Using Permutations Instead of Student's t Distribution for p-values in Paired-Difference Algorithm Comparisons

Joshua Menke and Tony R. Martinez
Computer Science Department
Brigham Young University
Provo, UT, 84602
E-mail: josh@axon.cs.byu.edu, martinez@cs.byu.edu

Abstract— **The paired-difference t-test is commonly used in the machine learning community to determine whether one learning algorithm is better than another on a given learning task. This paper suggests the use of the permutation test instead because it calculates the exact p-value instead of an estimate. The permutation test is also distribution free and the time complexity is trivial for the commonly used 10-fold cross-validation paired-difference test. Results of experiments on real-world problems suggest it is not uncommon to see the t-test estimate deviate up to 30-50% from the exact p-value.**

I. Introduction

A popular test used to compare two learning algorithms is the 10-fold cross-validation paired-difference t-test [1], [2]. This test uses Student's t distribution to estimate a p-value representing the probability that the mean of the differences observed occurred randomly. If the resulting p-value is very low (usually below 0.10) it can be concluded that the observed difference is more than can be explained by random chance, and is therefore statistically significant. In the context of machine learning, this amounts to comparing how well algorithm A does compared to algorithm B on a particular learning problem characterized by a data set D. If the mean difference between algorithm A's performance and algorithm B's performance using data set D is statistically significant, there is support to prefer using algorithm A for that particular learning problem. For this reason, tests comparing two or more algorithms are important in validating the utility of machine learning algorithms and comparing them to each other in different problem domains.

The problem with the t-test is that it does not yield the exact p-value, but instead an approximation based on assumptions about the distribution of a paired-difference test. It is possible, however, to calculate the exact p-value in a trivial amount of time for the common 10-fold version of the paired-difference test using a permutation test. Since this test yields a more accurate p-value and is easily calculated, this paper suggests using the permutation test instead of approximating the p-value with Student's t distribution. The results of experiments on several real world problems show it is not uncommon for the t-test approximation to deviate as much as 30-50% from the true p-value.

II. Background

A. Statistical Issues

There are two assumptions associated with using Student's t distribution to calculate the p value:

1) The differences are normally distributed.
2) The differences are independent.

Since the t-test is robust to the first assumption, it often yields a reasonable approximation to the the true p-value which, in a paired-difference test, is:

$$p = \frac{n}{N} \tag{1}$$

where n is the number of ways the mean difference can be as extreme or more extreme (for a two-sided p-value) than the observed mean difference and N is the total number of possible reassignments of the paired-differences given the results. It is a measure of how often it is expected that a difference as extreme or more extreme than the observed difference occurs randomly. The p-value can also be calculated exactly in $O(2^n)$ time where n is the number of pairs. Although exponential, for the small amount of pairs usually used in the literature ($k = 10$), calculating the p-value exactly is not unreasonable. In fact, in statistics the calculation of the exact p-value is known as a permutation test and is often available in popular statistical packages.

Tables I–III show a simple example of how to calculate the p-value using a permutation test. Consider permutation 1 to be the original results of an experiment involving two algorithms being compared with, in this simple example, a 3-fold cross validation paired-difference test. Table I uses 3 of the possible 8 permutations to show how different permutations are created by swapping the results between algorithms. Notice the only difference between permutation 1 and permutation 2 in table I is that the first row or fold results have been swapped. In the same table, the difference between permutations 1 and 3 is that in 3, both the first and second row's results have been swapped. Notice that swapping the results simply causes the sign on the difference to change, therefore, table II can show the 8 possible permutations giving only the change in the differences for each fold. For example, notice that the difference between permutations 1 and 5 in table II is that the sign on the third difference has changed, meaning the results for the third fold have been swapped. To obtain the p-value, the number of mean differences as extreme or more extreme than the observed data are counted up and used as n in table III. In this case, assuming permutation 1 represents the original results, the number of permutations that yield a mean greater than or equal to 0.0012 or less than or equal to -0.0012 are counted (for a two-sided p-value). The result includes permutations 1 (the original), 4, 5, and 8 for a total of 4. Finally, dividing 4 by the total number of possible permutations, 2^3 or 8 yields the p-value, 0.5 as shown in table III. Since the p-value is high in this case, the mean difference observed is not considered statistically significant. If the p-value had been below 0.10, the difference would have been considered significant.

