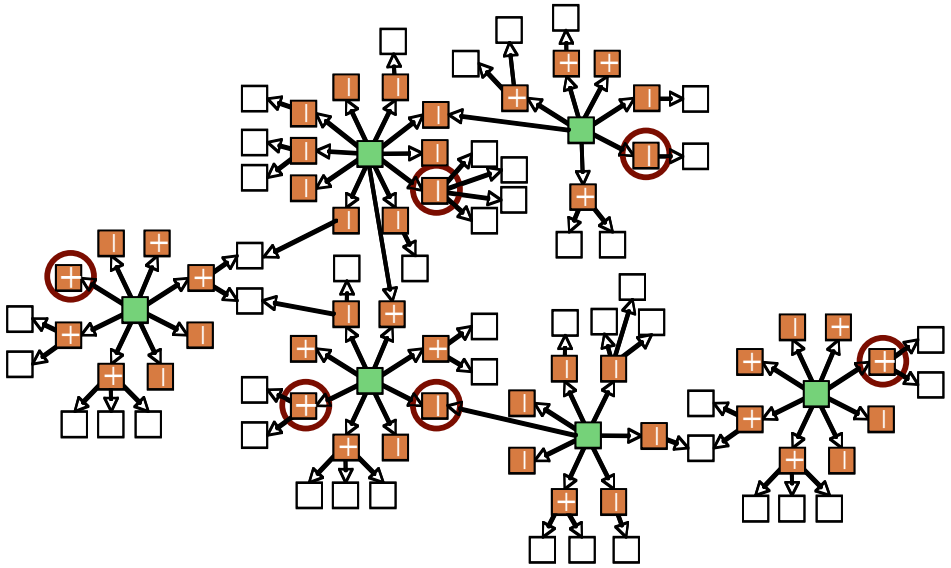
Comparing within-network relational-learning algorithms

Given two learning algorithms A and B
and a ***partially-labeled network***
from a domain D with ***S_L labeled*** *instances*
and ***S_U unlabeled*** *instances ($S = S_L + S_U$)...*

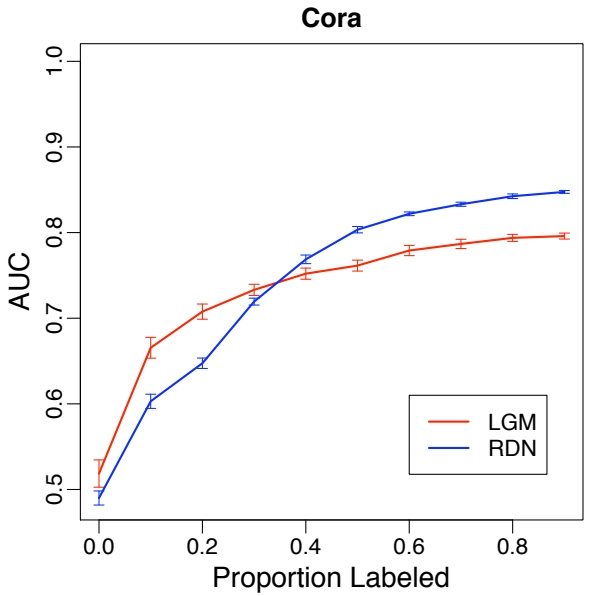
which algorithm will produce more
accurate classifiers when trained on other
partially-labeled networks *of size S from D?*

Within-network learning/inference

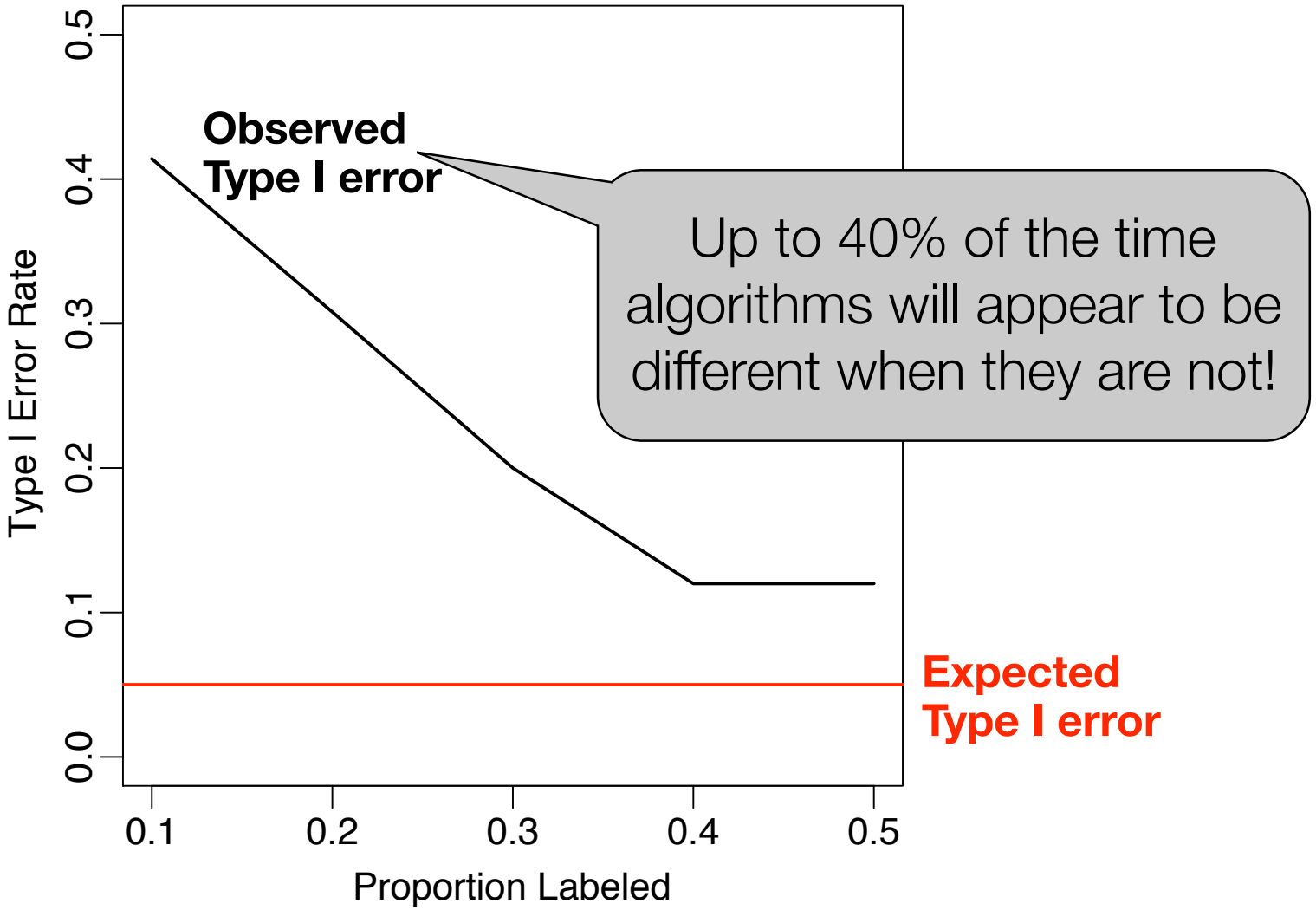
- Algorithms learn models from a partially-labeled network
- Models are then applied to predict the class labels in the remainder of the network (i.e., the unlabeled nodes)



