Permutation Tests for Studying Classifier Performance

Markus Ojala Gemma C. Garriga
Helsinki Institute for Information Technology HIIT
Department of Information and Computer Science
Helsinki University of Technology, Finland
Email: firstname.lastname@tkk.fi

***Abstract*—We explore the framework of permutation-based p-values for assessing the behavior of the classification error. In this paper we study two simple permutation tests. The first test estimates the null distribution by permuting the labels in the data; this has been used extensively in classification problems in computational biology. The second test produces permutations of the features within classes, inspired by restricted randomization techniques traditionally used in statistics. We study the properties of these tests and present an extensive empirical evaluation on real and synthetic data. Our analysis shows that studying the classification error via permutation tests is effective; in particular, the restricted permutation test clearly reveals whether the classifier exploits the interdependency between the features in the data.**

***Keywords*-classification, labeled data, permutation tests, restricted randomization, significance testing**

I. Introduction

Building effective classification systems is a central task for data mining and machine learning. Usually, a classification algorithm builds a model from a given set of data records in which the labels are known and later, the learned model is used to assign labels to new data points. Applications of such classification setting abound in many fields, for instance, in text categorization, fraud detection, optical character recognition, or medical diagnosis to cite some.

For all these applications, a desired property of a good classifier is the power of generalization to new unknown examples. The detection and characterization of significant predictive patterns is crucial for obtaining a good classification accuracy that generalizes beyond the training data. Unfortunately, it is very often the case that the number of available data points with labels is not sufficient. Data from medical or biological applications, for example, is characterized by high dimensionality (thousands of features) and small number of data points (tens of rows). A crucial question is whether we should believe in the classification accuracy returned by such classifiers.

The most traditional approach to this problem is to estimate the error of the classifier by means of cross-validation or leave-one-out cross-validation, among others. This estimate, together with a variance-based bound, would provide an interval for the expected error of the classifier. However, it has been argued that evaluating the classifier with an error measurement is ineffective for small data samples [1]–[4]. Also classical generalization bounds are not appropriate when the dimensionality of the data is too high. Indeed, for many other general cases, it is useful to have other statistics associated to the error in order to understand better the behavior of the classifier. For example, even if a classification algorithm produces a classifier with low error, the data itself may have no structure. Thus the question is, how can we trust that the classifier has learned a significant predictive pattern in the data and that the chosen classifier is appropriate for the specific classification task?

Dataset D_1		Dataset D_2	
o x x x x x x x	+	x x x o x x x x	+
x x o x x x x o	+	x x x x o x x x	+
x x x x o o x x	+	x x x x x x x x	+
x x x x x x x o	+	x o x x x x x x	+
x x o x o o o x	+	o o o o o o o x	+
x x x x x x x o	+	x o o o o o o o	+
x o o x o x x x	+	o o o o o x o o	+
x x x x o x x o	+	o o o o o o o o	+
o o o x x o o o	−	x x x x o o o x	−
o o o o o o o o	−	x x x x x o o o	−
x o x o o o o o	−	x x o x o o o o	−
x o x o o x o o	−	x x x x o o o o	−
o o x o o o o o	−	o o o o x x x x	−
o o o o o o x o	−	o o o o x x x x	−
x o o o o o o o	−	o x o o x x x o	−
o o o x o o o o	−	o o o x x x x x	−

Figure 1. Examples of two 16×8 nominal datasets D_1 and D_2 each having two classes. The last column in both datasets denotes the class labels (+, −) of the samples in the rows.

Consider the small toy example from Figure 1. There we have two nominal data matrices D_1 and D_2 of sizes 16×8; each row (data point) has two different values present, x and o. Both datasets have a clear separation into the two given classes, + and −. However, it seems at first sight that the structure within the classes for dataset D_1 is much simpler than for dataset D_2. If we train a 1-Nearest Neighbor classifier on the datasets of Figure 1, we have that the classification error (leave-one-out cross-validation) is 0.00 on both D_1 and D_2. However, is it true that the classifier is using a real dependency in the data? Or are the dependencies in D_1

or D_2 just a random artifact of some simple structure? This example will be analyzed with detail later on in the paper.

In the recent years, a number of papers have suggested to use permutation-based p-values for assessing the competence of a classifier [2], [3], [5], [6]. Essentially the permutation test procedure measures how likely the observed accuracy would be obtained by chance. A p-value represents the fraction of random datasets under a certain null hypothesis where the classifier behaved better than in the original data.