TABLE I
PERMUTATION TEST EXAMPLE PART 1

Algorithm A	Algorithm B	Difference
Permutation 1		
0.9330	0.9309	0.0021
0.9336	0.9315	0.0021
0.9302	0.9308	-0.0006
Mean		0.0012
Permutation 2		
0.9309	0.9330	-0.0021
0.9336	0.9315	0.0021
0.9302	0.9308	-0.0006
Mean		0.0002
Permutation 4		
0.9309	0.9330	-0.0021
0.9315	0.9336	-0.0021
0.9302	0.9308	-0.0006
Mean		-0.0016

TABLE II
PERMUTATION TEST EXAMPLE PART 2

Permutation	Difference	Permutation	Difference
1	0.0021	5	0.0021
	0.0021		0.0021
	-0.0006		0.0006
Mean	0.0012	Mean	0.0016
2	-0.0021	6	-0.0021
	0.0021		0.0021
	-0.0006		0.0006
Mean	-0.0002	Mean	0.0002
3	0.0021	7	0.0021
	-0.0021		-0.0021
	-0.0006		0.0006
Mean	-0.0002	Mean	0.0002
4	-0.0021	8	-0.0021
	-0.0021		-0.0021
	-0.0006		0.0006
Mean	-0.0016	Mean	-0.0012

TABLE III
PERMUTATION TEST EXAMPLE PART 3

n	N	p-value ($\frac{n}{N}$)
4	8	0.5

B. Past Research

Since being able to compare machine learning algorithms is one of the key elements in the research area, there have been several evaluations of popular practices, and suggestions for appropriate procedures. In [2], Reich evaluates common practices for comparing machine learning methods and suggests appropriate practices, including the use of the 10-fold cross-validation paired-difference t-test. Salzberg criticizes mainstream philosophies and statistical methods in machine learning in [3], especially when using statistics to compare algorithms. He also criticizes the use of Student's t-test in re-sampled approaches because the assumption of independence between samples is violated. Dietterich [1] supports Salzberg's criticism of the re-sampled paired t-test and also warns against the use of the 10-fold cross-validation paired-difference t-test. He offers a new t-test seeking to retain the power or ability of the 10-fold test to detect existing differences while improving on its error or tendency to detect non-existent differences. Dietterich also acknowledges that like the 10-fold test, his suggested test still violates the independence assumption. Kohavi [4] suggests a stratified approach to cross-validation that lessens the effect of non-independence by forcing the folds to retain the same distribution as the original data and yields more accurate p-values. In [5], Michaels further supports the use of stratified approaches because they represent the common "multi-modality" of classification error. He then suggests a unique method using replicate statistics that does not assume independence and results in improved confidence intervals that can be calculated efficiently.

Moving away from the usual paired approaches, [6] introduces a randomized ANOVA approach for comparing algorithms. In [7], Provost et. al. argue against the traditionally used classification accuracy for comparison and promote ROC analysis in its place. [8] gives an approach to using the area underneath ROC curves for evaluating machine learning algorithms. [9] shows the usage of ROC curves with artificial neural networks. In, [10], Maloof explains that ROC approaches employ analysis of variance methods (ANOVA) to determine if the results of the ROC analysis are statistically significant. He then empirically compares the standard ROC ANOVA approach to the LABMRMC method [11] used commonly in the medical decision making community. He finds that the standard ROC ANOVA approach and the LABMRMC method can make different decisions as to the preferred learning algorithm and recommends using LABMRMC-type techniques because they more accurately model the assumptions in a cross-validation experiment.

Although the 10-fold approach has received criticism in the above research, this paper focuses on using 10-fold cross-validation not because it is the best method, only because it is very common. Here, it is suggested that if 10-fold cross-validation is to be used anyways, the exact p-value might as well be calculated instead of an approximation.

III. Motivation

The most convincing reason to choose the permutation test instead of the t-test is because the p-value will be exact instead of approximated, thus yielding a more accurate prediction of how random a given result is.

There are two theoretical reasons for choosing one statistical test over another:

1) It is less likely to detect a difference when there is none.
2) It is less likely to miss a difference when there is one.

Since the permutation test yields the exact p-value, it will always, by definition, be less likely than the t-test to detect a difference where there is none or miss a difference where there is one.

Two practical considerations when choosing one statistical test over another are:

1) Is one statistic easier to calculate than the other in terms of time and space requirements?
2) Does the resulting conclusion change significantly enough in practice to choose one over the other?

