Statistical Tests for Comparing Supervised Classification Learning Algorithms

Thomas G. Dietterich
tgd@cs.orst.edu
Department of Computer Science
Oregon State University
Corvallis, OR 97331

June 27, 1997

Abstract

This paper reviews five statistical tests for determining whether one learning algorithm outperforms another on a particular learning task. These tests are compared experimentally to determine their probability of incorrectly detecting a difference when no difference exists (type 1 error). Two widely-used statistical tests are shown to have high probability of Type I error in certain situations and should never be used. These tests are (a) a test for the difference of two proportions and (b) a paired-differences t test based on taking several random train/test splits. A third test, a paired-differences t test based on 10-fold cross-validation, exhibits somewhat elevated probability of Type I error. A fourth test, McNemar's test, is shown to have low Type I error. The fifth test is a new test, 5x2cv, based on 5 iterations of 2-fold cross-validation. Experiments show that this test also has good Type I error. The paper also measures the power (ability to detect algorithm differences when they do exist) of these tests. The cross-validated t test is the most powerful. The 5x2cv test is shown to be slightly more powerful than McNemar's test. The choice of the best test is determined by the computational cost of running the learning algorithm. For algorithms that can be executed only once, McNemar's test is the only test with acceptable Type I error. For algorithms that can be executed ten times, the 5x2cv test is recommended, because it is slightly more powerful and because it directly measures variation due to the choice of training set.

1 Introduction

In the research, development, and application of machine learning algorithms for classification tasks, many questions arise for which statistical methods are needed. The purpose of this paper is to investigate one of these questions, demonstrate that existing statistical methods are inadequate for this question, and propose a new statistical test that shows acceptable performance in initial experiments.

To understand the question raised in this paper, it is helpful to consider a taxonomy of the different kinds of statistical questions that arise in machine learning. Figure 1 gives a taxonomy of nine statistical questions.

Let us begin at the root of the tree. The first issue we must consider is whether we are studying only a single application domain or whether we are studying multiple domains. In most applied research, there is a single domain of interest, and our goal is to find the best classifier or the best learning algorithm to apply in that domain. However, a fundamental goal of research in machine learning is to find learning algorithms that work well over a wide range of application domains. We will return to this issue below; for the moment, let us consider the single-domain case.

Within a single domain, there are two different sets of questions depending on whether we are analyzing classifiers or algorithms. A classifier is a function that, given an input example, assigns that example to one of K classes. A learning algorithm is a function that, given a set of examples and their classes, constructs a classifier. In a particular application setting, our primary goal is usually to find the best classifier and estimate its accuracy on future examples. For example, suppose we are working for a medical instrumentation company and we wish to manufacture and sell an instrument for classifying blood cells. At the time we are designing the instrument, we could gather a large collection of blood cells and have a human expert classify each cell. We could then apply a learning algorithm to produce a classifier from this set of classified cells. The classifier would be implemented in the instrument and sold. We want our instrument to contain the most accurate classifier we can find.

There are some applications, however, where we must select the best learning algorithm rather than finding the best classifier. For example, suppose we work for a computer company. Our company wishes to develop a smart disk controller that tries to predict, at each point in time, which cylinder of the disk will be accessed next. This disk controller will contain a learning algorithm that will analyze the last 1000 disk access events and produce a classifier that can predict future disk cylinder requests. When we design the controller, we must choose which learning algorithm to install in it. Hence, this is a case where we want to choose the algorithm that, when trained on future data sets of 1000 disk events, will output the most accurate classifier. (We might also want the algorithm to be incremental and efficient!)

The next level of the taxonomy distinguishes between two fundamental tasks: estimating accuracy and choosing between classifiers (or algorithms). When we market our blood cell diagnosis system, we would like to make a claim about its accuracy. How can we measure this accuracy? And of course, when we design the system, we want to choose the best classifier from some set of available classifiers.

The lowest level of the taxonomy concerns the amount of data available. If we have a large amount of data, then we can set some of it aside to serve as a test set for evaluating classifiers. Much simpler statistical methods can be applied in this case. However, in most situations, the amount of data is limited and we need to use all of it as input to our learning algorithms. This means that we must use some form of resampling (i.e., cross-validation or the bootstrap) to perform the statistical analysis.

Now that we have reviewed the general structure of the taxonomy, let's consider the nine

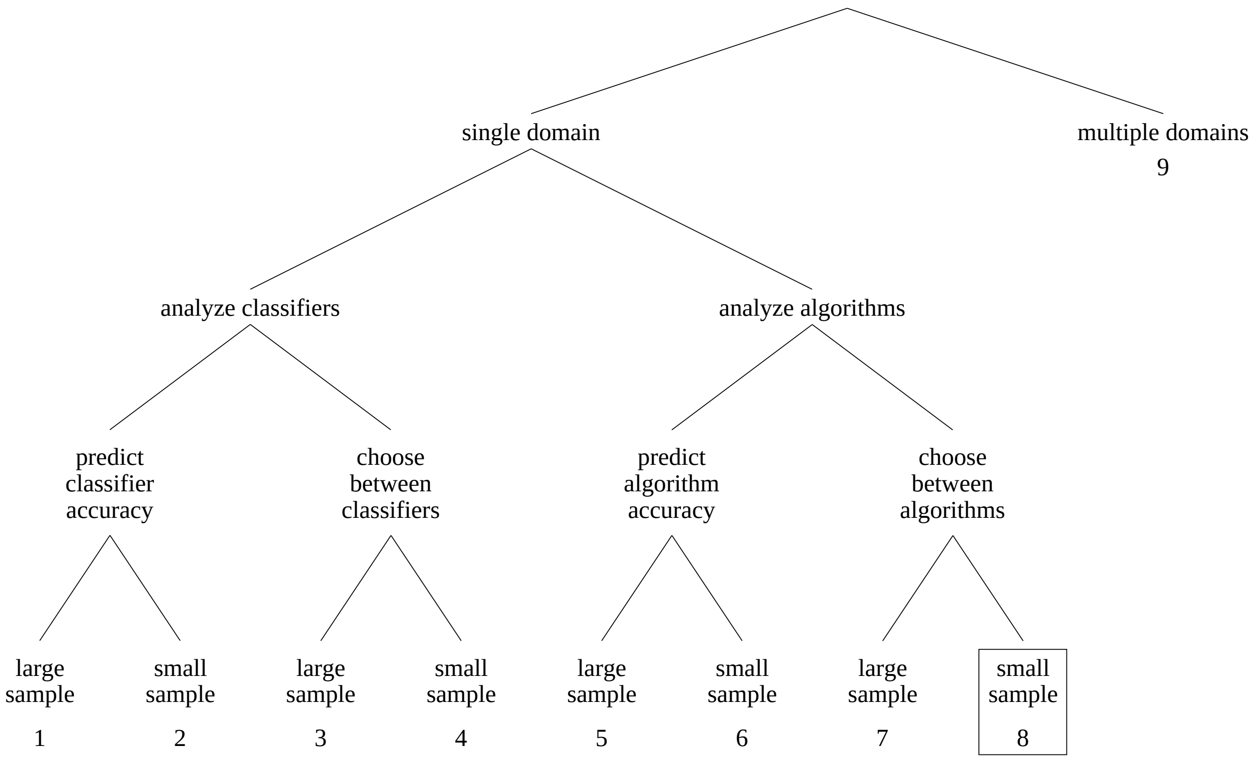

Figure 1: A taxonomy of statistical questions in machine learning. The boxed node (Question 8) is the subject of this paper.

statistical questions. We assume that all data points (examples) are drawn independently from a fixed probability distribution defined by the particular application problem.

Question 1: Suppose we are given a large sample of data and a classifier C. The classifier C may have been constructed using part of the data, but there is enough data remaining for a separate test set. Hence, we can measure the accuracy of C on the test set and construct a binomial confidence interval (Snedecor & Cochran, 1989; Efron & Tibshirani, 1993; Kohavi, 1995). Note that in Question 1, the classifier could have been produced by any method (e.g., interviewing an expert); it need not have been produced by a learning algorithm.

Question 2: Given a small data set, S, suppose we apply learning algorithm A to S to construct classifier C_A; how accurately will C_A classify new examples? Because we have no separate test set, there is no direct way to answer this question. A frequently-applied strategy is to convert this question into Question 6: Can we predict the accuracy of algorithm A when it is trained on randomly-selected data sets of (approximately) the same size as S? If so, then we can predict the accuracy of C_A which was obtained from training on S.

Question 3: Given two classifiers C_A and C_B and enough data for a separate test set, determine which classifier will be more accurate on new test examples. This question can be answered by measuring the accuracy of each classifier on the separate test set and applying McNemar's test, which will be described below.

Question 4: Given two classifiers C_A and C_B produced by feeding a small data set S to two learning algorithms A and B, which classifier will be more accurate in classifying new examples? Again, because we have no separate set of test data, we cannot answer this question directly.

Again, some researchers have taken the approach of converting this problem into a question about learning algorithms (Question 8). If we can determine which algorithm usually produces more accurate classifiers (when trained on data sets of approximately the same size), then we can select the classifier (C_A or C_B) created by that algorithm.

Question 5: Given a learning algorithm A and a large data set S, what is the accuracy of the classifiers produced by A when trained on new training sets of a specified size m? This question has not received much attention in the literature. One approach, advocated by the DELVE project (Hinton, Neal, Tibshirani, & DELVE team members, 1995; Rasmussen, 1996), is to subdivide S into a test set and several disjoint training sets of size m. Then A is trained on each of the training sets and the resulting classifiers are tested on the test set. The average performance on the test set estimates the accuracy of new runs.