- Typical evaluation approach:
  - Vary proportion of labeled nodes
  - Randomly vary label set to estimate performance
  - Use paired t-test to assess significance

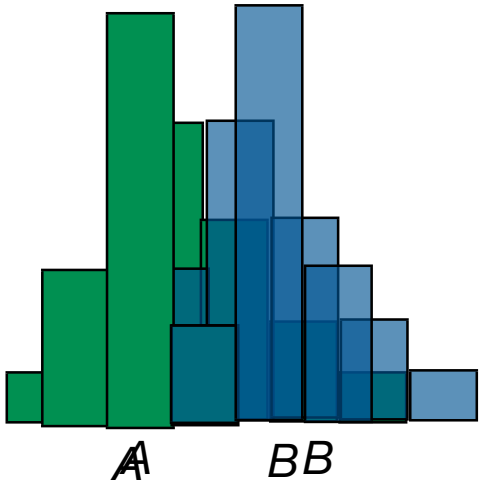


Evaluation of paired t-test on network data



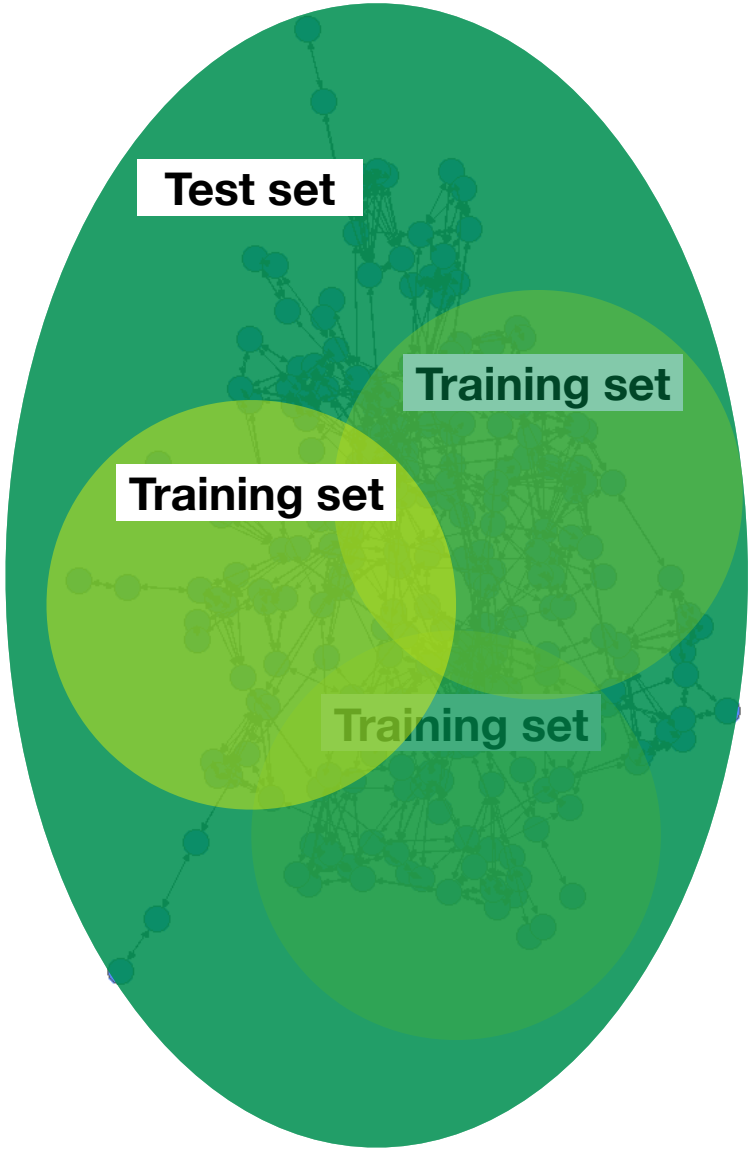
Why are conventional statistical tests biased?

- T-test results are **biased** if performance is estimated from **overlapping** test sets *(Dietterich'98)*
  - Overlapping samples leads to underestimation of variance... which increases the probability of Type I error
  - **Recommendation**: Use cross-validation to eliminate dependencies between test sets



How do we sample for
within-network classification?

Network sampling

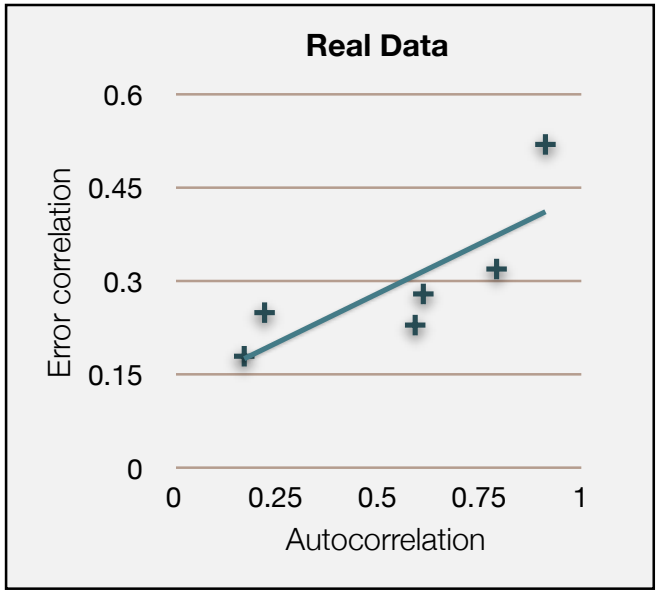


Typical approach

Use **repeated random sampling** to create multiple training/test (labeled/unlabeled) splits