Since the number differences usually used in a paired-difference test is 10, the amount of time and space is relatively small with only 1,024 permutations to evaluate. The far from optimized implementation used for the experiments in section V runs in just over a tenth of a second. Therefore, the unanswered question is whether or not the p-values can differ enough in practice to promote the use of the permutation test. The rest of the paper seeks to answer this question giving both methods of calculating the p-value in section IV, then descriptions of experiments comparing both methods in section V. Section VI gives the experimental results and discusses them and section VII contains the conclusion and suggestions for further research.

IV. Methods

The 10-fold cross-validated paired-difference test is used here to explain how to calculate both the t-test's approximation to the p-value and the exact p-value. The technique here can be used for a k-fold experiment or even for a re-sampled paired t-test (criticized in [1], [3]), although the complexity grows exponentially in k or the number of times the data is re-sampled. The cross-validated paired-difference test is chosen in particular for its popularity among machine learning researchers (see [2]). The idea is if 10 differences are already calculated, the amount of work to determine the exact p-value is trivial.

A. Run the Experiments

The first part of a 10-fold cross-validation experiment is to obtain results for each fold. Each fold usually consists of two sub-experiments which can be paired because they vary only in the treatment in question. For example, two equivalent *artificial neural networks* (ANNs) trained on the same data in the same order, but with different random initial weight settings. Each ANN is trained and then tested and the difference in accuracy between the two is saved. The details behind selecting the training and testing sets for k-fold cross-validation can be found in [1] and [2].

B. Calculating p from t

After calculating the difference in accuracy between each of the 10 folds, the 10 resulting differences can be used to calculate either the t-test or exact p-value. First, to approximate the p-value, the t-statistic is calculated using the 10 differences:

$$t = \frac{\bar{x}}{SE_x} \tag{2}$$

where $\bar{x}$ is the mean of k differences where k is the number of differences—10 in this case—and SE_x or the standard error of x is

$$SE_x = \frac{\sigma_x}{\sqrt{k}} \tag{3}$$

where σ_x is the standard deviation of the k differences. Calculated this way, t is known to follow Student's t distribution with $k - 1$ degrees of freedom.

C. Calculating p exactly

In order to calculate the exact p-value, the mean differences of all possible assignments of the given results must be evaluated. Evaluating all possible assignments involves calculating the mean difference of all possible reassignments of the group membership on each fold as in the example from section II-A. The process explained in section II-A can be restated as the pseudo-code in figure 1.

```
1)  given 10 differences
2)  mean difference $\leftarrow$ mean(the ten differences)
3)  $n \leftarrow 0$
4)  $N \leftarrow 2^{10} = 1024$
5)  repeat
6)      get next permutation
7)      if mean(abs(permutation differences)) $\geq$ abs(mean difference)
8)          $n \leftarrow n + 1$
9)  until all permutations tried
10) $p - value \leftarrow \frac{n}{N}$
```

Fig. 1. One way to calculate the exact p value using a permutation test.

A given permutation can be characterized by whether or not each fold has its signed changed. This can be implemented as a binary number where each binary digit corresponds to whether or not one of the differences has its sign changed. For 10 folds, this yields all 10-digit binary numbers or 1,024 permutations. Trying each permutation is equivalent to counting to 1,024, calculating and comparing the mean differences for each permutation. The binary number can be mapped onto the differences using a mask, changing line 6 in figure 1 to the pseudo-code in figure 2. Therefore, calculating the 1,024 differences is done easily and in a trivial amount of time using the given approach.

```
for permutation = 1 to 1024
    for pair = 1 to 10
        if (permutation (bit AND) $2^{pair}$) $= 1$
            difference[pair] = -difference[pair]
```

Fig. 2. One way to enumerate all permutations given 10 pairs of differences.

V. Experiment

To determine how well the p values generated by the permutation test compare with those resulting from a standard t-test, 10-fold cross-validation paired tests are conducted comparing the accuracy of two feed-forward multilayer perceptron (artificial neural networks or ANN) classifiers each with a single hidden layer. The data sets used are described in table IV. The table also includes information about the number of hidden nodes of the ANNs used in each experiment. Every ANN used a learning rate of 0.05. The training set for each fold is split into a training and hold-out set. The ANNs are trained on only the training partition, and then tested after each iteration on the hold-out partition. Training ceases when there has not been an increase in accuracy on the hold-out set for a period of 100 iterations. The ANN weight configuration resulting in the highest hold-out set accuracy is then tested on the test partition of the fold and that number is used as the final result to be compared with the other ANN in the the same fold. The only variation within each pair is different random initial weight settings, and therefore the resultant p-values should be relatively large as there is no significant difference between the ANNs. The only other source of variation is the difference in each of the 10 folds which is part of the design of the 10-fold cross-validation paired-difference test (as explained in [1] and [2].