Question 6: Given a learning algorithm A and a small data set S, what is the accuracy of the classifiers produced by A when A is trained on new training sets of the same size as S? Kohavi (1995) shows that stratified 10-fold cross-validation produces fairly good estimates in this case. Note that in any resampling approach, we cannot train A on training sets of exactly the same size as S. Instead, we train on data sets that have slightly fewer examples (e.g., 90% of the size of S in 10-fold cross-validation) and rely on the assumption that the performance of learning algorithms changes smoothly with changes in the size of the training data. This assumption can be checked experimentally (by performing additional cross-validation studies) with even smaller training sets, but it cannot be checked directly for training sets of the size of S. Results on the shape of learning curves show that in some cases this smoothness assumption will be violated (Haussler, Kearns, Seung, & Tishby, 1994). Nonetheless, it is observed to hold experimentally in most applications.

Question 7: Given two learning algorithms A and B and a large data set S, which algorithm will produce more accurate classifiers when trained on data sets of a specified size m? This question has not received much attention, although the DELVE team has studied this question for regression problems. They divide S into several disjoint training sets and a single test set. Each algorithm is trained on each training set, and all resulting classifiers are tested on the test set. An analysis of variance can then be performed that includes terms for the choice of learning algorithm, the choice of the training set, and each individual test example. The Quasi-F test (Lindman, 1992) is applied to determine whether the effect due to the choice of learning algorithms is significantly non-zero.

Question 8: Given two learning algorithms A and B and a small data set S, which algorithm will produce more accurate classifiers when trained on data sets of the same size as S? The purpose of this paper is to describe and compare several statistical tests for answering this question. Because S is small, it will be necessary to use holdout and resampling methods. As mentioned in Question 6 above, this means that we cannot answer this question exactly without making the assumption that the performance of the two learning algorithms changes smoothly with changes in the size of the training set. Specifically, we will need to assume that the relative difference in performance of the two algorithms changes slowly with changes in the size of the training set.

Question 9: Given two learning algorithms A and B and data sets from several domains, which algorithm will produce more accurate classifiers when trained on examples from new domains? This is perhaps the most fundamental and difficult question in machine learning. Some researchers have applied a simple sign test (or the Wilcoxon signed-ranks test) to try to answer this question, based on single runs or cross-validation-based estimates, but these tests do not take into account the uncertainty of the individual comparisons. Effectively, we want to combine the results from several answers to Question 8, where each answer has an associated uncertainty. This is an important question for future research.

Questions 7, 8, and 9 are the most important for experimental research on learning algorithms. When someone develops a new learning algorithm (or a modification to an existing algorithm), answers to these questions can determine whether the new algorithm is better than existing algorithms. Unfortunately, many data sets used in experimental research are too small to allow posing Question 7. Hence, this paper focuses on developed good statistical tests for Question 8. We define and compare five statistical tests for this question.

Before proceeding with the derivation of these statistical tests, it is worth noting that each of the questions posed above can be extended beyond classification algorithms and misclassification rates. For example, in many decision-making settings, it is important to estimate the conditional probability that a new example belongs to each of the K classes. One measure of the accuracy of probability estimates is the log loss, and Questions 1, 2, 5, and 6 can all be rephrased in terms of determining the expected log loss of a classifier or an algorithm. Similarly, Questions 3, 4, 7, and 8 can be rephrased in terms of determining which classifier or algorithm has the smaller log loss. We are unaware of any statistical research specifically addressing these questions in the case of log loss, however.

In many neural network applications, the task is to predict a continuous response variable. In these problems, the squared error is usually the natural loss function, and Questions 1, 2, 5, and 6 can be rephrased in terms of determining the expected mean squared error of a predictor or of an algorithm. Similarly, Questions 3, 4, 7, and 8 can be rephrased in terms of determining which predictor or algorithm has the smaller mean squared error. Question 1 can be addressed by constructing a confidence interval based on the normal or t distribution (depending on the size of the test set). Question 3 can be addressed by constructing a confidence interval for the expected difference. As mentioned above, The DELVE project has developed analysis of variance techniques for Questions 5 and 7. Appropriate statistical tests for the small sample questions (2, 4, 6, and 8) are still not well established.

The statistical tests for regression methods may suggest ways of designing statistical tests for the log loss case. This is an important area for future research.

To design and evaluate statistical tests, the first step is to identify the sources of variation that must be controlled by each test. For the case we are considering, there are four important sources of variation.

First, there is the random variation in the selection of the test data that is used to evaluate the learning algorithms. On any particular randomly-drawn test data set, one classifier may outperform another even though on the whole population the two classifiers would perform identically. This is a particularly pressing problem for small test data sets.

The second source of random variation results from the selection of the training data. On any particular randomly-drawn training set, one algorithm may out-perform another even though, on the average, the two algorithms have the same accuracy. Even small changes to the training set (such as adding or deleting a few data points) may cause large changes in the classifier produced by a learning algorithm. Breiman (1994, 1996) has called this behavior "instability", and he has shown that this is a serious problem for the decision tree algorithms, such as CART (Breiman, Friedman, Olshen, & Stone, 1984).

A third source of variance can be internal randomness in the learning algorithm. Consider, for example, the widely-used backpropagation algorithm for training feed-forward neural networks. This algorithm is usually initialized with a set of random weights which it then improves. The resulting learned network depends critically on the random starting state (Kolen & Pollack, 1991). In this case, even if the training data are not changed, the algorithm is likely to produce a different hypothesis if it is executed again from a different random starting state.

The last source of random variation that must be handled by statistical tests is random classification error. If a fixed fraction η of the test data points are randomly mislabeled, then no learning algorithm can achieve an error rate of less than η.

A good statistical test should not be fooled by these sources of variation. The test should conclude that the two algorithms are different if and only if their percentage of correct classifications would be different, on the average, when trained on a training set of a given fixed size and tested on all data points in the population.

To accomplish this, a statistical testing procedure must account for these sources of variation. To account for test-data variation and the possibility of random classification error, the statistical procedure must consider the size of the test set and the consequences of changes in the test set. To account for training-data variation and internal randomness, the statistical procedure must execute the learning algorithm multiple times and measure the variation in accuracy of the resulting classifiers.

This paper begins by describing five statistical tests for Question 8: McNemar's test, a test for the difference of two proportions, the resampled t test, the cross-validated t test, and a new test called the 5x2cv test. The paper then describes a simulation study that seeks to measure the probability that each test will incorrectly detect a difference when no difference exists (Type I error). The results of the simulation study show that only McNemar's test, the cross-validated t test, and the 5x2cv test have acceptable Type I error. The Type I error of the resampled t test is very bad and the test is very expensive computationally, so we do not consider it further. The Type I error of the difference-of-proportions test is unacceptable in some cases, but it is very cheap to evaluate, so we retained it for further study.

The simulation study is somewhat idealized and does not address all aspects of training-data variation. To obtain a more realistic evaluation of the four remaining tests, we conducted a set of experiments using real learning algorithms on realistic data sets. We measured both the Type I error and the power of the tests. The results show that the cross-validated t test has consistently elevated Type I error. The difference-of-proportions test has acceptable type I error, but low power. Both of the remaining two tests have good Type I error and reasonable power. The 5x2cv test is slightly more powerful than McNemar's test, but also ten times more expensive to perform. Hence, the paper concludes that the 5x2cv test is the test of choice for inexpensive learning algorithms, but that McNemar's test is better for more expensive algorithms. Lisp code implementing both tests is available from the author.

2 Formal Preliminaries

We will assume that there exists a set X of possible data points, called the *population*. There also exists some *target function*, f that classifies each $x \in X$ into one of K classes. Without loss of generality, we will assume that $K = 2$, although none of the results in this paper depend on this assumption, since our only concern will be whether an example is classified correctly or incorrectly.

In an application setting, a sample S is drawn randomly from X according to a fixed probability distribution D. A collection of *training examples* is constructed by labeling each $x \in S$ according to $f(x)$. Each training example therefore has the form $\langle x, f(x) \rangle$. In some applications, there may be a source of classification noise that randomly sets the label to an incorrect value.

A learning algorithm A takes as input a set of training examples R and outputs a classifier $\hat{f}$. The true error rate of that classifier is the probability that $\hat{f}$ will misclassify an example drawn randomly from X according to D. In practice, this error rate is estimated by taking our available sample S and subdividing it into a training set R and a test set T. The error rate of $\hat{f}$ on T provides an estimate of the true error rate of $\hat{f}$ on the population, X.

The null hypothesis to be tested is that for a randomly drawn training set R of fixed size, the two learning algorithms will have the same error rate on a test example randomly-drawn from X, where all random draws are made according to distribution D. Let $\hat{f}_A$ be the classifier output by algorithm A trained on training set R, and let $\hat{f}_B$ be the classifier output by algorithm B trained on R. Then the null hypothesis can be written as

$$Pr_{R,x}[\hat{f}_A(x) = f(x)] = Pr_{R,x}[\hat{f}_B(x) = f(x)],$$

where the notation $Pr_{R,x}$ indicates the probability taken with respect to the random draws of the training set R and the test example x.

3 Five Statistical Tests

We now describe the statistical tests that are the main subject of this paper. We begin with simple holdout tests and then consider tests based on resampling from the available data.

3.1 McNemar's test

To apply McNemar's test (Everitt, 1977), we divide our available sample of data S, into a training set R and a test set T. We train both algorithms A and B on the training set yielding classifiers $\hat{f}_A$ and $\hat{f}_B$. We then test these classifiers on the test set. For each example $x \in T$, we record how it was classified and construct the following contingency table:

number of examples misclassified by both $\hat{f}_A$ and $\hat{f}_B$	number of examples misclassified by $\hat{f}_A$ but not by $\hat{f}_B$
number of examples misclassified by $\hat{f}_B$ but not by $\hat{f}_A$	number of examples misclassified by neither $\hat{f}_A$ nor $\hat{f}_B$.