Bias due to network characteristics

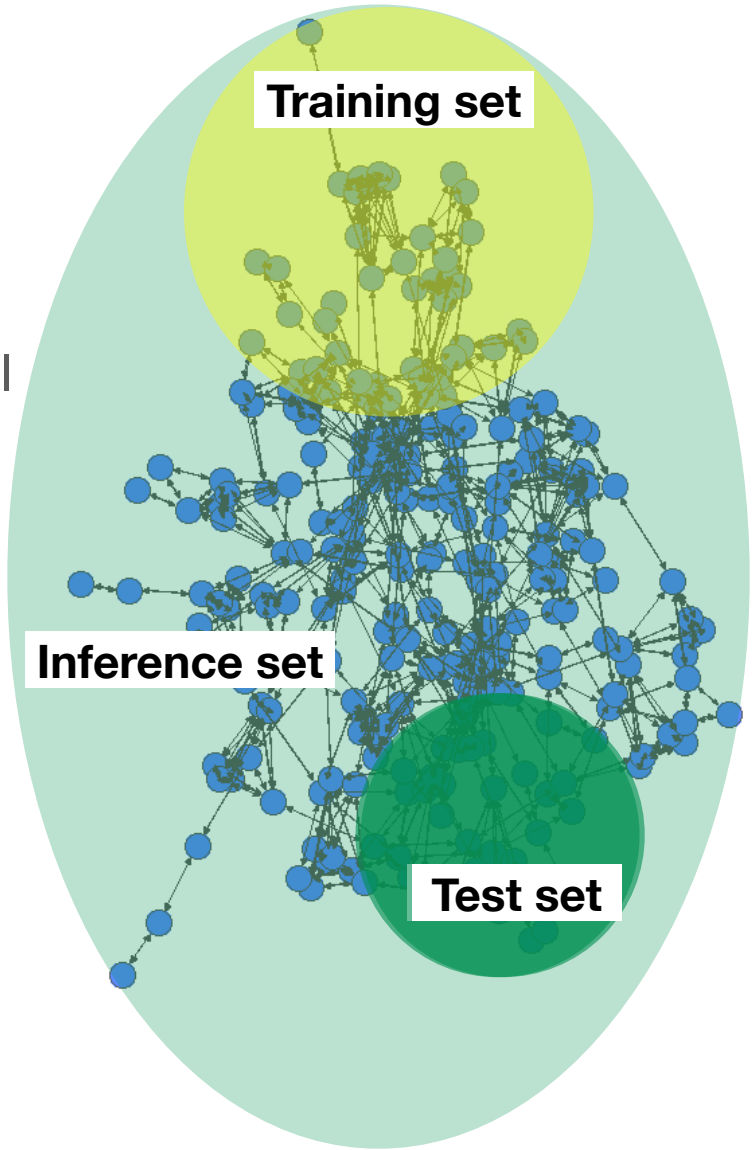
- **Training and test set sizes are dependent**
  - As the proportion of labeled data decreases, the size of the test set increases
  - As the size of the test (unlabeled) set increases, the overlap between test sets increase, which leads to increased Type I error
- **Network instances are not independent**
  - Dependencies among instances leads to correlated errors
  - Correlated error increases the variance of observed performance, which leads to increased Type I error



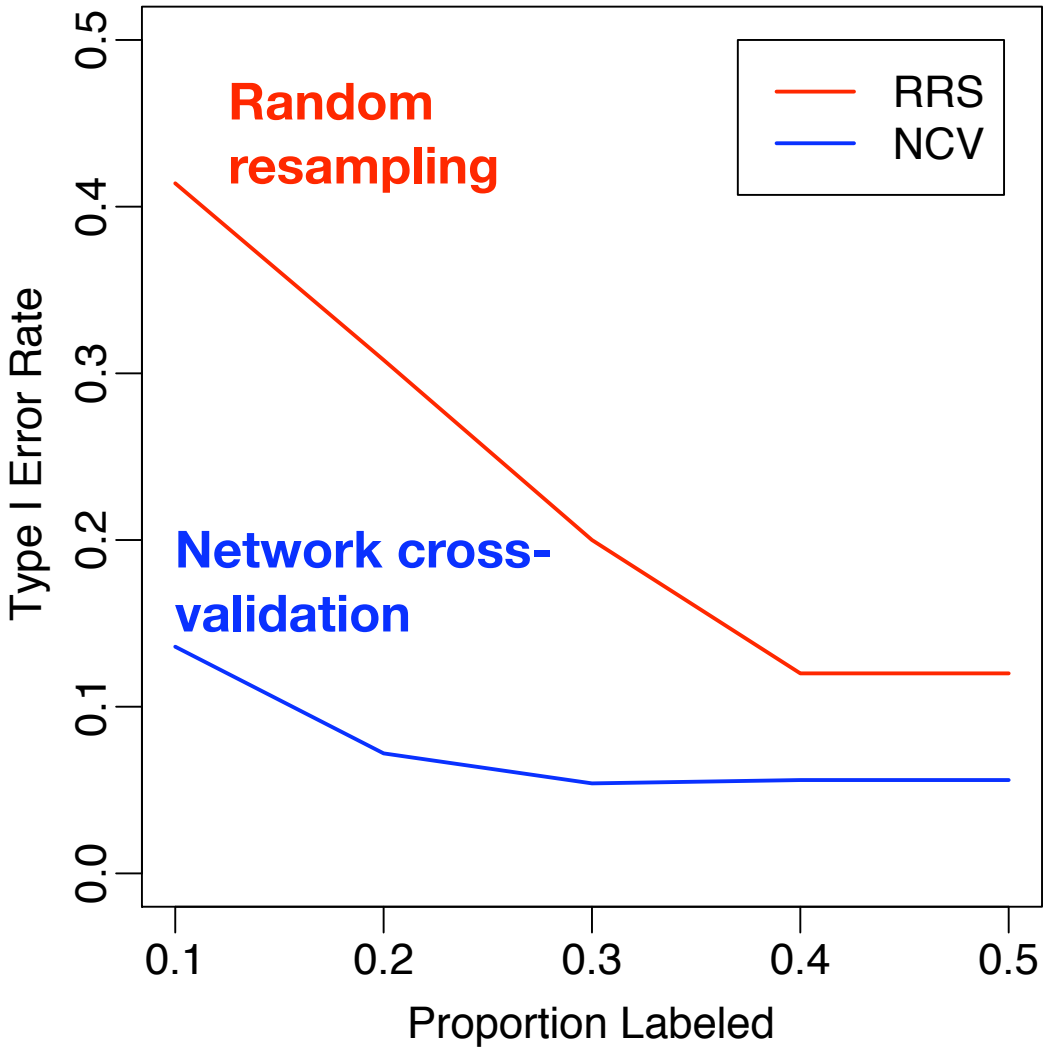
Can we use cross-validation
for within-network classification?

