Beware the Null Hypothesis: Critical Value Tables for Evaluating Classifiers

George Forman and Ira Cohen

Hewlett-Packard Labs,1501 Page Mill Rd,
Palo Alto, CA 94304, USA
ghforman, icohen@hpl.hp.com

Abstract. Scientists regularly decide the statistical significance of their findings by determining whether they can, with sufficient confidence, rule out the possibility that their findings could be attributed to random variation—the 'null hypothesis.' For this, they rely on tables with *critical values* pre-computed for the normal distribution, the t-distribution, etc. This paper provides such tables (and methods for generating them) for the performance metrics of binary classification: accuracy, F-measure, area under the ROC curve (AUC), and true positives in the top ten. Given a test set of a certain size, the tables provide the critical value for accepting or rejecting the null hypothesis that the score of the best classifier would be consistent with taking the best of a set of random classifiers. The tables are appropriate to consult when a researcher, practitioner or contest manager selects the best of many classifiers measured against a common test set. The risk of the null hypothesis is especially high when there is a shortage of positives or negatives in the testing set (irrespective of the training set size), as is the case for many medical and industrial classification tasks with highly skewed class distributions.

1 Introduction

Much practice and research work in the field of data mining amounts to trying a number of models or parameterizations, and selecting or recommending the best based on the performance scores on a test set. Sometimes the difference in performance of the top two scoring methods is not statistically significant according to standard statistical tests, in which case one is usually satisfied that either is a good choice. Unfortunately, even this conclusion may be suspect if the test set is too small, and it may not be obvious how small is 'too small.' Often it is difficult or expensive to obtain additional validated test data, as in many medical or industrial classification tasks. Furthermore, even a large test set can yield insignificant conclusions if the number of positives or negatives is unsuited for the particular performance metric.

As the number of competing models grows, the performance level required for statistical significance may be surprisingly large. For example, the organizers of the 2001 KDD Cup provided an interesting real-world biology classification challenge with a respectably large test set (150 positives and 484 negatives). However, the winning score of the 114 contestants was later found to be no greater than one should expect from 114 randomly generated trivial classifiers [4]. Consider also well-known datasets, such as the Wisconsin Breast cancer dataset (241 positive/malignant and 458 negative/benign), to which many researchers have applied a variety of learning

J. Gama et al. (Eds.): ECML 2005, LNAI 3720, pp. 133 – 145, 2005.

models and published the best of these [7]. Other examples abound. Examining the datasets contributed to the UCI machine learning repository [2], 45 out of the 69 datasets contain less than 2000 samples for both training and testing. Of these 45 datasets, many were collected in real world medical experiments, which further raises the importance of determining the statistical significance of the results. Medical researchers regularly attempt to evaluate classifiers with fewer than 100 patients [1].

Table 1. Summary of conditions

α = 0.01	significance level: 1% chance of failing to reject the null hypothesis
C = 10;100;1000	number of competing classifiers
P = 2..1000	positives in test set
N = 2..1000	negatives in test set
Performance metrics:	
AUC	area under the ROC curve (true- vs. false-positives)
TP10	true positives in top 10
Accuracy	percent correct (= 1 – error rate)
F-measure	2 × Precision × Recall ÷ (Precision + Recall) (harmonic average of precision and recall)

Practitioners and researchers need a convenient method or statistical reference table in order to determine whether the selection of the best classifier based on its winning score on their limited test set is statistically significant—that is, ruling out with sufficient probability that the best score found could have been obtained without substantial learning. (Note this differs from common pair-wise testing to determine whether the scores of one method are significantly better than the scores of another method—which is blind to the possibility that both are excellent or both terrible.) This paper lays out explicit significance tests for binary classification performance metrics based on the standard statistical method of rejecting the null hypothesis with high probability. We also provide reference charts showing the critical value for various test set sizes, to evaluate the significance of one's 'best' classifier.

The critical value depends on the number of positives and negatives in the test set, irrespective of the size of the training set, and applies to both cross-validation studies and held-out test sets. We develop the method and tables for each of the following four performance metrics: accuracy, F-measure, area under the ROC curve (AUC), and number of positives identified in the top ten cases predicted to be positive (TP10). Table 1 summarizes the range of conditions for which we offer pre-computed results.