We will use the notation

n_{00}	n_{01}
n_{10}	n_{11}

where $n = n_{00} + n_{01} + n_{10} + n_{11}$ is the total number of examples in the test set T.

Under the null hypothesis, the two algorithms should have the same error rate, which means that $n_{01} = n_{10}$. McNemar's test is based on a χ^2 test for goodness-of-fit that compares the distribution of counts expected under the null hypothesis to the observed counts. The expected counts under the null hypothesis are

n_{00}	$(n_{01} + n_{10})/2$
$(n_{01} + n_{10})/2$	n_{11}

The following statistic is distributed (approximately) as χ^2 with 1 degree of freedom; it incorporates a "continuity correction" term (of -1 in the numerator) to account for the fact that the statistic is discrete while the χ^2 distribution is continuous:

$$\frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}}.$$

If the null hypothesis is correct, then the probability that this quantity is greater than $\chi^2_{1,0.95} = 3.841459$ is less than 0.05. So we may reject the null hypothesis in favor of the hypothesis that the two algorithms have different performance when trained on the particular training set R.

Note however, that this test has two shortcomings with regard to Question 8. First, it does not directly measure variability due to the choice of the training set or the internal randomness of the learning algorithm. A single training set R is chosen, and the algorithms are only compared using that training set. Hence, McNemar's test should only be applied if we believe these sources of variability are small. Second, it does not directly compare the performance of the algorithms on training sets of size $|S|$, but only on sets of size $|R|$, which must be substantially smaller than $|S|$ to ensure a sufficiently large test set. Hence, we must assume that the relative difference observed on training sets of size $|R|$ will still hold for training sets of size $|S|$.

3.2 A test for the difference of two proportions

A second simple statistical test is based on measuring the difference between the error rate of algorithm A and the error rate of algorithm B (Snedecor & Cochran, 1989). Specifically, let $p_A = (n_{00} + n_{01})/n$ be the proportion of test examples incorrectly classified by algorithm A and let $p_B = (n_{00} + n_{10})/n$ be the proportion of test examples incorrectly classified by algorithm B. The assumption underlying this statistical test is that when algorithm A classifies an example x from the test set T, the probability of misclassification is p_A. Hence, the number of misclassifications of n test examples is a binomial random variable with mean np_A and variance $p_A(1 - p_A)/n$.

The binomial distribution can be well-approximated by a normal distribution for reasonable values of n. Furthermore, the difference between two *independent* normally distributed random variables is itself normally distributed. Hence, the quantity $p_A - p_B$ can be viewed as normally distributed if we assume that the measured error rates p_A and p_B are independent. Under the null hypothesis, this will have a mean of zero and a standard error of

$$se = \sqrt{\frac{2p(1-p)}{n}}$$

where $p = (p_A + p_B)/2$ is the average of the two error probabilities.

From this analysis, we obtain the statistic

$$z = \frac{p_A - p_B}{\sqrt{2p(1-p)/n}}$$

which has (approximately) a standard normal distribution. We can reject the null hypothesis if $|z| > Z_{0.975} = 1.96$ (for a 2-sided test with probability of incorrectly rejecting the null hypothesis of 0.05).

This test has been used by many researchers including the author (Dietterich, Hild, & Bakiri, 1995). However, there are several problems with this test. First, because p_A and p_B are each measured on the same test set T, they are not independent. Second, the test shares the drawbacks of McNemar's test: it does not measure variation due to the choice of training set or internal variation of the learning algorithm and it does not directly measure the performance of the algorithms on training sets of size $|S|$, but rather on the smaller training set of size $|R|$.

The lack of independence of p_A and p_B can be corrected by changing the estimate of the standard error to be

$$se' = \sqrt{\frac{n_{01} + n_{10}}{n^2}}.$$

This estimate focuses on the probability of disagreement of the two algorithms (Snedecor & Cochran, 1989). The resulting z statistic can be written as

$$z' = \frac{|n_{01} - n_{10}| - 1}{\sqrt{n_{01} + n_{10}}},$$

which we can recognize as the square root of the χ^2 statistic in McNemar's test.

In this paper, we have experimentally analyzed the uncorrected z statistic, since this statistic is in current use and we wanted to determine how badly the (incorrect) independence assumption affects the accuracy of the test.

For small sample sizes, there are exact versions of both McNemar's test and the test for the difference of two proportions that avoid the χ^2 and normal approximations.

3.3 The resampled paired t test

The next statistical test we consider is currently the most popular in the machine learning literature. A series of (usually) 30 trials is conducted. In each trial, the available sample S is randomly divided into a training set R of a specified size (e.g., typically two thirds of the data) and a test set T. Learning algorithms A and B are both trained on R and the resulting classifiers are tested on T. Let $p_A^{(i)}$ (respectively, $p_B^{(i)}$) be the observed proportion of test examples misclassified by algorithm A (respectively B) during trial i. If we assume that the 30 differences $p^{(i)} = p_A^{(i)} - p_B^{(i)}$ were drawn independently from a normal distribution, then we can apply Student's t test, by computing the statistic

$$t = \frac{\overline{p} \cdot \sqrt{n}}{\sqrt{\frac{\sum_{i=1}^{n} (p^{(i)} - \overline{p})^2}{n-1}}},$$

where $\overline{p} = \frac{1}{n} \sum_{i=1}^{n} p^{(i)}$. Under the null hypothesis, this statistic has a t distribution with $n-1$ degrees of freedom. For 30 trials, the null hypothesis can be rejected if $|t| > t_{29,0.975} = 2.04523$.

There are many potential drawbacks of this approach. First, as observed above, the individual differences $p^{(i)}$ will not have a normal distribution, because $p_A^{(i)}$ and $p_B^{(i)}$ are not independent. Second, the $p^{(i)}$'s are not independent, because the test sets in the trials overlap (and the training sets in the trials overlap as well). We will see below that these violations of the assumptions underlying the t test cause severe problems that make this test unsafe to use.

3.4 The k-fold cross-validated paired t test

This test is identical to the previous one except that instead of constructing each pair of training and test sets by randomly dividing S, we instead randomly divide S into k disjoint sets of equal size, $T_1, \ldots, T_k$. We then conduct k trials. In each trial, the test set is T_i, and the training set is the union of all of the other $T_j, j \neq i$. The same t statistic is computed.

The advantage of this approach is that each test set is independent of the others. However, this test still suffers from the problem that the training sets overlap. In a 10-fold cross-validation, each pair of training sets shares 80% of the examples. This overlap may prevent this statistical test from obtaining a good estimate of the amount of variation that would be observed if each training set were completely independent of previous training sets.

To illustrate this point, consider the nearest neighbor algorithm. Suppose our training set contains two clusters of points: a large cluster belonging to one class and a small cluster belonging to the other class. If we perform a 2-fold cross-validation, we must subdivide this training data into two disjoint sets. If all of the points in the smaller cluster go into one of those two sets, then both runs of the nearest neighbor algorithm will have elevated error rates, because when the small cluster is in the test set, every point in it will be misclassified. When the small cluster is in the training set, some of its points may (incorrectly) be treated as nearest neighbors of test set points, which also increases the error rate. Conversely, if the small cluster is evenly divided between the two sets, then the error rates will improve, because for each test point there will be a corresponding nearby training point that will provide the correct classification. Either way, we can see that the performance of the two folds of the cross-validation will be correlated rather than independent.

We verified this experimentally for 10-fold cross-validation on the Letter Recognition task (300 total training examples) in an experiment where the null hypothesis was true (described below). We measured the correlation coefficient between the differences in error rates on two folds within a cross-validation, $p^{(i)}$ and $p^{(j)}$. The observed value was 0.03778, which according to a t test is significantly different from 0 with $p < 10^{-10}$. On the other hand, if the error rates $p^{(i)}$ and $p^{(j)}$ are drawn from independent 10-fold cross-validations (i.e., on independent data sets), the correlation coefficient is -0.00014, which according to a t test is not significantly different from zero.

3.5 The 5x2cv paired t test

In some initial experiments with the k-fold cross-validated paired t test, we attempted to determine why the t statistic was too large in some cases. The numerator of the t statistic estimates the mean difference in the performance of the two algorithms (over the k folds), while the denominator estimates the variance of these differences. With synthetic data, we constructed k non-overlapping training sets and measured the mean and variance on those training sets. We found that while the variance was slightly underestimated when the training sets overlapped, the means were occasionally very poorly estimated, and this was the cause of the large t values. The problem can be traced to the correlations between the different folds as described above.

We found that if we replaced the numerator of the t statistic with the observed difference from a *single* fold of the k-fold cross-validation, the statistic became well-behaved. This led us to the 5x2cv paired t test.

In this test, we perform 5 replications of 2-fold cross-validation. In each replication, the available data is randomly partitioned into two equal-sized sets S_1 and S_2. Each learning algorithm (A or B) is trained on each set and tested on the other set. This produces four error estimates: $p_A^{(1)}$ and $p_B^{(1)}$ (trained on S_1 and tested on S_2) and $p_A^{(2)}$ and $p_B^{(2)}$ (trained on S_2 and tested on S_1). Subtracting corresponding error estimates gives us two estimated differences $p^{(1)} = p_A^{(1)} - p_B^{(1)}$ and $p^{(2)} = p_A^{(2)} - p_B^{(2)}$. From these two differences, the estimated variance is $s^2 = (p^{(1)} - \overline{p})^2 + (p^{(2)} - \overline{p})^2$. Let s_i^2 be the variance computed from the i-th replication, and let $p_1^{(1)}$ be the $p^{(1)}$ from the very first of the five replications. Then define the following statistic

$$\tilde{t} = \frac{p_1^{(1)}}{\sqrt{\frac{1}{5}\sum_{i=1}^{5} s_i^2}},$$

which we will call the 5x2cv $\tilde{t}$ statistic. We claim that under the null hypothesis, $\tilde{t}$ has approximately a t distribution with 5 degrees of freedom. The argument goes as follows.