Network cross-validation

- Use k-fold cross-validation to select disjoint **test sets** of size N/k
- From remaining N(k-1)/k of data randomly select labeled **training set** of appropriate size (e.g., for p% labeled, select p·N instances to label as the training set)
- Add all unlabeled instances to the **inference set** (e.g., network = training set + inference set)
  - Run collective inference over entire inference set to make predictions
  - But only evaluate accuracy of predictions on test set



NCV reduces Type I error (on real network data)

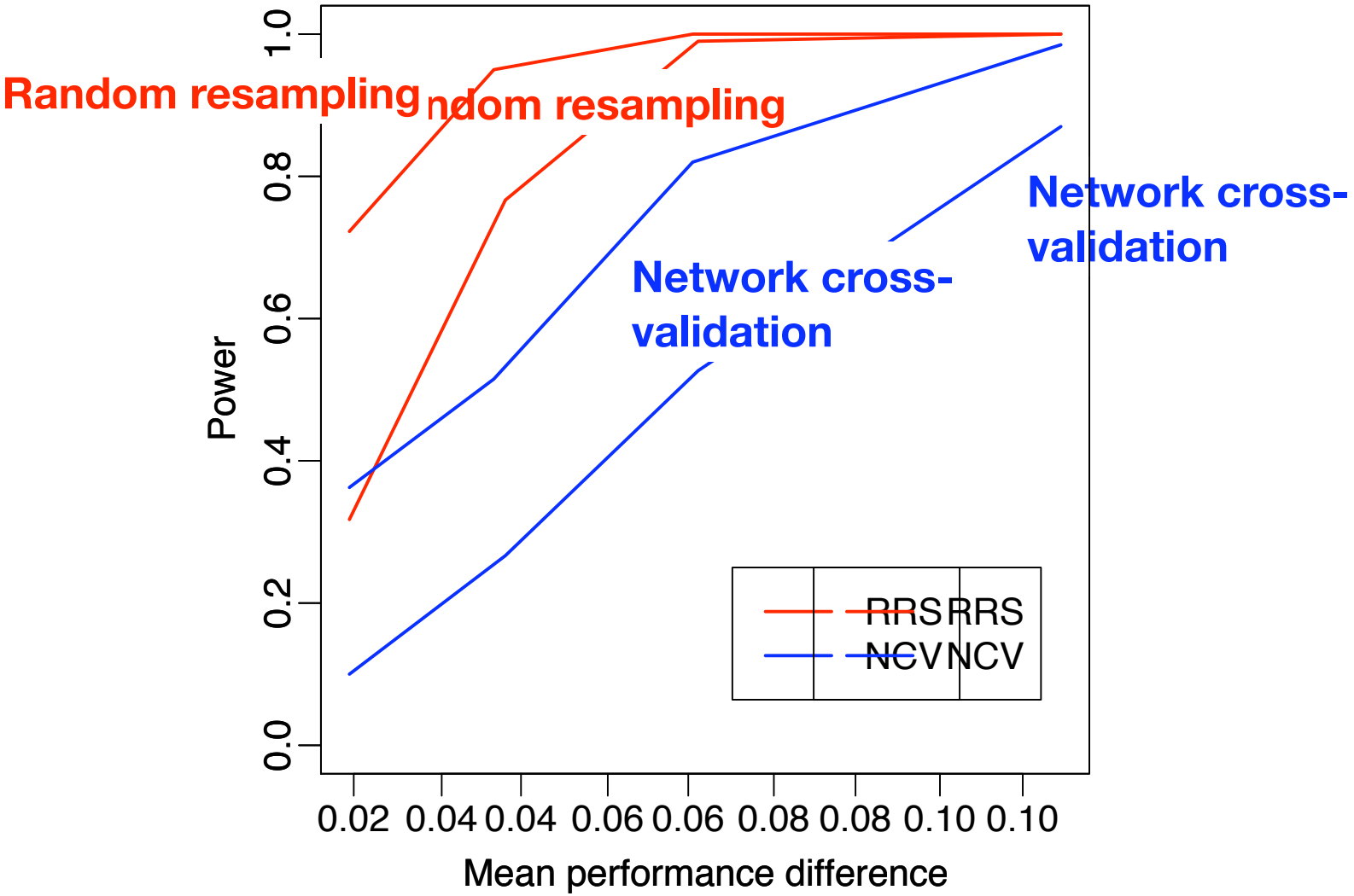


Data: AdHealth dataset, six middle- and high-school social networks

Task: Predict whether a student smokes or not

Models: Compare wvRN and nBC algorithms
(Macskassy, Provost JMLR'07)

NCV results in decreased statistical power



Applicability of results

- High Type I error indicates that many algorithm differences will be judged incorrectly as significant when in fact performance is equivalent
  - These findings apply to much of the recent work in relational learning (see paper for detailed survey)
- The bias will also affect:
  - More complex relational models -- since any relational model that attempts to exploit relational autocorrelation is likely to produce correlated errors
  - Across-network tasks -- if evaluation is on partially-labeled networks
  - Other forms of hypothesis testing (standard error will be underestimated)
- The extent of the bias will depend on:
  - Level of error correlation in network
  - Amount of overlap between samples

} See paper for simulation results

Conclusion

- Our analysis shows that a commonly-used form of evaluation in relational learning can result in unacceptably high levels of Type I error (e.g., 40-50%)
- Network cross-validation produces more acceptable levels of Type I error while still providing reasonable levels of statistical power
- Current work:
  - Theoretical proof of variance underestimation
  - Analytical adjustment for bias in t-test

Questions?