Furthermore, with a qualitative understanding of these results, one can more intelligently select the distribution of positives and negatives in future test sets and/or select the performance metrics appropriate to a given test set.

Sections 2 and 3 lay out the statistical foundation and define the null hypothesis for four different performance metrics, one of which is computed analytically. Section 4 presents the critical value charts, with additional detail provided in tables in the appendix, which is only available in the online version of this paper [5].

2 Statistics Background

Generally, to determine whether an apparent measured difference in performance is *statistically significant*, one must consider the probability that the same measured result would occur under the *null hypothesis*—the hypothesis that the difference is simply due to natural random variation and not due to true differences in the methods. To decide this, one must establish an acceptable level of risk that one will mistakenly *fail to reject the null hypothesis*. This is characterized as the level of significance α, and is usually chosen to be 0.01. That is, a result is reported to be statistically significant if its probability of occurring by chance under the null hypothesis is less than 1%. Given α, one can determine the region of values of the test statistic where one can safely reject the null hypothesis. The statistical test in our case has the form 'reject the null hypothesis if m > m*,' where m is the maximum test score of the classifiers. The value m* here is called the *critical value* and is defined as $F(m^*) = (1-\alpha)$, where F(x) is the *cumulative distribution function* (*CDF*) of the test statistic under the null hypothesis, i.e. F(x) equals the probability that a random sample drawn from the distribution under the null hypothesis is less than or equal to x.

Table 2. Significance test for competing classifiers

Input:	C:	number of competing classifiers
	m:	maximum score by the winner
	P,N:	positives and negatives in test set
	α:	significance level, conventionally 0.01

For $R = 1000 \div (1 - (1 - \alpha)^{1/C})$ repetitions:
  | Randomly shuffle P 1's and N 0's in an array
  | Score this ordering by the desired performance metric
  | Keep track of the top 1000 scores in a priority queue/heap
m^* = the 1000^{th} best score retained, i.e. $F^{-1}((1-\alpha)^{1/C})$.
Decide statistically significant iff $m > m^*$

Given a competition among C=10 competitors where the winner achieves a maximum score m under some performance measure, one determines whether this result is statistically significant as follows: Consider the null hypothesis that each competitor performs its task in a trivial and random fashion. Determine the distribution of scores one competitor would expect to achieve under the null hypothesis. This establishes a CDF F(x) for a single competitor. We assume under the null hypothesis that the scores of the C competitors are drawn independently and identically distributed (iid) from this distribution. Given that the maximum score is m in the true competition, the probability under the null hypothesis that all of the C independent competitors would score $\leq$ m is $F(m)^C$. Given this joint CDF, we solve the equation given in the previous paragraph to determine the critical value m*:

$$F(m^*)^C = (1 - \alpha)$$
$$F(m^*) = (1-\alpha)^{1/C} = (1-0.01)^{1/10} = 0.99^{0.1} = 99.8995^{th} \text{ percentile}$$

If the maximum score m exceeds this critical value m*, then we can safely reject the null hypothesis. (Alternately, one may report the p-value for the maximum m.) All that remains is to determine the inverse CDF value $F^{-1}(0.998995)$ for the given

performance measure for a single competitor under the null hypothesis. We instantiate this for four binary classification performance measures in the next section.

3 Null Hypothesis for Classifiers

A classifier under the null hypothesis learns nothing whatsoever from the training set. Its ranking of the test set amounts to random shuffling of the positives and negatives.

Given the arbitrary ranking of the test cases by the classifier under the null hypothesis, the AUC score and the TP10 score are computed just as if a well-trained classifier generated the ranking. TP10 performance simply measures the number of true positives identified in the first ten positions of the ranking—a precision measure commonly used in information retrieval benchmarks. The AUC score measures the area under the ROC curve (x-axis = false positive rate, y-axis = true positive rate). We walk the array incrementally, counting the number of true positives and false positives collected as we adjust a hypothetical threshold to encompass a growing prefix of the array. See [3] for explicit AUC subroutines and useful guidelines on ROC curves.