TABLE IV

Data sets used in the experiments.

Data set	size	features	classes	nodes
OCR	500,000	64	83	32
iris	150	4	3	5
pima	768	8	2	5
letter-recognition	20,000	256	26	32
wave21	300	21	3	5
bupa	345	6	2	5
glass	214	9	7	5

VI. Results and Discussion

The resulting p-values are shown in table V. The p-values are the same or close for the iris and OCR data sets, but up to 50% different for the others. Although the conclusions would still be the same for both statistics in these cases, if the question of significance was closer than in the experiment, the t-test would be more likely to incorrectly detect a difference where there was none or miss a true existing difference. For example, if the relative difference remains at around 30% between the t-test and permutation test p-values, a true p-value of 0.1429—not usually held as significant—could yield a t-test p-value of 0.10, often held as significant in the literature. Since the results suggests the t-test p-value can deviate as much as 30-50% from the exact p-value, using the permutation test will result in finding less false significant values and missing less truly significant results.

TABLE V

t-test and permutation test p values on several data sets.

Data set	t	exact	difference	relative difference
OCR	0.9847	0.9746	0.0101	1%
iris	1.0	1.0	—	—
pima	0.6783	1.0	-0.3217	32%
letter-recognition	0.0117	0.0176	-0.0059	34%
wave21	0.8534	0.7852	0.0682	9%
bupa	0.2091	0.5000	-0.2909	58%
glass	0.2443	0.5000	-0.2557	51%

VII. Conclusions

The permutation test is suggested for use in place of the standard t-test for small (10 pairs) paired-difference cross-validation tests because it is more accurate and the t-test approximation can deviate significantly from the true p-value. A method of computing the permutation test p-value is given. The complexity of the permutation test is reasonable for small numbers of pairs and the resulting p-value is more precise and therefore less likely to detect significance where there is none, and more likely to detect significance if present.

Future work will investigate the theoretically expected difference between permutation and t-test calculated p-values. Also, a more thorough test comparing the error and power of the permutation test relative to the t-test is appropriate. 10-fold cross-validation is criticized as having a high error in [1]. It is interesting to investigate how much of this error can be decreased through exact vs. approximate calculation of the resulting p-values, despite the fact that the error is more likely to come from the violated independence assumptions as discussed in [1], [3], [5].

References

[1] T. G. Dietterich, "Approximate statistical test for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, 1998. [Online]. Available: citeseer.nj.nec.com/dietterich98approximate.html

[2] Y. Reich and S. Barai, "Evaluating machine learning models for engineering problems," 1999. [Online]. Available: citeseer.nj.nec.com/reich99evaluating.html

[3] S. Salzberg, "On comparing classifiers: Pitfalls to avoid and a recommended approach," *Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 317–328, 1997. [Online]. Available: citeseer.nj.nec.com/salzberg97comparing.html

[4] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," 1995. [Online]. Available: gunther.smeal.psu.edu/kohavi95study.html

[5] R. Micheals and T. Boult, "Efficient evaluation of classification and recognition systems," 2001. [Online]. Available: citeseer.nj.nec.com/article/micheals01efficient.html

[6] J. H. Piater, P. R. Cohen, X. Zhang, and M. Atighetchi, “A randomized ANOVA procedure for comparing performance curves,” in *Proc. 15th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1998, pp. 430–438. [Online]. Available: citeseer.nj.nec.com/67061.html

[7] F. Provost, T. Fawcett, and R. Kohavi, “The case against accuracy estimation for comparing induction algorithms,” in *Proc. 15th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 1998, pp. 445–453. [Online]. Available: citeseer.nj.nec.com/article/provost98case.html

[8] A. Bradley, “The use of the area under the roc curve in the evaluation of machine learning algorithms,” *PR*, vol. 30, no. 7, pp. 1145–1159, July 1997.

[9] K. Woods and K. Bowyer, “Generating roc curves for artificial neural networks,” *IEEE Transactions on Medical Imaging*.

[10] M. Maloof, “On machine learning, ROC analysis, and statistical tests of significance,” in *Proceedings of the Sixteenth International Conference on Pattern Recognition*. Los Alamitos, CA: IEEE Press, 2002, pp. 204–207.

[11] K. B. D. Dorfman and C. Metz, “Receiver operator characteristics rating analysis: Generalization to the populations of readers and patients with the jackknife method.” *Investigative Radiology*.