neville@cs.purdue.edu
bgallagher@llnl.gov
eliassirad1@llnl.gov

Backup slides

Autocorrelation and error correlation

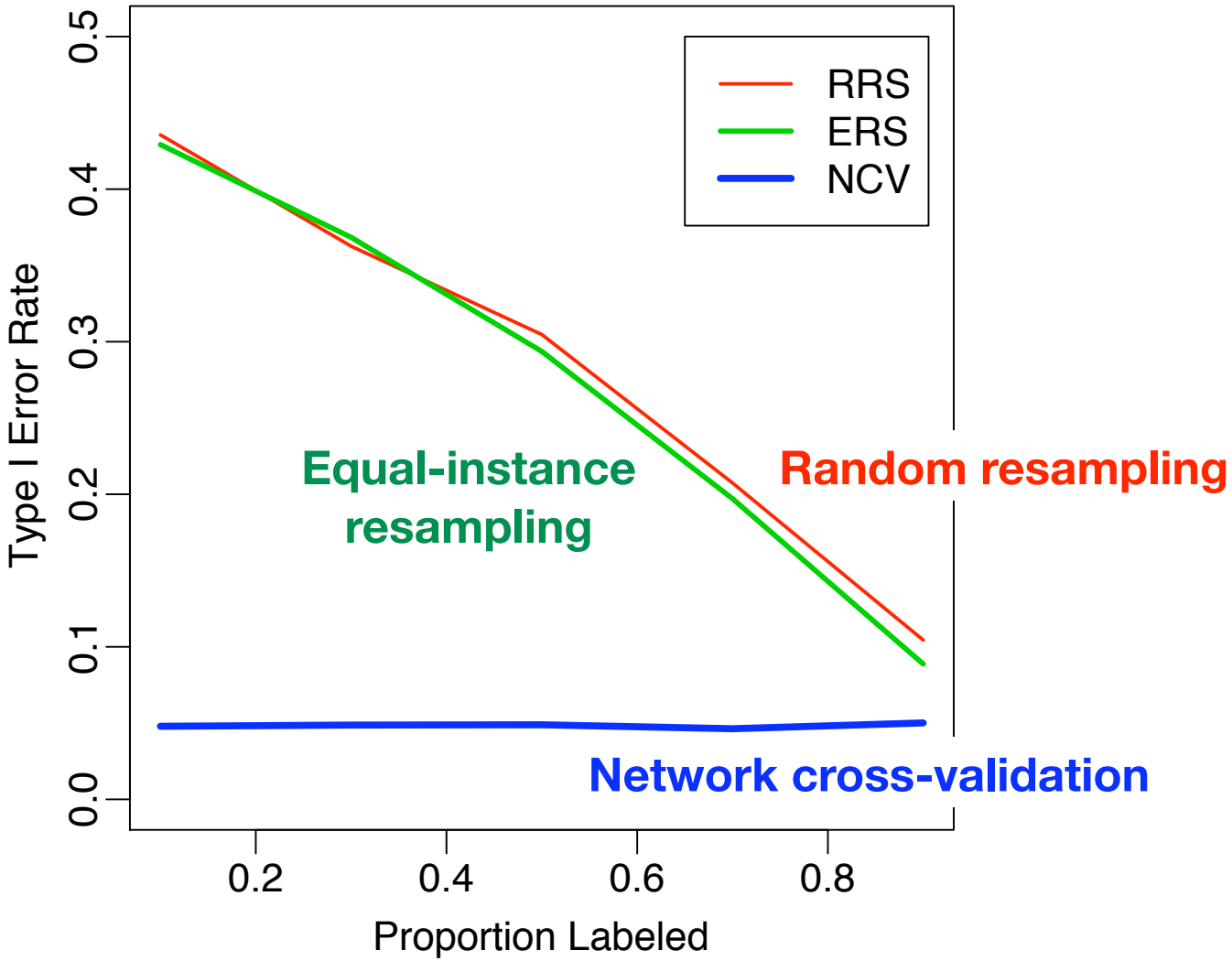
Data Set	Task	Error Corr.	Autocorr.
Enron Email	Executive?	0.18	0.17
Citeseer	Neural Nets?	0.23	0.59
Political Books	Neutral?	0.25	0.22
Cora	Info. Retrieval?	0.28	0.61
Reality Mining	In Study?	0.32	0.79
Reality Mining	Student?	0.52	0.91