In order to determine accuracy, a specific threshold is required on the ROC curve, indicating that all cases above this threshold are predicted positive, and the rest negative. One choice is to also select the threshold randomly, but this would rarely perform as well as majority voting. Instead, we take a conservative approach and select the threshold along the randomly generated ROC curve having the greatest accuracy. We call this measure the *best accuracy* under the null hypothesis. This choice of definition is conservative in that if a given competition exceeds the critical value for *best* accuracy, it surely exceeds the critical value for a less optimally chosen threshold. While some may suggest this might be too conservative for their liking, we would be uncomfortable promoting a classifier that does not exceed this null hypothesis. (Naturally, the popular *error rate* metric is equal to (1-accuracy), thus our definition provides a natural equivalent for *best error rate*.)

The same applies for F-measure: to measure the 'best F-measure' achieved by the random ranking, we walk along the ROC curve, counting true-positives and false-positives to determine the precision and recall at each point, and record the maximum F-measure achieved over the entire curve. F-measure is a popular metric in information retrieval settings, where the class of interest is in such a minority as to give majority voting a very high accuracy. For example, to select 30 relevant (positive) articles out of a database of 1030, majority voting achieves an accuracy of 97% by predicting the negative class for all test cases. This trivial strategy gives zero recall to the positive class, and so achieves zero F-measure. A high F-measure is achievable only by balancing high precision with high recall.

Given the statistical machinery described in the previous section, the definition of the null hypothesis and the description of how to score each of the four performance metrics, we give in Table 2 an explicit procedure for determining the statistical significance that the winner of a competition of C binary classifiers achieves score m on a test set comprising P positives and N negatives. The loop determines the required inverse CDF value empirically. We note that the empirical method presented in Table 2 is general for any performance metric and can be used without a need for analytical knowledge of the statistics behind a particular metric.

Figure 1 shows the entire CDF for three of the performance metrics over a test set with P=10 positives and N=1000 negatives. (TP10 is not shown because its x-axis spans 0 to 10.) The horizontal line near the top indicates the 99th percentile, above which each of the CDFs extend to the right substantially. Despite the large size of the test set, the highest percentiles of the CDF can yield surprisingly good scores.

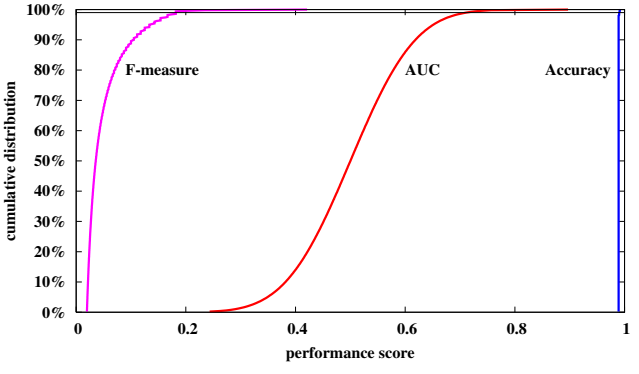


Fig. 1. CDF of scores for AUC, accuracy and F-measure for P=10, N=1000.

Computational Efficiency: All of the computations are simple to implement, yet its cost is in CPU cycles to accurately estimate the tail of the distribution. For C=100, it requires R=9,950,416 repetitions, which takes 5.5 hours for our implementation to compute for P=N=100 on a 1.8GHz HP Evo laptop. For C=1000, nearly 100M repetitions are called for, though fewer may be performed at some loss of precision in the estimation of the critical value. By keeping only the top scores, the memory required is minor; but to record the entire CDF would consume R floating point numbers, e.g. 380MB for C=1000.