Traditional permutation tests study the null hypothesis that the features and the labels are independent, that is, that there is no difference between the classes. The null distribution under this null hypothesis is estimated by permuting the labels of the dataset. This corresponds also to the most traditional statistic methods [7], where the results on a control group are compared against the result on a treatment group. This simple test has been proven effective already for selecting relevant genes in small data samples [8] or for attribute selection in decision trees [9]. However, the related literature has not performed extensive experimental studies for this traditional test in more general cases. Sub-sampling methods such as bootstrapping [10] use randomizations to study the properties of the underlying distribution instead of testing the data against some null model.

The goal of this paper is to study permutation tests for assessing the behavior of the error in classifiers. We first study the permutation test that simply permutes the labels; our experimental studies suggest that this traditional null hypothesis leads to very low p-values, thus rendering the classifier significant most of the time. We therefore propose a new test and study its relation to the traditional methods. Our test is inspired by restricted randomization techniques [7], traditionally used in statistics. The new null distribution tests the dependency between the features.

The idea is to provide users with practical p-values for the analysis of the classifier. The permutation tests give us useful statistics about the underlying reasons for the obtained classification result. No test is better than the other, but all provide us with information about the classifier performance. Each p-value depends on the original data (whether it contains some real structure or not) and the classifier (whether it is able to use certain structure in the data or not).

II. Background

Let X be an $n \times m$ data matrix. For example, in gene expression analysis the values of the matrix X are numerical expression measurements, each row is a tissue sample and each column represents a gene. We denote the i-th row vector of X by X_i and the j-th column vector of X by X^j. Rows are also called observations or data points, while columns are also called attributes or features. Observe that we do not restrict the data domain of X and therefore the scale of its attributes can be categorical or numerical.

Associated to the data points X_i we have a class label y_i. We assume a finite set of known class labels $\mathcal{Y}$, so $y_i \in \mathcal{Y}$. Let D be the set of labeled data $D = \{(X_i, y_i)\}_{i=1}^n$. For the gene expression example above, the class labels associated to each tissue sample could be, e.g., "sick" or "healthy".

In a traditional classification task the aim is to predict the label of new data points by training a classifier from D. The function learned by the classification algorithm is denoted by $f : \mathcal{X} \to \mathcal{Y}$. A test statistic is typically computed to evaluate the classifier performance: this can be either the training error, cross-validation error or jackknife estimate, among others. Here we give as an example the leave-one-out cross-validation error,

$$e(f, D) = \frac{1}{n} \sum_{i=1}^{n} \mathrm{I}(f_{D \setminus D_i}(X_i) \neq y_i) \tag{1}$$

where $f_{D \setminus D_i}$ is the function learned by the classification algorithm by removing the i-th observation from the data and $\mathrm{I}(\cdot)$ is the indicator function.

Recently, a number of papers use permutation-based p-values for assessing the competence of a classifier.

Definition 1 (Permutation-based p-value)**.** *Let $\widehat{D} = \{D'_1, \ldots, D'_k\}$ be the set of k randomized versions of the original data D sampled from a given null distribution. The empirical p-value for the classifier f is calculated as follows [7],*

$$p = \frac{|\{D' \in \widehat{D} : e(f, D') \leq e(f, D)\}| + 1}{k + 1}.$$

Intuitively, the p-value of Definition 1 measures how likely the observed accuracy would be obtained by chance, only because the classifier identified in the training phase a pattern that happened to be random. It represents the fraction of randomized samples where the classifier behaved better in the random data than in the original data. Therefore, if the p-value is small enough—usually under a certain threshold, e.g., $\alpha = 0.05$—we can say that the value of the error in the original data is indeed significantly small and in consequence, that the classifier is significant under the given null hypothesis, i.e., the null hypothesis is rejected.

III. Permutation Tests for Labeled Data

In this section we describe in detail two very simple permutation methods to estimate the null distribution of the error under two different null hypotheses. Let π be a permutation of natural numbers $\{1, \ldots, n\}$: we denote with $\pi(y)_i$ the i-th value of the vector label y induced by the permutation π; for the general case of a column vector X^j, we use $\pi(X^j)$ to represent the permutation of the vector X^j induced by π. Finally, we denote the concatenation of column vectors into a matrix by $X = [X^1, X^2, \ldots, X^m]$.

The first permutation method is the standard permutation test used in statistics [7]. The null hypothesis assumes

that the data X and the labels y are independent, that is, $p(X, y) = p(X)p(y)$. The distribution under this null hypothesis is estimated by permuting the labels in D.