Let A be a standard normal random variable and B be a χ^2 random variable with $n-1$ degrees of freedom. Then by definition, the quantity

$$\frac{A}{\sqrt{B/(n-1)}} \tag{1}$$

has a t distribution with $n-1$ degrees of freedom if A and B are independent.

The usual t statistic is derived by starting with a set of random variables $X_1, \ldots X_n$ having a normal distribution with mean μ and variance σ^2. Let $\overline{X} = \frac{1}{n}\sum_i X_i$ be the sample mean and $S^2 = \sum_i (X_i - \overline{X})^2$ be the sum of squared deviations from the mean. Then define

$$\begin{aligned} A &= \sqrt{n}(\overline{X} - \mu)/\sigma \\ B &= S^2/\sigma^2. \end{aligned}$$

Well-known results from probability theory state that A has a standard normal distribution and B has a χ^2 distribution with $n-1$ degrees of freedom. A more remarkable result from probability theory is that A and B are also independent, provided the original X_i's were drawn from a normal distribution. Hence, we can plug them into Equation (1) as follows:

$$\begin{aligned} t &= \frac{A}{\sqrt{B/(n-1)}} \\ &= \frac{\sqrt{n}(\overline{X}-\mu)/\sigma}{\sqrt{S^2/(\sigma^2 \cdot (n-1))}} \\ &= \frac{\sqrt{n}(\overline{X}-\mu)}{\sqrt{S^2/(n-1)}}. \end{aligned}$$

This gives the usual definition of the t statistic when $\mu = 0$.

We can construct $\tilde{t}$ by analogy as follows. Under the null hypothesis, the numerator of $\tilde{t}$, $p_1^{(1)}$, is the difference of two identically distributed proportions, so we can safely treat it as an approximately normal random variable with zero mean and unknown standard deviation σ if the underlying test set contained at least 30 points. Hence, let $A = p_1^{(1)}/\sigma$.

Also under the null hypothesis, s_i^2/σ^2 has a χ^2 distribution with 1 degree of freedom if we make the additional assumption that $p_i^{(1)}$ and $p_i^{(2)}$ are independent. This assumption is false, as we have seen above, because these two differences-of-proportions are measured on the opposite folds of a 2-fold cross-validation. Still, the assumption of independence is probably more appropriate for 2-fold cross-validation than for 10-fold cross-validation, because in the 2-fold case, the training sets are completely non-overlapping (and, as always in cross-validation, the test sets are non-overlapping).

(We chose 2-fold cross-validation because it gives large test sets and disjoint training sets. The large test set is needed, because we are only using one paired difference $p_1^{(1)}$ in $\tilde{t}$. The disjoint training sets help make $p_i^{(1)}$ and $p_i^{(2)}$ more independent. A drawback, of course, is that the learning algorithms are trained on training sets half of the size of the training sets for which, under Question 8, we seek their relative performance.)

We could set $B = s_1^2/\sigma^2$, but when we tested this experimentally, we found that the resulting estimate of the variance was very noisy, and often zero. In similar situations, others have found that combining the results of multiple cross-validations can help stabilize an estimate, so we perform five 2-fold cross-validations and define

$$B = \left(\sum_{i=1}^{5} s_i^2 \right) \Big/ \sigma^2.$$

If we assume that the s_i^2 from each 2-fold cross-validation are independent of each other, then B is the sum of 5 independent random variables each having a χ^2 distribution with 1 degree of freedom. By the summation property of the χ^2 distribution, this means B has a χ^2 distribution with 5 degrees of freedom. This last independence assumption is also false, because each 2-fold cross-validation is computed from the same training data. However, experimental tests showed that this is the least problematic of the various independence assumptions underlying the 5x2cv test.

Finally, to use Equation (1), we must make the assumption that the variance estimates s_i are independent of $p_1^{(1)}$. This must be assumed (rather than proved as in the usual t distribution derivation), because we are using only one of the observed differences of proportions rather than the mean of all of the observed differences. As discussed above, the mean difference tends to overestimate the true difference, because of the lack of independence between the different folds of the cross-validation.

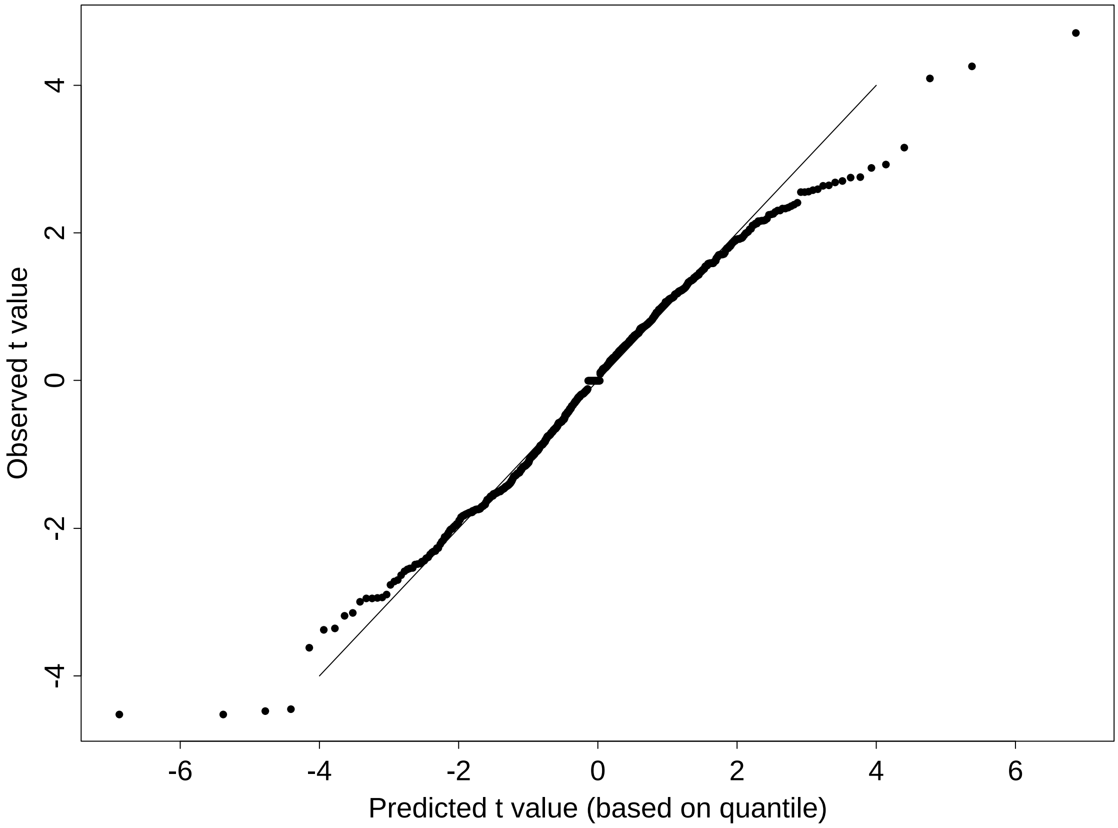

Figure 2: QQ plot comparing the distribution of 1,000 values of $\tilde{t}$ to the values they should have under a t distribution with 5 degrees of freedom. All points would fall on the line $y = x$ if the distributions matched.

With all of these assumptions, we can plug in to Equation (1) to obtain $\tilde{t}$.

Let us summarize the assumptions and approximations involved in this derivation. First, we employ the normal approximation to the binomial distribution. Second, we assume pairwise independence of $p_i^{(1)}$ and $p_i^{(2)}$ for all i. Third, we assume independence between the s_i's. Finally, we assume independence between the numerator and denominator of the $\tilde{t}$ statistic.

One way to evaluate the 5x2cv statistic experimentally is to make a quantile-quantile plot (QQ plot), as shown in Figure 2. The QQ plot shows 1,000 computed values of the 5x2cv statistic for a case where the null hypothesis is known to apply (the EXP6 task as described below). To generate a QQ plot, the 1,000 values are sorted and assigned quantiles (i.e., their rank in the sorted list divided by 1,000). Then the inverse cumulative t distribution (with 5 degrees of freedom) is used to compute for each quantile, the value that a t-distributed random variable would have taken if it had had that rank. This value becomes the x coordinate, and the original value $\tilde{t}$ becomes the y coordinate. In other words, for each observed point, based on its ordinal position within the 1,000 points, we can compute what value it should have had if the 1,000 points had been truly drawn from a t distribution. If the 1,000 points have a t distribution with 5 degrees of freedom, then they should lie on the line $y = x$. The figure shows a fairly good fit to the line. However, at the tails of the distribution, $\tilde{t}$ is somewhat more conservative than it should be.

Our choice of five replications of cross-validation is not arbitrary. Exploratory studies showed that using fewer or more than five replications increased the risk of Type I error. A possible explanation for this is that there are two competing problems. With fewer replications, the noise in the measurement of the s_i's becomes troublesome. With more replications, the lack of independence among the s_i's becomes troublesome. Whether five is the best value for the number of replications is an open question.

This completes our description of the five statistical tests of the null hypothesis that two learning algorithms have the same error rate on a given population (based on a sample of that population).

4 Simulation Experiment Design

We now turn to an experimental evaluation of these five methods. The purpose of the simulation was to measure the probability of Type I error of the algorithms. A Type I error occurs when the null hypothesis is true (i.e., there is no difference between the two learning algorithms) and the learning algorithm rejects the null hypothesis.