Generating the critical value charts in the following section consumed more than a year of CPU time, run on over a hundred CPUs provided by the HP Labs Utility Data Center. Calling the procedure outlined in Table 2 for each performance metric and test condition would have required nearly 12 years of CPU time. To make this reference work feasible, a more efficient computation was performed, which computes the critical value for all performance measures and all values of C during a single run of R repetitions. This procedure is given in Table 3. We use R=10M, being sufficient for the level of accuracy displayable in the charts following. The number of top scores to keep track of is established at initialization by the smallest C value. Because such high scores are rare in the top tail of the distribution, a great deal of priority heap management can be avoided by not inserting values smaller than the smallest value in the heap once it reaches the maximum size required. Otherwise, the value is inserted, and the smallest value is deleted if the heap is full.

Table 3. Procedure for computing critical value charts

```
For P = 2..1000:
  For N = 2..1000:
    Declare an empty scores array for each performance metric
    For R=10,000,000 repetitions:
      Randomly order P positives and N negatives
      Score the ordering by each performance metric
      Keep only the top scores for each in the associated array
    For each C = 10; 100; 1000:
      Output the (R * (1 – 0.99^(1/C) ))^-th best score
```

Analytical Solution for TP10: We derive the solution for TP10 analytically rather than by simulation. The formula is valid for TP<n>, where n is any positive integer.

For a random classifier, the TP10 score represents the number of positives drawn *without replacement* in ten trials from an 'urn' containing P positives and N negatives. Therefore, the TP10 score is represented by the hypergeometric distribution, with parameters P, N+P and ten trials. The CDF of the hyper-geometric distribution for any number of trials n is given by:

$$p = F(x \mid N+P, P, n) = \sum_{i=1}^{x} \frac{\binom{P}{i}\binom{N}{n-i}}{\binom{N+P}{n}}$$

The result, p, is the probability of drawing up to x of the P positives in n drawings without replacement from a group of $N+P$ positives and negatives.

Given the desired α value, we can compute the inverse of the CDF above at the point $(1-\alpha)$ and get the smallest TP10 score (that is, x) for which the CDF is at least $(1-\alpha)$. Using this analytical knowledge, we compute the entire set of significance tables for TP10 shown in the following section, varying N and P as discussed earlier. The computations were performed in MATLAB, and take under a minute for all the points presented in the tables. The results also allowed us to corroborate the correctness of the simulation software.

4 Critical Value Charts

In this section we provide the critical value charts for each of the four performance metrics: AUC, accuracy, F-measure and TP10. From these charts, researchers and practitioners can read an estimate of the critical value for their experiment for up to 1000 positives and 1000 negatives in their test set.

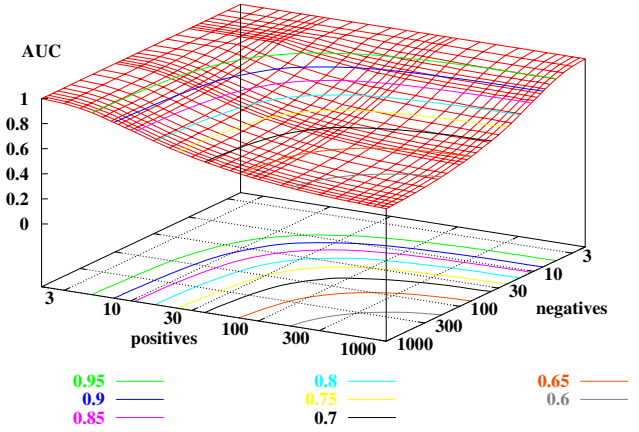


Fig. 2. Critical values for the AUC performance metric, significance level α=1% for C=1000 competing classifiers