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

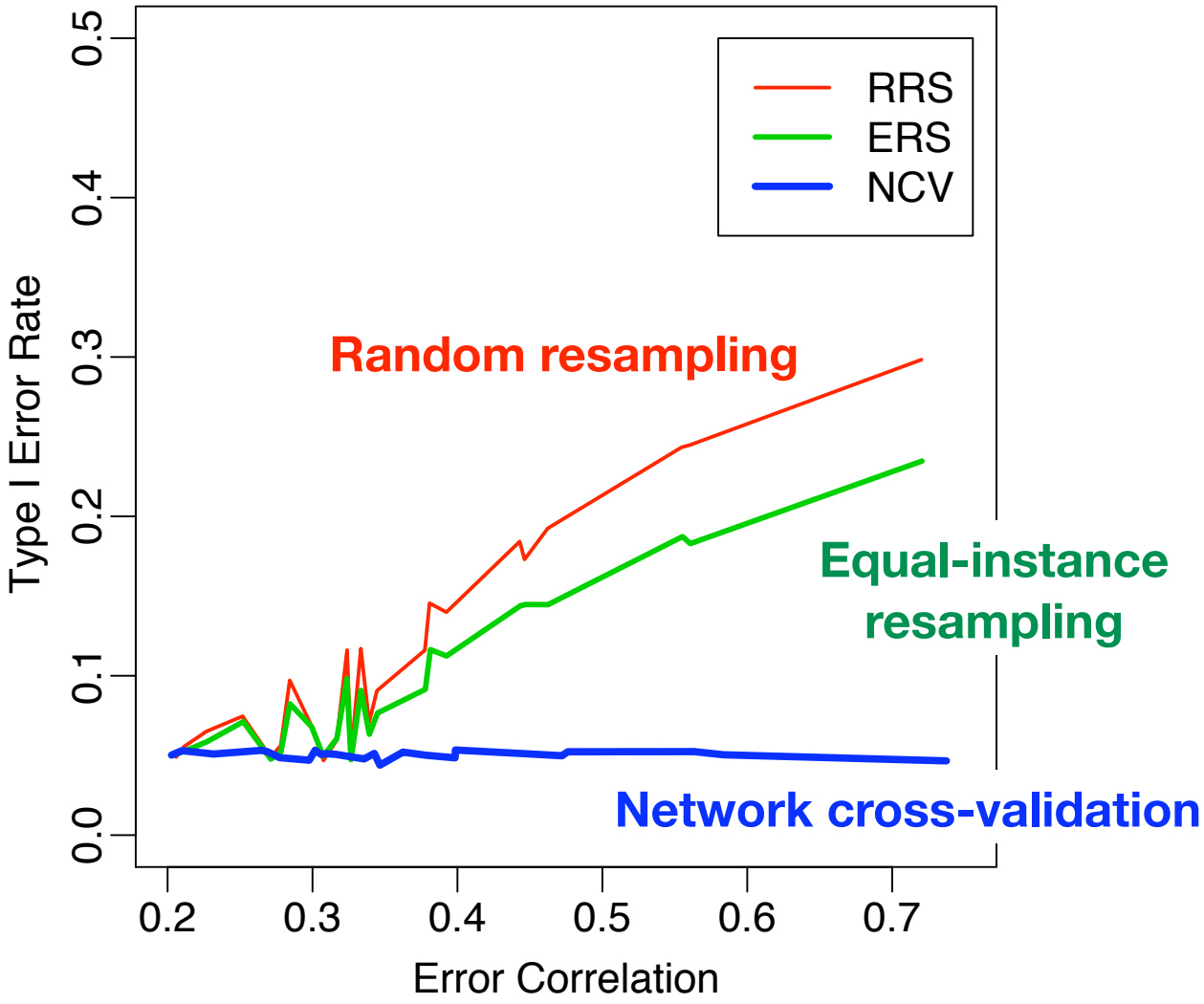
Methodology

- Synthetic data
  - Partition data instances into groups; errors within each group are correlated
  - Dataset size = 300
    Number of groups = 10
- Simulated classifiers
  - Create two classifiers with equal overall error rate
  - Associate algorithm errors with different groups in data
- Experiments
  - 1000 trials
  - 10 simulations of randomly choosing set of labeled “training” set and unlabeled “test” set
  - Report average Type I error
    Note: algorithms are equivalent so any assessment of significance is considered a Type I error
  - Vary error correlation and label proportion

Type I error increases as test set size increases



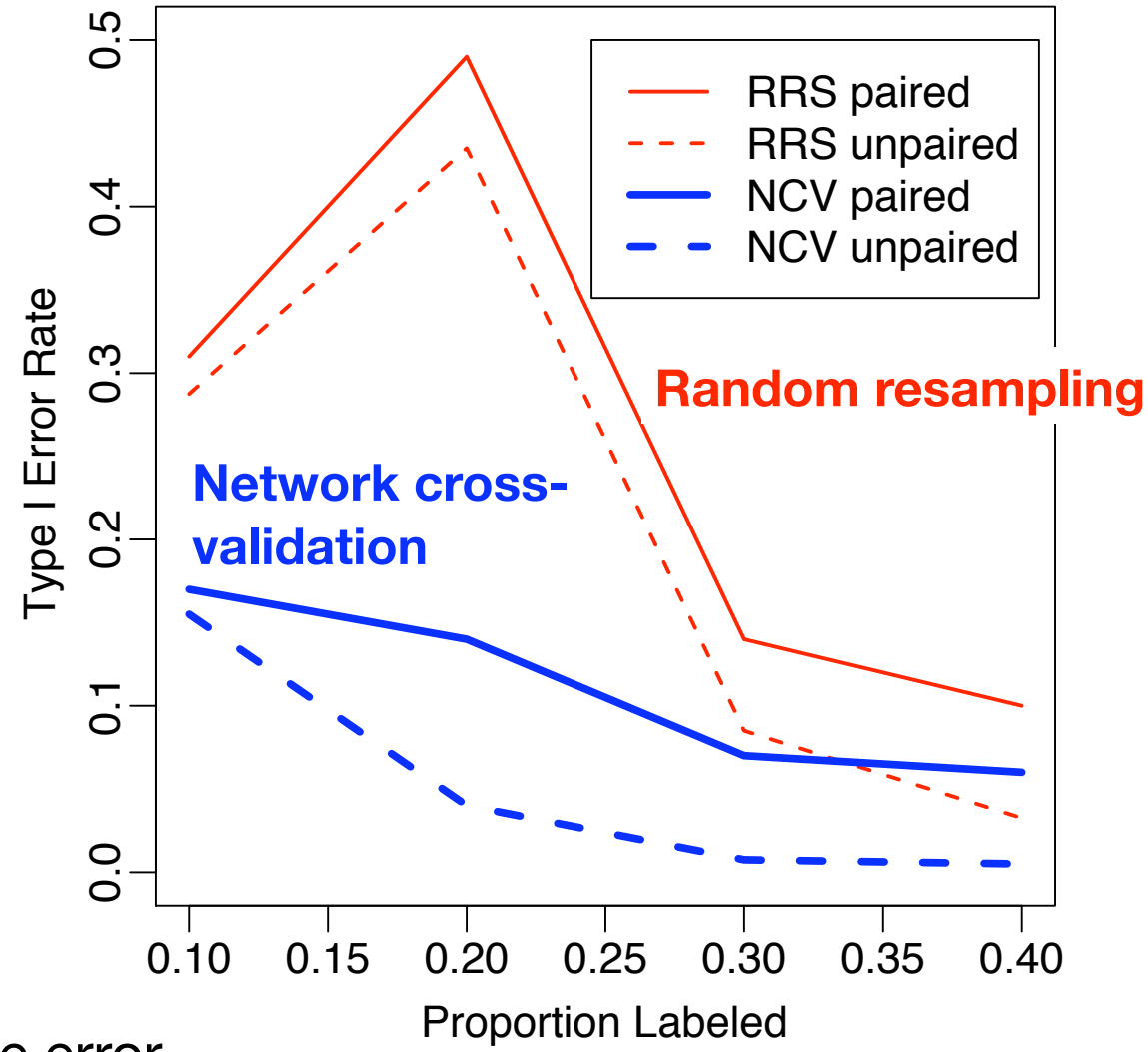
Type I error increases as error correlation increases



Methodology

- Latent group model to generate multiple networks from the same domain *D*
  - Dataset size = 300
    Average group size = 10
  - Two types of groups A/B:
    A: higher intra-group linkage, positive class labels
    B: lower intra-group linkage, negative class labels
  - Goal: Classifiers will make different types of errors
- Real classifiers
  - wvRN: no learning, just assume autocorrelation exists and infer label as average of neighbor labels
  - nBC: learn CPD to predict class label based on neighbor labels, use collective inference to propagate inference
- Equalize classifiers
  - Perturb predictions of wvRN until performance is within 0.5% over 500 calibration sets