Test 1 (Permute labels)**.** *Let $D = \{(X_i, y_i)\}_{i=1}^n$ be the original dataset and let π be a permutation of natural numbers $\{1, \ldots, n\}$. One randomized version D' of D is obtained by applying the permutation π on the labels, $D' = \{(X_i, \pi(y)_i)\}_{i=1}^n$. Compute the p-value as in Definition 1.*

A significant classifier for Test 1 rejects the null hypothesis that the features and the labels are independent, i.e., that there is no difference between the classes. If the original data contains dependency between data points and labels, then: (1) a significant classifier f will use such information to achieve a good classification accuracy, resulting into a small p-value; (2) if the classifier f is not significant with Test 1, f was not able to use the existing dependency between data and labels in the original data. Finally, if the original data did not contain any real dependency between data points and labels, then all classifiers would have a high p-value and the null hypothesis would never be rejected.

Applying randomizations on the original data is therefore a powerful way to understand how the different classifiers use the structure implicit in the data, if such structure exists. However, notice that a classifier might be using some additional dependency structure in the data, for example the dependency between features, which is not checked by Test 1. Indeed, it is very often the case that the p-values obtained from Test 1 are very small, making the test practically worthless for real data. Therefore, we will study also the dependency between the features within the same class. The second null hypothesis assumes that the columns in X are mutually independent inside the same class, thus $p(X(c)) = p(X(c)^1) \cdots p(X(c)^m)$, where $X(c)$ represents the submatrix of X in class label $c \in \mathcal{Y}$. Test 2 is inspired by the restricted randomizations from statistics (see e.g. [7]).

Test 2 (Permute data columns per class)**.** *Let $D = \{(X_i, y_i)\}_{i=1}^n$ be the data. A randomized version D' of D is obtained by applying independent permutations to the columns of X within each class. That is:*

For each class label $c \in \mathcal{Y}$ do,

- *Let $X(c)$ be the submatrix from X in class label c, that is: $X(c) = \{X_i | y_i = c\}$ of size $l_c \times m$.*
- *Let $\pi_1, \ldots, \pi_m$ be m independent permutations from numbers $\{1, \ldots, l_c\}$.*
- *Let $X(c)'$ be a randomized version of $X(c)$ where each π_j is applied independently to the column $X(c)^j$. That is $X(c)' = [\pi_1(X(c)^1), \ldots, \pi_m(X(c)^m)]$.*

Finally, let $X' = \{X(c)' | c \in Y\}$ and obtain one randomized version $D' = \{(X'_i, y_i)\}_{i=1}^n$. Next, compute the p-value as in Definition 1.

Thus, a classification result can be regarded as nonsignificant with Test 2, if either the features are independent of each other inside the classes or if the classifier does not exploit the interdependency between the features. If the dependency is not used, simpler methods could suffice.

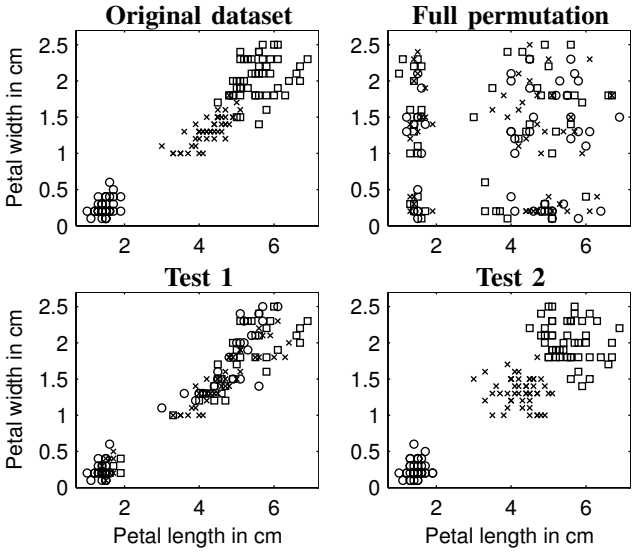

Figure 2. Scatter plots of original Iris dataset and randomized versions for full permutation of the data and for Tests 1 and 2. The data points belong to three different classes denoted by different markers.

Table I
AVERAGE ERROR AND p-VALUE FOR TEST 1 AND TEST 2 WHEN USING THE 1-NEAREST NEIGHBOR CLASSIFIER TO DATASETS OF FIGURE 1.