To measure the probability of Type I error, we constructed some simulated learning problems. To understand these problems, it is useful to think abstractly about the behavior of learning algorithms.

4.1 Simulating the behavior of learning algorithms

Consider a population of N data points and suppose that the training set size is fixed. Then for a given learning algorithm A, define $\epsilon_A(x)$ to be the probability that the classifier produced by A when trained on a randomly-drawn training set (of the fixed size) will misclassify x. If $\epsilon_A(x) = 0$, then x is always correctly classified by classifiers produced by A. If $\epsilon_A(x) = 1$, then x is always misclassified.

Figure 3 shows the measured values of $\epsilon(x)$ for a population of 7,670 points with respect to the C4.5 decision tree algorithm (Quinlan, 1993) trained on randomly-drawn training sets of 100 examples. The points were sorted by their ϵ values. Given these ϵ values, we could simulate the behavior of C4.5 on a randomly drawn test set of points by taking each point x and misclassifying it with probability $\epsilon(x)$. This would not exactly reproduce the behavior of C4.5, because it assumes that the misclassification errors made by C4.5 are independent for each test example, whereas in fact the classifications of data points that are close together will tend to be highly correlated. However, this simulated C4.5 procedure would have the same average error rate as the real C4.5 algorithm, and it will exhibit a similar degree of variation from one random trial to the next.

We can simulate learning algorithms with various properties by defining a population of points X and assigning a value $\epsilon(x)$ to each point. If we want a learning algorithm to have high variance, we can assign values of ϵ near 0.5, which is the value giving the maximum variance for a binomial random variable. If we want two learning algorithms to have the same error rate on the population, we can ensure that the average value of ϵ over the population X is the same for both algorithms.

In our studies, we wanted to construct simulated learning problems that would provide a "worst-case" for our statistical tests. To accomplish this, we sought to maximize the two main sources of random variation: variation resulting from the choice of test data sets and variation resulting from the choice of training sets. We ignored the issue of classification noise. Because it affects training and test data equally, it can be incorporated into the overall error rate. We also ignored internal

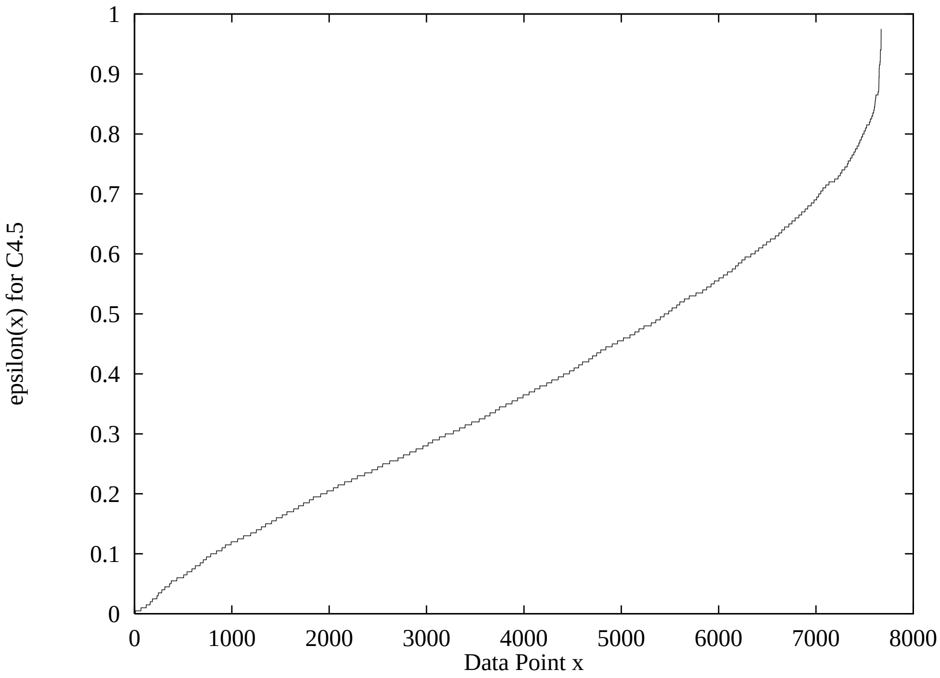

Figure 3: Measured values of $\epsilon_{C4.5}(x)$ for a population of 7670 data points.

randomness in the learning algorithms, since this will manifest itself in the same way as training set variance—by causing the same learning algorithm to produce different classifiers.

For tests of Type I error, we designed two sets of ϵ values: $\epsilon_A(x)$ for algorithm A and $\epsilon_B(x)$ for algorithm B. We established a target error rate ϵ and chose only two distinct values to use for $\epsilon_A(x)$ and $\epsilon_B(x)$: $\frac{1}{2}\epsilon$ and $\frac{3}{2}\epsilon$. We generated a population of points. For the first half of the population, we assigned $\epsilon_A(x) = \frac{1}{2}\epsilon$ and $\epsilon_B(x) = \frac{3}{2}\epsilon$. For the remaining half of the population, we reversed this and assigned $\epsilon_A(x) = \frac{3}{2}\epsilon$ and $\epsilon_B(x) = \frac{1}{2}\epsilon$. Figure 4 shows this configuration of ϵ values. The size of the population is irrelevant, because we are sampling with replacement and there are only two kinds of points. The important thing is that the population is evenly divided between these two kinds of training points.

The effect of this is that each algorithm has an overall error rate of ϵ, and each algorithm has the same total variance. However, for any given test example, the algorithms have very different error rates. This makes the effect of the random choice of test data sets very apparent. Indeed, unless the test data set is exactly equally divided between the first 50 data points and the second 50 data points, there will be an apparent advantage for one algorithm over the other. Our statistical tests will need to avoid being fooled by this apparent difference in error rates. Experimental measurements confirmed that these choices of $\epsilon_A(x)$ and $\epsilon_B(x)$ did the best job of simultaneously maximizing within-algorithm variance and between-algorithm variation while achieving the desired overall error rate of ϵ. In most of our experiments, we used a value of $\epsilon = 0.10$, although we also investigated $\epsilon = 0.20$, $\epsilon = 0.30$, and $\epsilon = 0.40$.

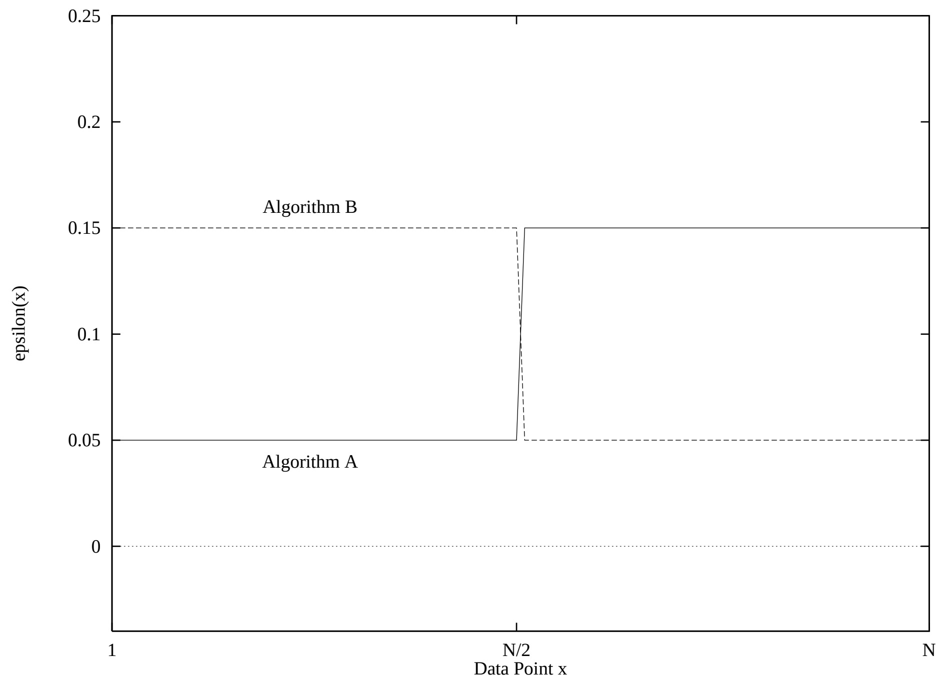

Figure 4: Designed values of $\epsilon_A(x)$ and $\epsilon_B(x)$ for an overall error rate of $\epsilon = 0.10$.

4.2 Details of the simulation of each statistical test

Each simulation is divided into a series of 1000 trials. In each trial, a data set S of size 300 is randomly drawn with replacement and then analyzed using each of the five statistical tests described above. The goal is to measure the proportion of trials in which the null hypothesis is rejected.

For the first two tests, McNemar's test and the normal test for the difference of two proportions, the data set S is randomly divided into a training set R containing two thirds of S and a test set T containing the remaining one third of S. The training set is ignored, because we do not actually execute the learning algorithms. Rather, the performance of each algorithm on the test set is simulated by classifying each test example x randomly according to its value of $\epsilon_A(x)$ and $\epsilon_B(x)$. A random number in the range [0,1) is drawn. If it is less than $\epsilon_A(x)$, then x is considered to be misclassified by algorithm A. A second random number is drawn, and x is considered misclassified by algorithm B if that number is less than $\epsilon_B(x)$. The results of these classifications are then processed by the appropriate statistical test. All tests were performed using 2-sided tests with confidence level 0.05.

For the resampled paired t test, this process of randomly splitting the data set S two-thirds/one-thirds was repeated 30 times. Each test set was classified as described in the previous paragraph. The 30 differences in the error rates of the two algorithms were collected and employed in the t test.