Type I error is as high as 50% for standard RRS



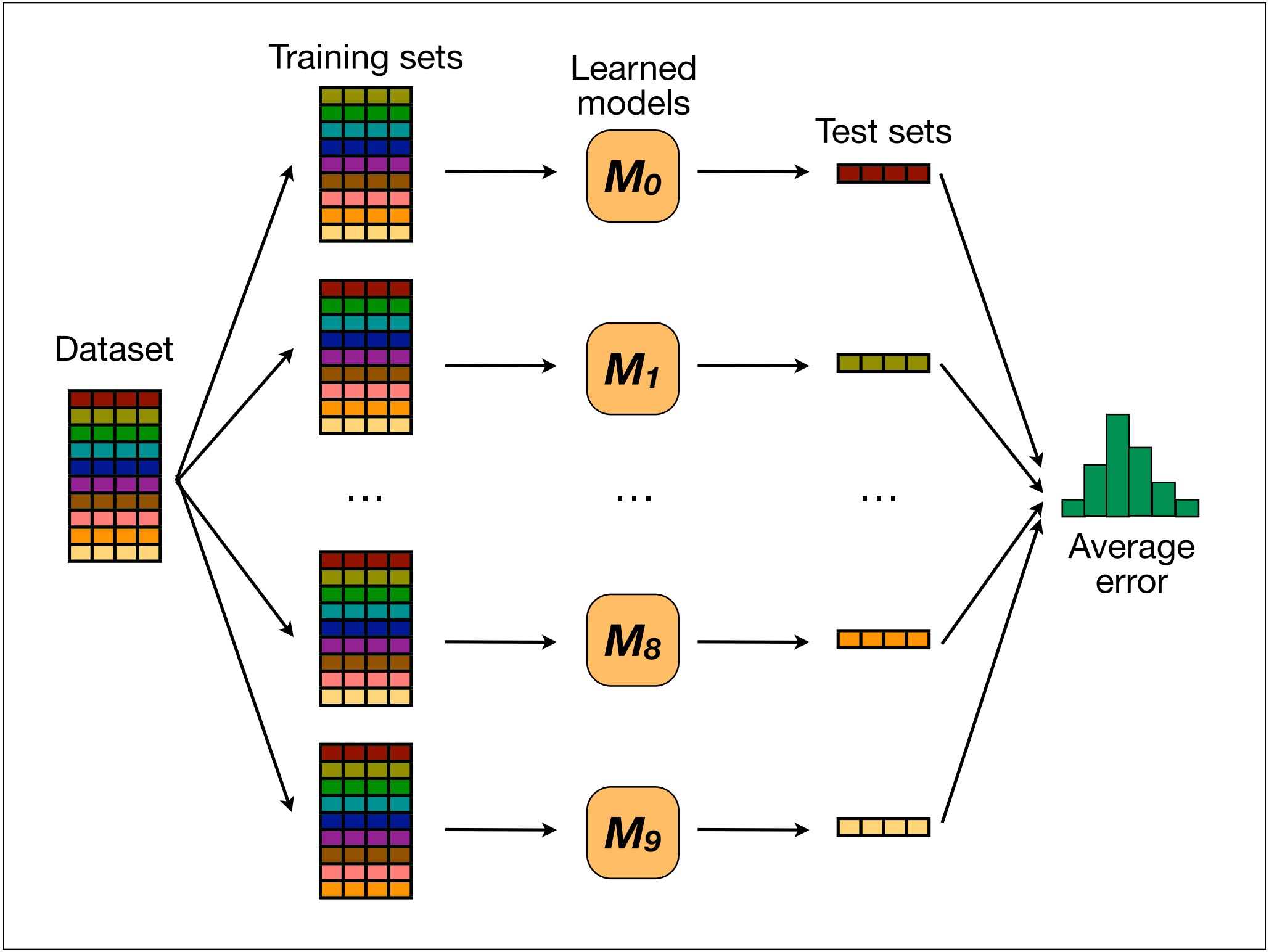
* Unpaired t-test can further reduce error

Measuring statistical power

- Statistical power:
  - Probability that the null hypothesis is rejected when the algorithms are not equal (i.e., test concludes that the performance difference is significant)
- To measure Type I error we need classifiers that are equivalent, to measure statistical power we need classifiers that perform **differently**
- Vary classifier difference
  - Perturb predictions of nBC and measure performance over 500 calibration sets
  - Perturbation rates = [0.025, 0.075, 0.15, 0.30] to increase difference between models

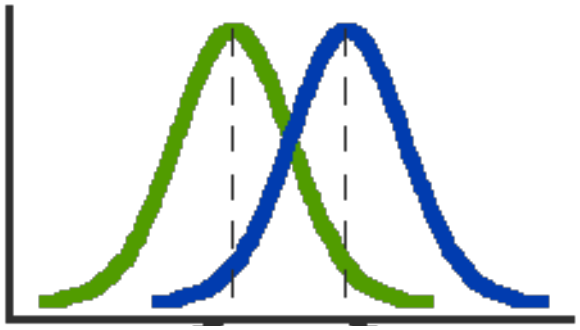
Methodology

- Adolescent Health Data
  - Survey information from 144 middle and high-schools, collected in 1994-1995
  - We used the social networks from size schools with similar autocorrelation and link patterns
  - Classification task: Whether a student smokes or not
  - Network sizes: 300-700; Average degree: 7-8; Autocorrelation: [0.25,0.35]
- Real classifiers
  - wvRN: no learning, just assume autocorrelation exists and infer label as average of neighbor labels
  - nBC: learn CPD to predict class label based on neighbor labels, use collective inference to propagate inference
- Equalize classifiers
  - 500 calibration sets created from 5 held-out schools, for each of the six schools.