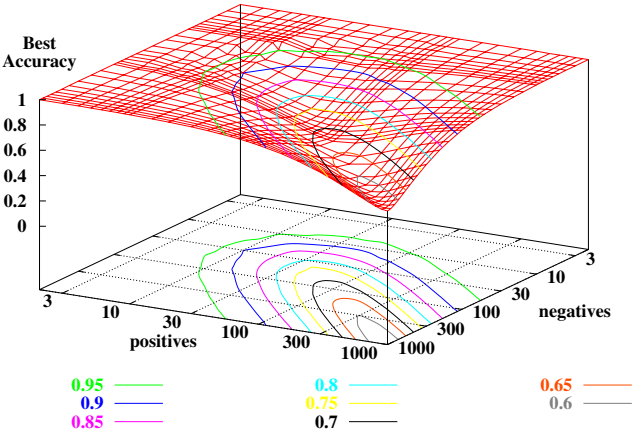


Fig. 3. Critical values for Accuracy, α=1%, C=1000

Figures 2—5 show 3D perspectives of the critical values for C=1000 competing classifiers as we vary N and P along the x-axis and y-axis, respectively. These charts demark the surface of critical values with colored isoclines overlaid and also projected onto the plane below the surface to aid in reading the absolute z-value. For example, given a test set of 100 positives and 300 negatives, the critical value for AUC with 1000 competitors is about 0.65. If the best classifier achieved an AUC greater than this value, the null hypothesis is rejected, deeming the result significant.

The surfaces of the different measures reveal information on what mix of positives and negatives provides low critical values (easier to obtain significance), and what test sets are more demanding for obtaining significant results. Both the accuracy and

AUC surfaces are symmetric with respect to the number of positives and negatives. The critical value for AUC is nearly 1.0 with very few positives *or* very few negatives, but as the test set becomes large, the critical value approaches 0.5—the expected area under the ROC curve of a single random classifier. AUC has a relatively large region with low values, indicating it is relatively insensitive to the mix of positives and negatives, as long as the neither is especially small.

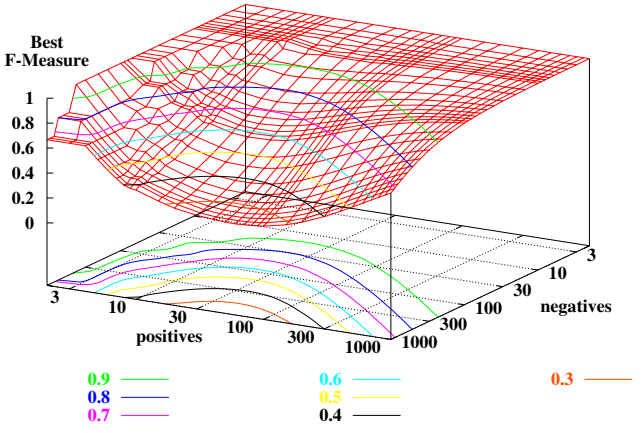


Fig. 4. Critical values for F-measure. α=1%, C=1000

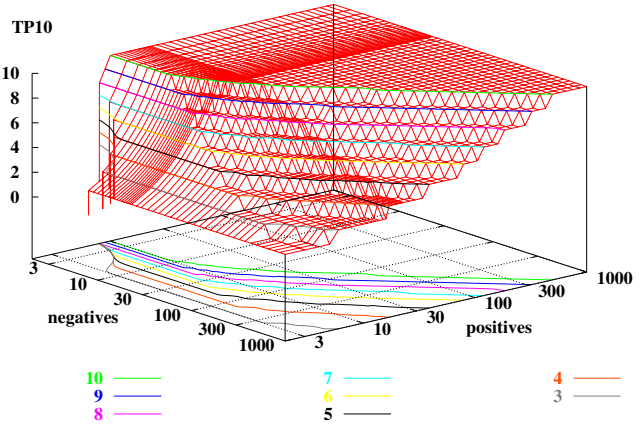
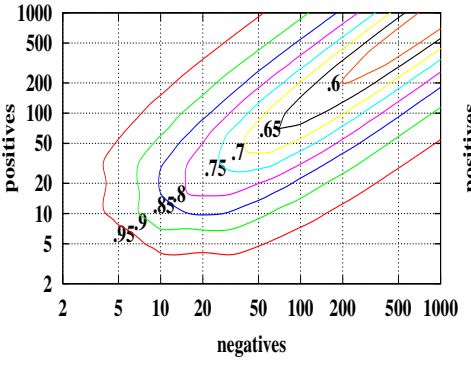
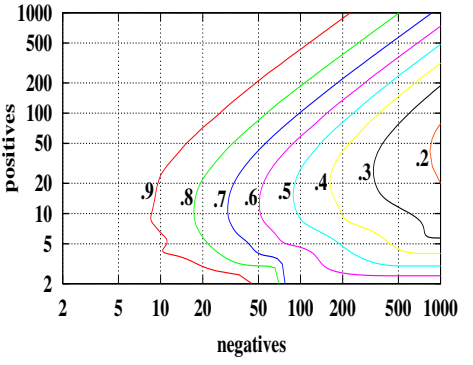