1-Nearest Neighbor					
	Orig.	Test 1		Test 2	
Dataset	Err.	Err. (Std)	p-val.	Err. (Std)	p-val.
D_1	0.00	0.52 (0.14)	0.001	0.06 (0.06)	**0.358**
D_2	0.00	0.53 (0.14)	0.001	0.62 (0.14)	0.001

In Figure 2, we give as example one randomization for each test on the well-known Iris dataset. For comparison, we include a test corresponding to full permutation of the data. Note how well Test 2 has preserved the class structure compared to other tests. We will discuss the Iris data more in the experiments.

A. Example

We illustrate the concept of the tests by studying the small artificial example presented in the introduction in Figure 1. Both datasets D_1 and D_2 have a clear separation into the two given classes, + and −. However, the structure inside the dataset D_1 is much simpler than in the dataset D_2. We analyze this with the 1-Nearest Neighbor classifier using the leave-one-out cross-validation given in Equation (1). The classification error obtained in the original data is 0.00 for both D_1 and D_2, which is expected since the datasets were generated to contain clear class structure.

We apply Test 1 and Test 2 to study the performance of 1-Nearest Neighbor classifier on the datasets D_1 and D_2. We

produce 1000 random samples for both the datasets with Tests 1 and 2, and perform the same leave-one-out cross-validation procedure to obtain a classification error for each randomized dataset. The results are summarized in Table I.

We can say that the classifiers are significant under the null hypothesis that data and labels are independent (Test 1); However, it is easy to argue that the results of Test 1 do not provide much information about the classifier performance. Actually the main problem of Test 1 is that p-values tend to be always very low as the null hypothesis is typically easy to reject. On the other hand, for Test 2, the 1-Nearest Neighbor classifier is significant for dataset D_2 but not for dataset D_1. Indeed, the dataset D_1 was generated so that the features are independent inside the classes, and hence, the good classification accuracy of the algorithm on D_1 is simply due to different value distributions across the classes. For dataset D_2 we have that the dependence between the columns inside the classes is essential for the good classification result, and the classifier has been able to exploit that information.

IV. Behavior of the Tests

To understand better the behavior of the tests, consider next the following simulated data, inspired by the data used by Golland *et al.* in [2]: 100 data points are generated from two-dimensional normal distribution with mean vector (1,0), unit variance and covariance $\rho \in [-1, 1]$. Another 100 data points are generated from similar normal distribution with mean $(-1, 0)$, unit variance and same covariance ρ. The first 100 samples are assigned with class label $y = +1$ and the other 100 samples with class label $y = -1$. Note that the correlation between the features improves the class separation: if the correlation $\rho = 1$, we have that the class $y = x_1 - x_2$ where x_1, x_2 are the values of the first and second features, respectively.

For these datasets (with varying correlation) we use the stratified 10-fold cross-validation error. We study the behavior of four classifiers: 1-Nearest Neighbor, Decision Tree, Naive Bayes and Support Vector Machine. We use Weka 3.6 data mining software [11] with the default parameters of those classification algorithms. The Decision Tree classifier is similar to C4.5 algorithm, and the default kernel used with Support Vector Machine is linear.

Figure 3 shows the behavior of the classifiers on datasets with the correlation ρ between features inside classes varying from -1 to 1. The Decision Tree, 1-Nearest Neighbor and Support Vector Machine classifiers have been able to exploit the dependency between the features, i.e., the classification error goes to zero when there is either a high positive or negative correlation between the features. However, with Naive Bayes classifier the classification error seems to be independent of the correlation between the features.

For all classifiers we observe that the null hypothesis associated to Test 1 (i.e. labels and data are independent) is always rejected. Thus the data contains a clear class structure