Training sets
Learned models
Test sets
Dataset
M_0
M_1
M_8
M_9
Average error

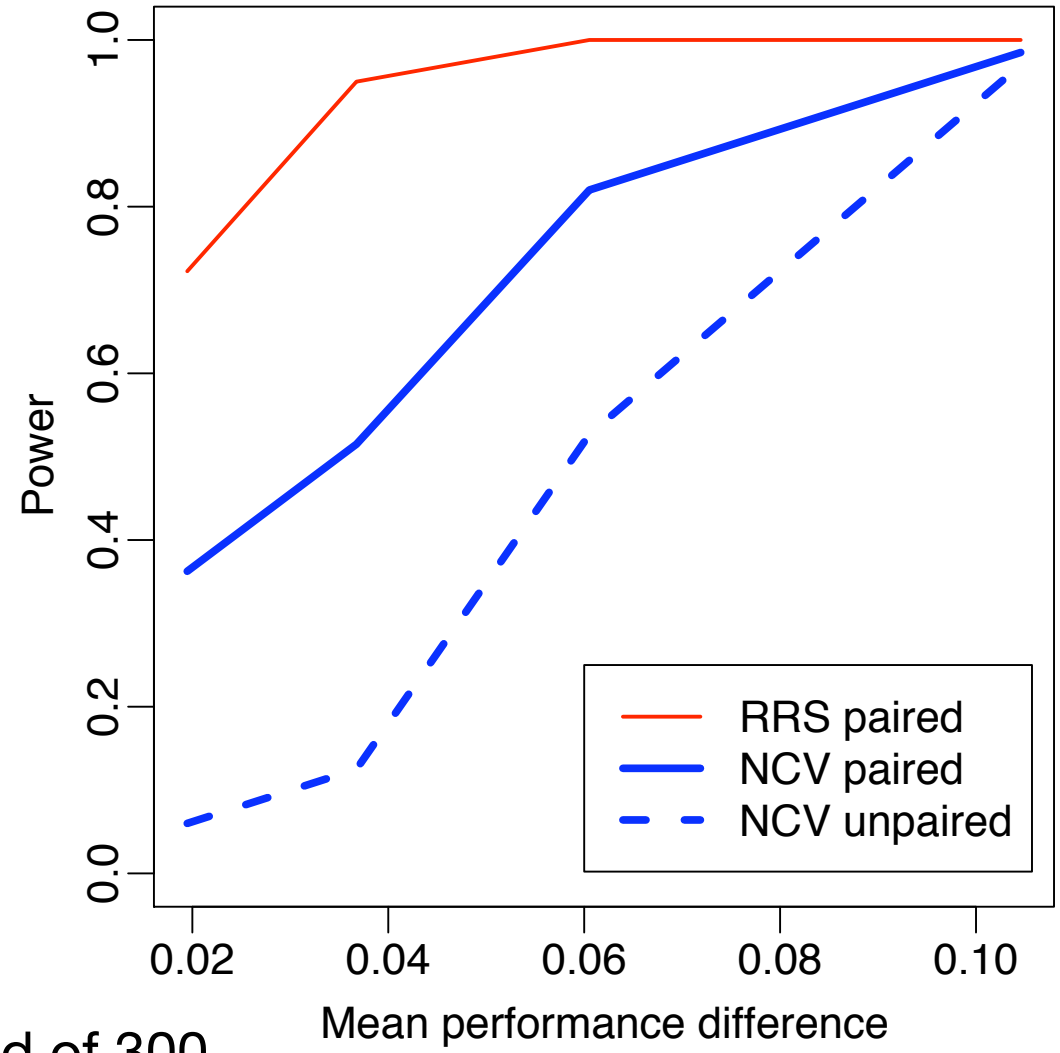
Comparison of algorithm performance

- Typical approach: two sample **t-test**
- Assess whether observed difference in algorithm performance is **significant**
- Compare to differences that would be observed under the null hypothesis
  (H_0: error rates are drawn from same distribution)
- T-test assumptions
  - Population is normally distributed
  - Variance of two populations are equal
  - Samples are independent, random draws from the population



$$t = \frac{\bar{X}_1 - \bar{X}_1}{S_{X_1 X_2} \cdot \sqrt{\frac{2}{n}}}$$

Larger network size increases power



* 600 nodes instead of 300

Previous results *(Dietterich '98)*

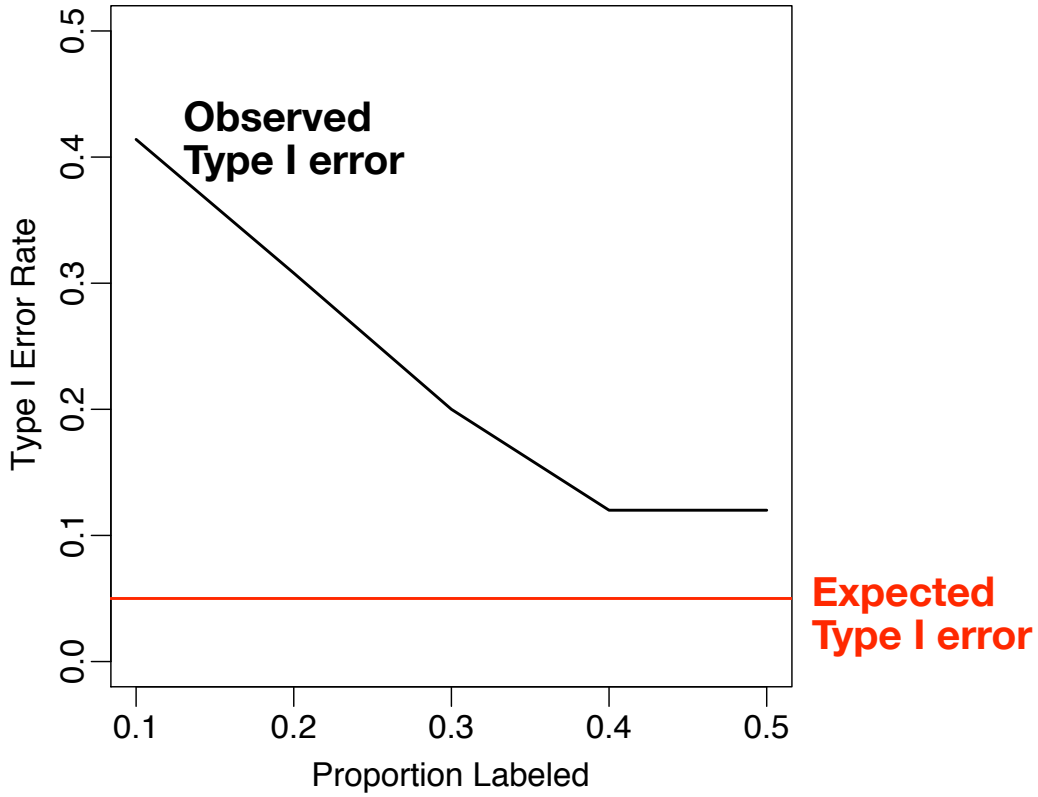
- T-tests results are **biased** if performance is estimated from **overlapping** test sets
  - Overlapping samples leads to **underestimation** of variance, which increases the probability of Type I error

$$t = \frac{\bar{X}_1 - \bar{X}_2}{S_{X_1 X_2} \cdot \sqrt{\frac{2}{n}}}$$

- **Type I error**:
  t-test concludes the algorithms are different when in fact they are not

- **Recommendation**:
  Use cross-validation to eliminate overlap in test sets

Network sampling



Typical approach

Use **repeated random sampling** to create multiple training/test (labeled/unlabeled) splits