For the k-fold cross-validated paired t test, the data set S was divided into $k = 10$ random subsets of equal size. Each of the 10 test sets was then classified by both algorithms using the random procedure described above. However, during the classification process, to simulate random variation in the quality of each training set, we generated a random value β in the range $[-0.02, +0.02]$

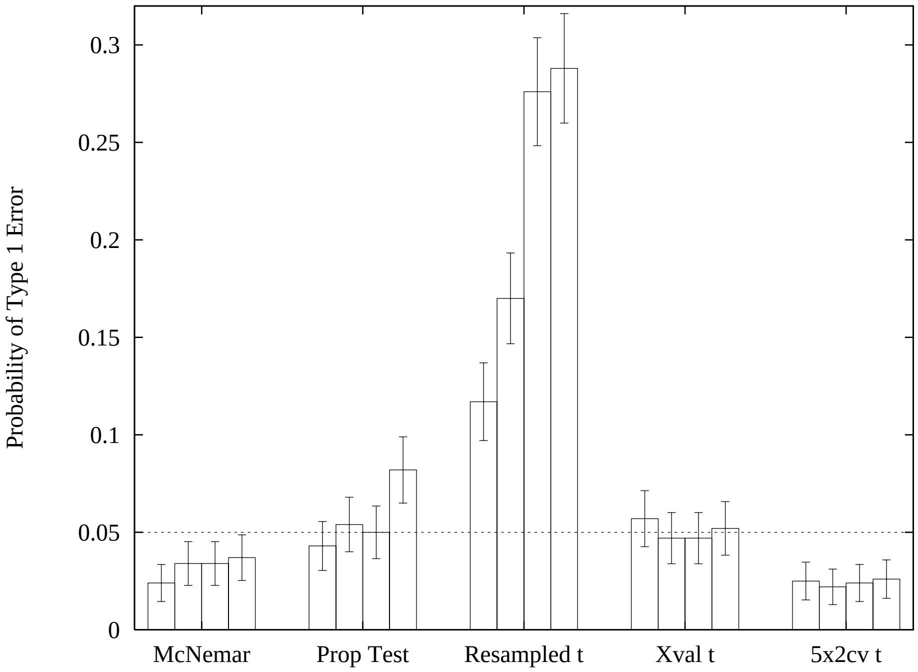

Figure 5: Probability of Type I error for each statistical test. The four adjacent bars for each test represent the probability of Type I error for $\epsilon = 0.10$, 0.20, 0.30, and 0.40. Error bars show 95% confidence intervals for these probabilities. The horizontal line shows the target probability of 0.05.

and added this value β to every $\epsilon_A(x)$ and $\epsilon_B(x)$ before generating the classifications. The results of the classifications were then collected and subjected to the t test.

For the 5x2cv test, five replications of 2-fold cross-validation were performed, and the t statistic was constructed as described above.

It is important to note that this experiment does not simulate training set variance. In particular, it does not model the effect of overlapping training sets on the behavior of the cross-validated t test or the 5x2cv test. We will correct this shortcoming below in our experiments with real learning algorithms.

5 Results

Figure 5 shows the probability of making a Type I error for each of the five procedures when the data set S contains 300 examples and the overall error rate ϵ was varied from 0.10 to 0.40.

Two of the five tests have a probability of Type I error that exceeds the target value of 0.05: the difference-of-proportions test and the resampled t test. The remaining tests have acceptable probability of making a Type I error according to this simulation.

The resampled t test has a much higher probability of Type I error than the other tests. This results from the fact that the randomly-drawn data set S is likely to contain an imbalance of points from the first half of the population compared to the second half of the population. The resampled t test can detect and magnify this difference until it is "statistically significant." Indeed, the probability of making a Type I error with this test can be increased by increasing the number of resampled training/test splits. Figure 6 shows the effect of various numbers of resampled splits

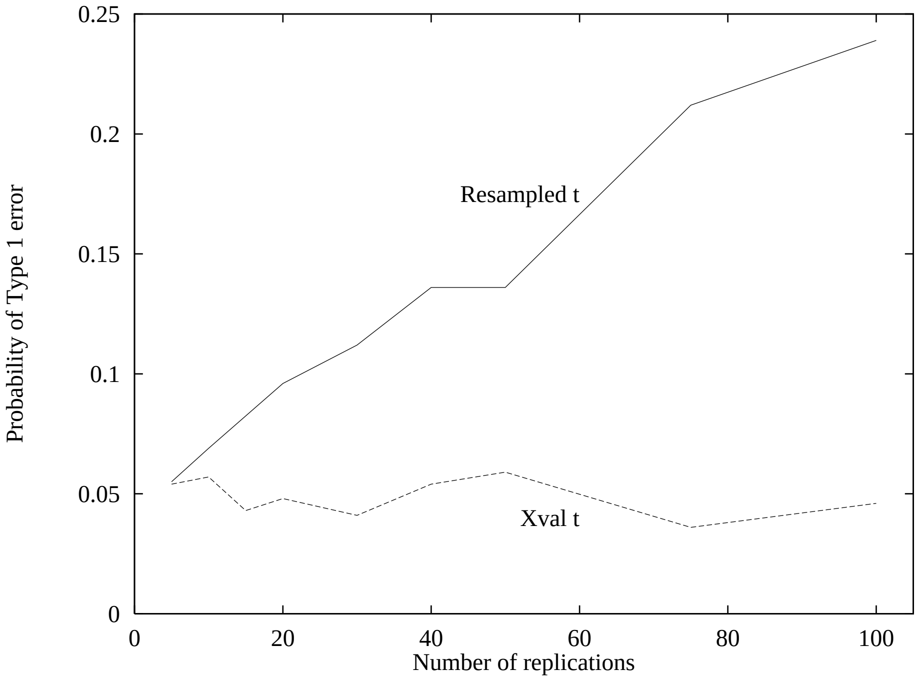

Figure 6: Probability of Type I error for the resampled t test and the k-fold cross-validated t test as the number of resampling replications is varied. The error rate $\epsilon = 0.10$.

for both the resampled t test and the cross-validated t test. Notice that the cross-validated t test does not exhibit this problem.

The difference-of-proportions test suffers from essentially the same problem. When the sample S is unrepresentative, the measured difference in the two proportions will be large, especially when ϵ is near 0.5. It is interesting that McNemar's test does not share this problem. The key difference is that the difference-of-proportions test only looks at the difference between two proportions and not at their absolute values. Consider the following two 2×2 contingency tables:

0	40
60	0

40	0
20	40

Both of these tables have the same difference in error rates of 0.20, so the difference-of-proportions test treats them identically (and rejects the null hypothesis $p < 0.005$). However, McNemar's test finds no significant difference in the left table, but finds an extremely significant difference ($p < 0.001$) in the right table. This is because in the left table, McNemar's test asks the question "What is the probability in 100 tosses of a fair coin that we will receive 40 heads and 60 tails?" In the right table, it asks the question "What is the probability in 20 tosses of a fair coin that we will receive 20 heads and 0 tails?" The way we have constructed our simulated learning problems, we are more likely to produce tables like the one on the left, especially when ϵ is near 0.5. Note that if we had used the corrected version of the difference-of-proportions test, it would not have suffered from this problem.

Because of the poor behavior (and high cost) of the resampled t test, we excluded it from further experiments.

The biggest drawback of the simulation is that it does not capture or measure training set variance or variance resulting from the internal behavior of the learning algorithm. To address these problems, we conducted a second set of experiments with real learning algorithms and real data sets.

6 Experiments on Realistic Data

6.1 Methods

To evaluate the Type I error rates of our four statistical tests with real learning algorithms, we needed to find two learning algorithms that had identical performance when trained on training sets of a given size. We also needed the learning algorithms to be very efficient, so that the experiments could be replicated many times. To achieve this, we chose C4.5 Release 1 (Quinlan, 1993) and the first nearest-neighbor (NN) algorithm (Dasarathy, 1991). We then selected three difficult problems: the EXP6 problem developed by Kong (1995), the Letter Recognition data set (Frey & Slate, 1991), and the Pima Indians Diabetes Task (Merz & Murphy, 1996). Of course, C4.5 and NN do not have the same performance on these data sets. In EXP6 and Letter Recognition, NN performs much better than C4.5; the reverse is true in the Pima data set.

Our next step was to "damage" the learning algorithms so that their performance was identical. In the EXP6 and Letter Recognition tasks, we modified the distance metric employed by NN to be a weighted Euclidean distance with bad weights. This allowed us to reduce the performance of NN until it matched C4.5 on those data sets. To equalize the performance of the algorithms on the Pima data set, we modified C4.5 so that when classifying new instances, it would make random classification errors at a specified rate.

More precisely, each data set was processed as follows. For EXP6, we generated a calibration set of 22,801 examples (spaced on a uniform grid of resolution 0.1). We then generated 1000 data sets, each of size 300, to simulate 1000 separate trials. From each of these 1000 data sets, we randomly drew subsets of size 270, 200, and 150. These sizes were chosen because they are the sizes of training sets used in the 10-fold cross-validated t test, the McNemar and difference-of-proportions tests, and the 5x2cv t test, respectively, when those tests are given an initial data set of 300 examples. For each size of training set (270, 200, and 150), we adjusted the distance metric for NN so that the average performance (over all 1000 data sets, measured on the 22,801 calibration examples) matched the average performance of C4.5 to within 0.1%.

For Letter Recognition, we randomly subdivided the 20,000 examples into a calibration set of 10,000 and an experimental set of 10,000. We then drew 1000 data sets, each of size 300, randomly from the experimental set of 10,000 examples. Again, from each of these data sets, we drew random subsets of size 270, 200, and 150. For each size of training set, we adjusted the distance metric for NN so that the average performance (over all 1000 data sets, measured on the 10,000 calibration examples) matched the average performance of C4.5 to within 0.1%.