Fig. 5. Critical values for TP10. α=1%, C=1000
Note: The x and y axes are rotated differently than Fig. 4

Accuracy, on the other hand, has a much smaller valley of low critical values, concentrated around the line N=P (having ~50% accuracy). This is due to the fact that the expected value under the null hypothesis depends on the ratio of P and N; with highly skewed class distributions, the critical value can approach 100% accuracy, just

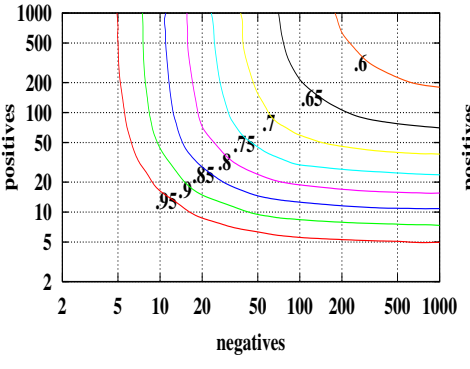


(a) Accuracy

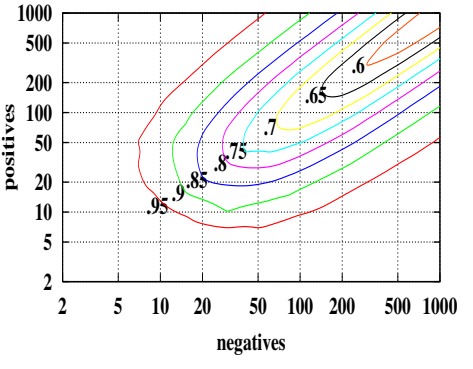


(b) F-Measure

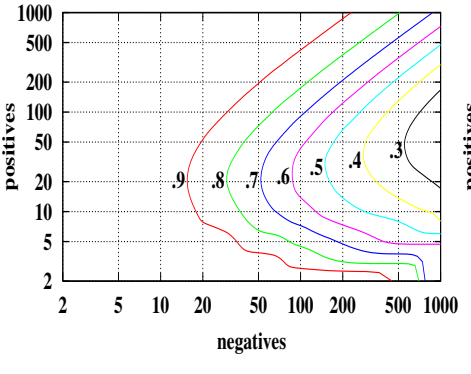
Fig. 6. Contours of critical values for α=1%, C=10 competitors



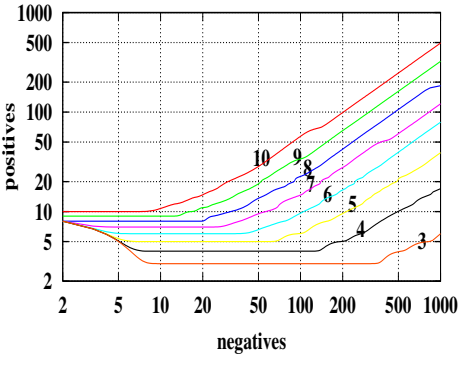
(a) AUC



(b) Accuracy



(c) F-Measure



(d) TP10

Fig. 7. Contours of critical values for α=1%, C=1000 competitors

as with majority voting. Majority voting, however, maintains 50% accuracy for P=N=10, whereas with C=1000 competitors, the critical value for best accuracy climbs to 95%, as witnessed at the back of the valley. (For comparison, at P=N=100, the critical value is 67% accuracy for C=1000 competitors, and 65% for C=100 competitors.)

F-measure and TP10 are not symmetric with respect to N and P, and depend more on varying P. With TP10, a small number of positives and large number of negatives yields lower critical values, and higher values as the number of positives increases. When P=N, TP10 is useless for discriminating good performance. For F-measure, a test set with large number of negatives and smaller number of positives, yields the lowest critical value. Note that F-measure has a larger valley than accuracy, making it preferable over a wider range of P and N values, albeit over a somewhat different region than accuracy.

To make the charts easier to read, we show in Figures 6-7 only the projections of the isoclines onto the plane below the surfaces, for C=10 and 1000 for each performance measure. The isoclines are labeled with their critical value, making it easy to find the critical value for a given P and N in a test set. For visual clarity, the density of the isoclines is low; to obtain precise critical values, refer to the tables in the appendix provided in the online version of this paper [5]. It also contains a complete set of color charts for C=10, C=100 and C=1000 competitors, for each of the four performance metrics.

5 Discussion

Consider the realistic scenario of a data-mining practitioner at a pharmaceutical company who is given a difficult biomedical classification task with a limited dataset that was expensive and slow to obtain. If the best of a dozen learning methods obtains only a mediocre AUC score, the method and critical value charts in this paper provide the practitioner with a simple way to determine whether the finding of the 'best' classifier is statistically significant. If not, the company can make an informed decision either to accept greater risk of failing to reject the null hypothesis, or to collect additional testing data.

As researchers and practitioners, we want to perform effective experiments that are statistically significant. Towards this end, we desire to select our test metrics and/or test set class distributions such that we operate in a region with generally low critical values. When selecting test metrics, accuracy may be appropriate when $P \approx N$, but AUC may be preferred for its larger valley, or F-measure when $P < N$.

In some situations the choice of metric is fixed by the application, e.g. TP10 is most appropriate for many information retrieval applications. Suppose we are given P=N=1000 test examples. For proper testing, it would be most effective to omit a large fraction of the positives, so that the critical value for TP10 is small. A similar scenario can be painted for F-measure when too great a ratio of positives to negatives is available for testing. Of course, if the class distribution of the target population is known, it may be the most appropriate for comparison. Commonly, however, the number of positives and negatives available is due to irrelevant historical reasons.

The independence assumption used in this work is between the competitors, and so it holds equally for cross-validation testing as for held-out validation testing. However, if the many competitors amount to a single learning algorithm with hundreds of different parameterizations, then the independence assumption is in question. One might choose to reduce C below the number of parameterizations attempted, but there is no sanctioned method for doing this.

Finally, we note that by averaging performance scores across many independent test sets, one increases the effective size of the test set, but this also changes the distribution under the null hypothesis. Separate critical value calculations may be required in this case.

6 Related Work

The most common form of significance testing in machine learning papers or anywhere is in determining whether one method is statistically significantly better than another method on average. For this, one computes the mean and standard deviation of the differences over a sample of n test problems, and refers to reference tables of the critical values for the standard Normal(0,1) distribution, or the Student t-distribution if the sample size is small (less than 30). This significance test only compares a single pair of methods, so if there are 100 methods, there are ~100 x 100 comparisons to make, and at α=0.01, there could easily be ~100 cases where we fail to reject the null hypothesis and mistakenly claim significance. This is known as the problem of *multiple comparisons* [6]. The problem also occurs for the common practice of counting wins/ties/losses for each pair of competing classifiers.

The *Bonferroni correction* is a well-known method for adjusting α when there are multiple comparisons. For example, with 100 competitors and performing each of the (100 choose 2) pair-wise comparisons, there would be ~50 'statistically significant differences' found by chance alone if we use the uncorrected α of 0.01. With the Bonferroni correction, one would have to lower α for each test to 0.00000203 to bring the overall α risk back to 0.01. This correction has several problems [8]. It requires evaluating the inverse CDF much further down the tail of the distribution, resulting in 50× as much computation for C=100. Moreover, by being so extremely conservative for the type II error of failing to reject the null hypothesis, it greatly increases the type I risk of failing to accept a significant difference when one is present. The root problem stems from the quadratic number of pair-wise comparisons, which are not actually the desired result for most purposes.