For the Pima Indians Diabetes data set, we drew 1000 data sets of size 300 from the 768 available examples. For each of these data sets, the remaining 468 examples were retained for calibration. Each of the 1000 data sets of size 300 was further subsampled to produce random subsets of size 270, 200, and 150. For each size of training set, we measured the average error rate of C4.5 and NN (over 1000 data sets, when tested on the 468 calibration examples corresponding to each data set). We then adjusted the random noise rate of C4.5 so that the average error rates would be identical.

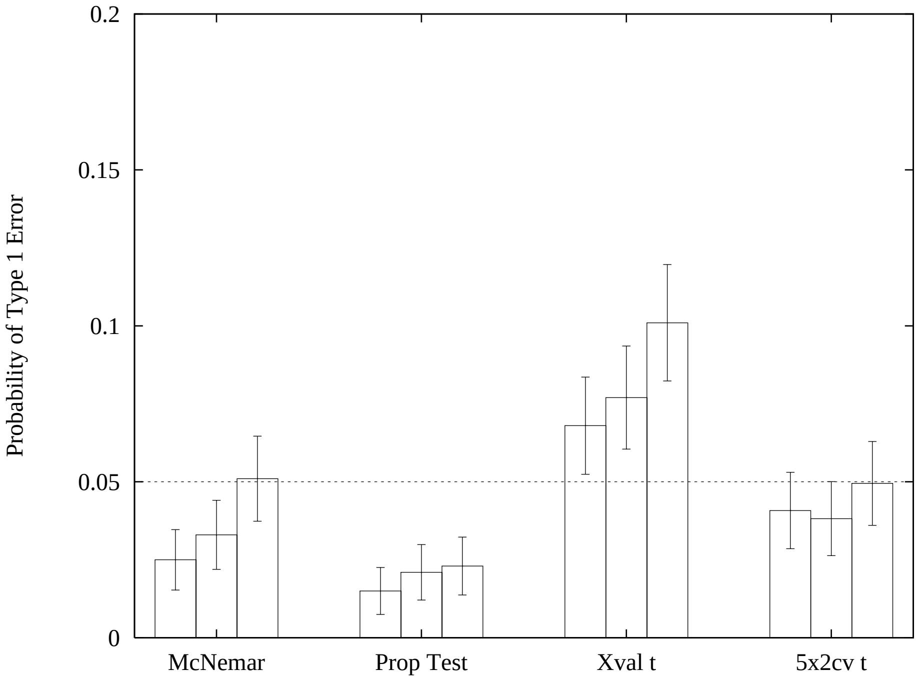

Figure 7: Type I error rates for four statistical tests. The three bars within each test correspond to the EXP6, Letter Recognition, and Pima data sets. Error bars are 95% confidence intervals on the true Type I error rate.

6.2 Type I Error Results

Figure 7 shows the measured Type I error rates of the four statistical tests. The 10-fold cross-validated t test is the only test whose Type I error exceeds 0.05. All of the other tests, even the difference-of-proportions test ("Prop Test"), show excellent Type I error rates.

6.3 Power Measurements

Type I error is not the only important consideration in choosing a statistical test. It is the most important criterion if one's goal is to be confident that an observed performance difference is real. But if one's goal is to determine that there is *no difference* between two learning algorithms, then the *power* of the statistical test is important. The power of a test is the probability that it will reject the null hypothesis when the null hypothesis is false.

To measure the power of the tests, we recalibrated the distance metric for nearest neighbor (for the EXP6 and Letter Recognition tasks) and the random classification error rate (for Pima) to achieve various differences in the performance of C4.5 and NN. Specifically, we did the following. First, we measured the performance for C4.5 and NN when trained on 300 examples and tested on the appropriate calibration examples as before (denote these error rates ϵ_{C4} and ϵ_{NN}). Then we chose various target error rates between these extremes. For each target error rate ϵ_{target}, we computed the fraction λ such that $\epsilon_{target} = \epsilon_{C4.5} - \lambda(\epsilon_{C4.5} - \epsilon_{NN})$. We then calibrated the error rates of C4.5 and NN for training sets of size 150, 200, and 270 so that the same value of λ applied. In other words, we adjusted the learning curve for the nearest neighbor algorithm (with damaged distance metric) so that it was positioned at a fixed fraction of the way between the learning curves

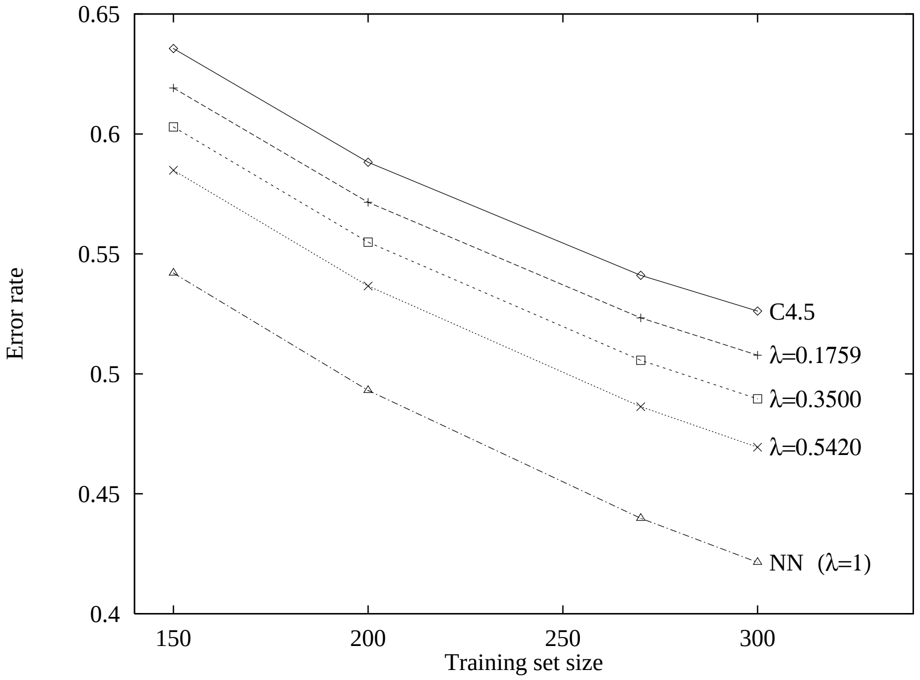

Figure 8: Learning curves interpolated between C4.5 and NN for the purpose of measuring power.

for C4.5 and for NN (with an undamaged distance metric). Figure 8 shows the calibrated learning curves for the Letter Recognition task for various values of λ.

Figures 9, 10, and 11 plot power curves for the four statistical tests. These curves show that the cross-validated t test is much more powerful than the other three tests. Hence, if the goal is to be confident that there is no difference between two algorithms, then the cross-validated t test is the test of choice, even though its Type I error is unacceptable.

Of the tests with acceptable Type I error, the 5x2cv t test is the most powerful. However, it is sobering to note that even when the performance of the learning algorithms differs by 10 percentage points (as in the Letter Recognition task), these statistical tests are only able to detect this about one third of the time.

7 Discussion

The experiments suggest that the 5x2cv test is the most powerful among those statistical tests that have acceptable Type I error. This test is also the most satisfying, because it assesses the effect of both the choice of training set (by running the learning algorithms on several different training sets) and the choice of test set (by measuring the performance on several test sets).

Despite the fact that McNemar's test does not assess the effect of varying the training sets, it still performs very well. Indeed, in all of our various experiments, we never once saw the Type I error rate of McNemar's test exceed the target level (0.05). In contrast, we did observe cases where both the 5x2cv and differences-of-proportions tests were fooled.

The 5x2cv test will fail in cases where the error rates measured in the various 2-fold cross-validation replications vary wildly (even when the *difference* in error rates is unchanged). We were able to observe this in some simulated data experiments where the error rates fluctuated between

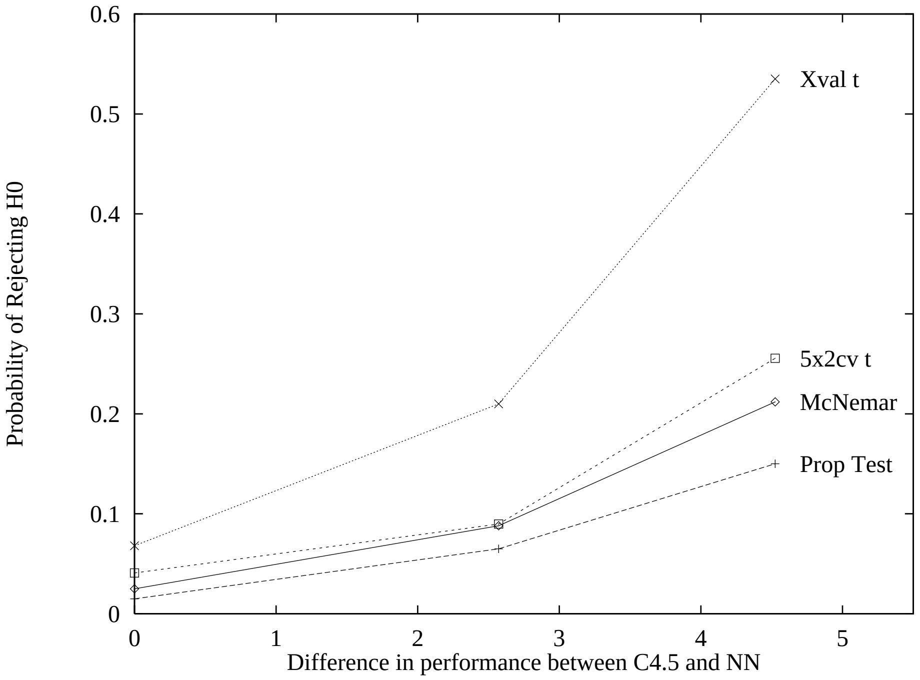

Figure 9: Power of four statistical tests on the EXP6 task. The horizontal axis plots the number of percentage points by which the two algorithms (C4.5 and NN) differ when trained on training sets of size 300.

0.1 and 0.9. We did not observe it during any of our experiments on realistic data. Wild variations cause bad estimates of the variance. It is therefore advisable to check the measured error rates when applying this test.

The difference-of-proportions test will fail in cases where the two learning algorithms have very different "regions" of poor performance and where the error rates are close to 0.5. We did not encounter this problem in our experiments on realistic data, even though C4.5 and NN are very different algorithms. We suspect that this is because most errors committed by learning algorithms are near the decision boundaries. Hence, most learning algorithms with comparable error rates have very similar regions of poor performance, so the pathology that we observed in our simulated data experiments does not arise in practice. Nonetheless, given the superior performance of McNemar's test and the incorrect assumptions underlying our version of the difference-of-proportions test, there can be no justification for ever employing the uncorrected difference-of-proportions test.

The 10-fold cross-validated t test has high Type I error. However, it also has high power, and hence, it can be recommended in those cases where Type II error (i.e., the failure to detect a real difference between algorithms) is more important.

8 Conclusions

The experiments in this paper lead us to recommend either the 5x2cv t test, for situations in which the learning algorithms are efficient enough to run ten times, or McNemar's test, for situations where the learning algorithms can be run only once. Both tests have similar power. This recommendation should be regarded as tentative, however, because our experiments are based on only two learning

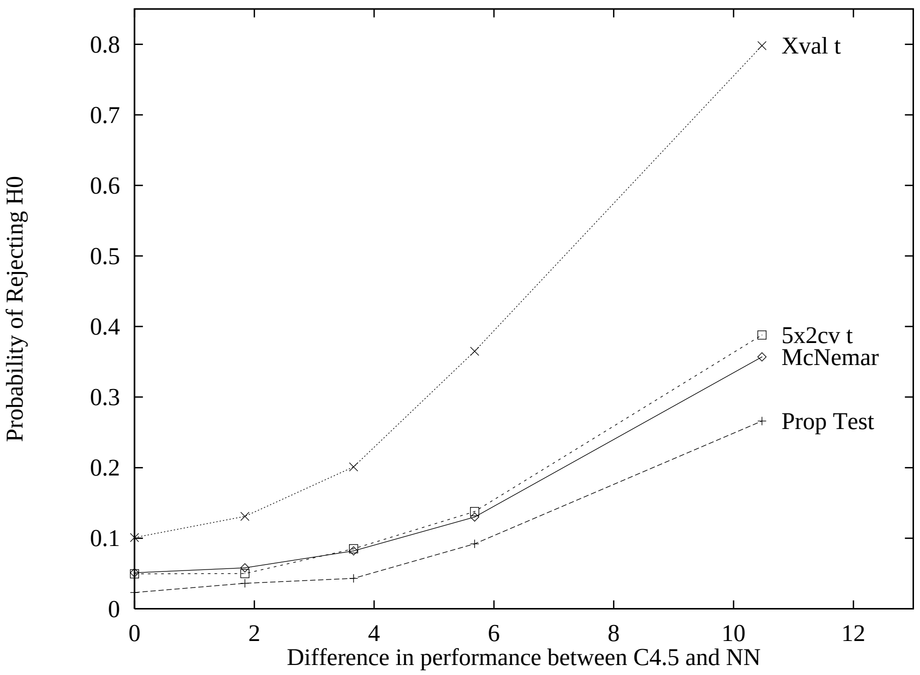

Figure 10: Power of four statistical tests on the Letter Recognition task. The horizontal axis plots the number of percentage points by which the two algorithms (C4.5 and NN) differ when trained on training sets of size 300.

algorithms and three data sets.

However, our experiments have revealed the causes of the shortcomings of the other statistical tests, so we can certainly recommend that the resampled t test should never be employed. This test has very high probability of Type 1 error, and results obtained using this test cannot be trusted. The experiments also suggest caution in interpreting the results of the 10-fold cross-validated t test. This test has a elevated probability of Type I error (as much as twice the target level), although it is not nearly as severe as the problem with the resampled t test.

We hope that the results in this paper will be useful to scientists in the machine learning and neural network communities as they develop, understand, and improve machine learning algorithms.

9 Acknowledgements

The author first learned of the k-fold cross-validated t test from Ronny Kohavi (personal communication). The author thanks Jim Kolsky of the OSU Statistics Department for statistical consulting assistance and Radford Neal, Ronny Kohavi, and Tom Mitchell for helpful suggestions. The author is also very grateful to the referees for their careful reading, which identified several errors in an earlier version of this manuscript. This research was supported by grants IRI-9204129 and IRI-9626584 from the National Science Foundation and grant number N00014-95-1-0557 from the Office of Naval Research.

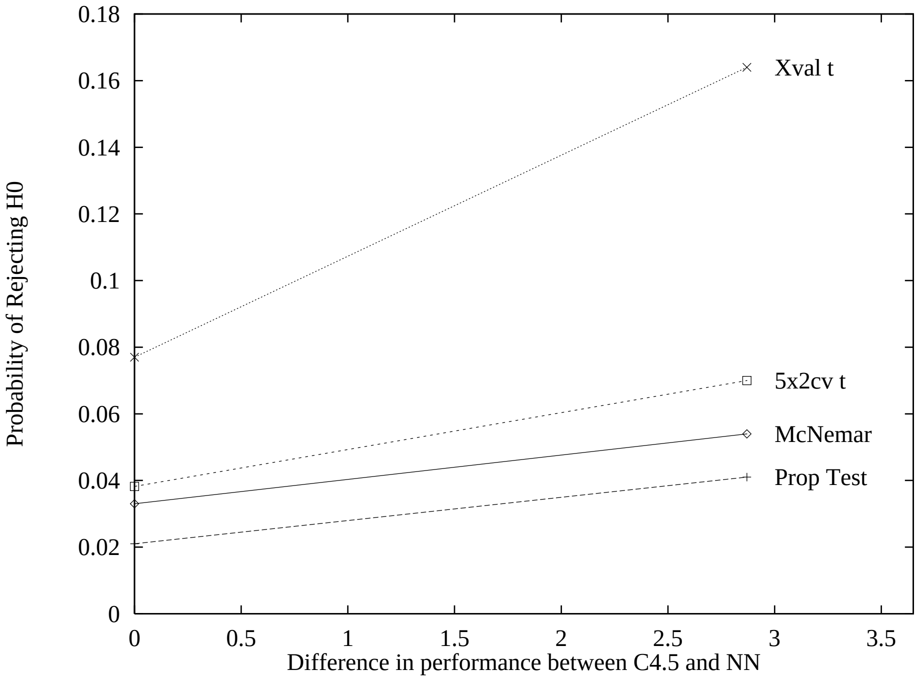

Figure 11: Power of four statistical tests on the Pima task. The horizontal axis plots the number of percentage points by which the two algorithms (C4.5 and NN) differ when trained on training sets of size 300.

References

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth International Group.

Breiman, L. (1994). Heuristics of instability and stabilization in model selection. Tech. rep. 416, Department of Statistics, University of California, Berkeley, CA.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Dasarathy, B. V. (Ed.). (1991). *Nearest neighbor (NN) norms: NN pattern classification techniques.* IEEE Computer Society Press, Los Alamitos, CA.

Dietterich, T. G., Hild, H., & Bakiri, G. (1995). A comparison of ID3 and backpropagation for English text-to-speech mapping. *Machine Learning*, *18*, 51–80.

Efron, B., & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap.* Chapman and Hall, New York, NY.

Everitt, B. S. (1977). *The analysis of contingency tables.* Chapman and Hall, London.

Frey, P. W., & Slate, D. J. (1991). Letter recognition using holland-style adaptive classifiers. *Machine Learning*, *6*, 161–182.

Haussler, D., Kearns, M., Seung, H. S., & Tishby, N. (1994). Rigorous learning curve bounds from statistical mechanics. In *Proc. 7th Annu. ACM Workshop on Comput. Learning Theory*, pp. 76–87. ACM Press, New York, NY.

Hinton, G. E., Neal, R. M., Tibshirani, R., & DELVE team members (1995). Assessing learning procedures using DELVE. Tech. rep., University of Toronto, Department of Computer Science, http://www.cs.utoronto.ca/neuron/delve/delve.html.

Kohavi, R. (1995). *Wrappers for performance enhancement and oblivious decision graphs.* Ph.D. thesis, Stanford University.

Kolen, J. F., & Pollack, J. B. (1991). Back propagation is sensitive to initial conditions. In *Advances in Neural Information Processing Systems*, Vol. 3, pp. 860–867 San Francisco, CA. Morgan Kaufmann.

Kong, E. B., & Dietterich, T. G. (1995). Error-correcting output coding corrects bias and variance. In Prieditis, A., & Russell, S. (Eds.), *The Twelfth International Conference on Machine Learning*, pp. 313–321 San Francisco, CA. Morgan Kaufmann.

Lindman, H. R. (1992). *Analysis of Variance in Experimental Design.* Springer Verlag, New York.

Merz, C. J., & Murphy, P. M. (1996). UCI repository of machine learning databases. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Quinlan, J. R. (1993). *C4.5: Programs for Empirical Learning.* Morgan Kaufmann, San Francisco, CA.

Rasmussen, C. E. (1996). *Evaluation of Gaussian processes and other methods for non-linear regression.* Ph.D. thesis, University of Toronto, Department of Computer Science, Toronto, Canada.

Snedecor, G. W., & Cochran, W. G. (1989). *Statistical Methods.* Iowa State University Press, Ames, IA. Eighth Edition.