Ultimately, what people want to know is which model is best, with confidence in the significance of the finding—our focus. The *randomization method* [6] addresses this issue by training the chosen best learning model repeatedly on the training set, but randomly overwriting the labels of the training set, to produce a distribution of scores under this null hypothesis. For large training sets and/or computation-intense learning models such as neural networks, this approach can be computationally intractable. Also, this approach is infeasible in some situations, such as in a data mining competition or a proprietary model generated for evaluation by a business. The *randomized distribution analysis method* [4] resolves these issues by generating many trivial classifier models that are quick to train and evaluate, such as Naïve Bayes

based on one or a few randomly chosen features. Note that the training labels are not randomly overwritten, so this null hypothesis is a stronger condition—stating that the result could be achieved by trivial classifiers. Assuming some of the features are predictive individually, this null hypothesis is likely to achieve higher scores, and thus reject the statistical significance of comparisons more often. However, it is a good baseline to use when deciding, for example, whether some complicated, expensive method is worthwhile to deploy over simple methods. This null hypothesis helps determine whether the winning learning model has a competitive advantage over simple methods available to all. One disadvantage of the method as reported is that it is not founded on the principles of statistical significance, but on expected value: If the maximum score achieved by the best classifier is a small amount greater than the expected value, it gives one no guidance on how rare this event is. The method could be recast in terms of statistical significance if sufficiently many features are available to generate enough random samples.

One advantage our method has over both of these randomized methods is that it does not depend on the training data or the features of the dataset whatsoever. In this way, we can pre-compute critical value tables for quick reference by all researchers.

7 Conclusion

This paper applied statistical foundations to develop the critical value charts and procedure for determining when the best classifier performance found from among C independent competitors on a test set containing P positives and N negatives is a statistically significant finding. When not, there is a $\alpha \geq 1\%$ chance that the finding could have been generated by random processes under the null hypothesis. We developed the method for four commonly used performance metrics for binary classification tasks.

The charts presented in this paper, and in the online appendix [5], serve as a quick reference guide for practitioners seeking to reject the null hypothesis. The charts can easily be extended to cover other situations, using the procedures described.

To conclude, in addition to providing the critical values for significance testing of binary classifiers, this paper tries to emphasize the importance of the null hypothesis test in machine learning and data mining research and to remind ourselves to beware of the null hypothesis, so we know that our results are really significant. Nonetheless, passing these statistical tests cannot guarantee that a given classifier is genuinely useful, as always.

Acknowledgments

We wish to thank the volunteers who arrange interesting challenges for each year's KDD Cup—valuable lessons are regularly brought to light through them. We are grateful to Hsiu-Khuern Tang for his valuable statistics consulting, and the Hewlett-Packard Utility Data Center for computing cycles.

References

1. Abroise, C. & McLachlan, G.: Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl. Acad. Sci. USA,* 99, 10 (2002), 6562-6566.
2. Blake, C.L. & Merz, C.J.: UCI Repository of machine learning databases. University of California, Dept. of Information and Computer Science, Irvine, CA. (1998).
3. Fawcett, T.: *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*. Tech. Report HPL-2003-4, Hewlett-Packard, (2003).
4. Forman, G.: *A Method for Discovering the Insignificance of One's Best Classifier and the Unlearnability of a Classification Task.* Data Mining Lessons Learned Workshop, 19th International Conference on Machine Learning (ICML), Sydney, Australia, (2002).
5. Forman, G. & Cohen, I.: Beware the Null Hypothesis: Critical Value Tables for Evaluating Classifiers. Tech. Report HPL-2005-70, Hewlett-Packard, (2005). http://www.hpl.hp.com/techreports/2005/HPL-2005-70.html
6. Jensen, D. & Cohen, P.: Multiple Comparisons in Induction Algorithms. *Machine Learning*, *38*, 3 (2000), 309-338.
7. Mangasarian, O. L. & Wolberg, W. H.: Cancer diagnosis via linear programming. *SIAM News*, *23,* 5 (Sept. 1990), 1-18.
8. Perneger, T. V.: What is wrong with Bonferroni adjustments. *British Medical Journal, 136* (1998), 1236-1238.