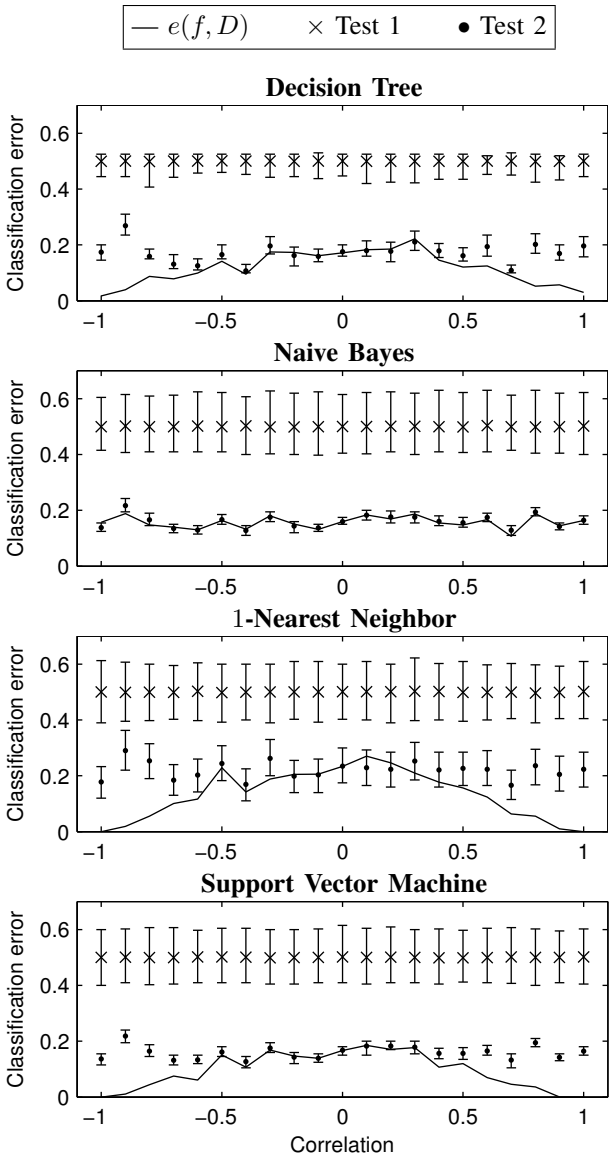

Figure 3. Average values of stratified 10-fold cross-validation error (y-axis) for varying values of correlation between the features per class (x-axis). The continuous line shows the error on the original data, and symbols × and • represent the average of the error on 1000 randomized samples obtained from Test 1 and from Test 2, respectively. Each average of the error on the randomized samples × and • is depicted together with the $[1\%, 99\%]$-deviation bar inside which the associated null hypothesis cannot be rejected with significance level $\alpha = 0.01$. That is, if the continuous line falls outside the bars the null hypothesis associated to the test is rejected; if the continuous line crosses inside the bars the null hypothesis cannot be rejected.

as expected since there exists no class noise in the data. All classifiers are therefore significant under Test 1.

Another expected observation is that the null hypothesis for Test 2 (i.e. features are independent within class) tends to be rejected as the correlation between features increases. That is, the correlation is useful in classifying the data. When the magnitude of the correlation is larger than approximately 0.4, the Decision Tree, Nearest Neighbor and Support Vector

Table II
CLASSIFICATION ERRORS AND EMPIRICAL p-VALUES OBTAINED WITH DECISION TREE FOR TEST 1 AND TEST 2. THE EMPIRICAL p-VALUES ARE CALCULATED OVER 1000 RANDOMIZED SAMPLES. BOLD p-VALUES CORRESPOND TO NONSIGNIFICANT RESULTS.

Decision Tree

	Orig.	Test 1		Test 2	
Dataset	Err.	Err. (Std)	p-val.	Err. (Std)	p-val.
Anneal	0.07	0.24 (0.00)	0.001	0.13 (0.01)	0.001
Audiology	0.22	0.82 (0.03)	0.001	0.23 (0.02)	**0.482**
Autos	0.19	0.76 (0.04)	0.001	0.38 (0.04)	0.001
Balance	0.22	0.55 (0.02)	0.001	0.29 (0.02)	0.001
Breast	0.26	0.30 (0.00)	0.001	0.29 (0.02)	**0.116**
German	0.29	0.31 (0.01)	0.005	0.28 (0.01)	**0.666**
Glass	0.33	0.72 (0.03)	0.001	0.34 (0.03)	**0.457**
Hepatitis	0.22	0.23 (0.02)	**0.319**	0.15 (0.03)	**0.955**
Ionosphere	0.10	0.38 (0.02)	0.001	0.07 (0.01)	**0.964**
Iris	0.05	0.67 (0.03)	0.001	0.05 (0.01)	**0.765**
Lymph	0.22	0.51 (0.05)	0.001	0.23 (0.04)	**0.437**
Mushroom	0.00	0.50 (0.01)	0.001	0.01 (0.00)	0.001
Pima	0.25	0.35 (0.01)	0.001	0.24 (0.02)	**0.642**
Promoters	0.21	0.50 (0.06)	0.002	0.22 (0.05)	**0.377**
Segment	0.13	0.86 (0.03)	0.001	0.17 (0.02)	**0.132**
Sonar	0.27	0.49 (0.03)	0.001	0.27 (0.03)	**0.507**
Spect	0.19	0.22 (0.01)	0.004	0.15 (0.02)	**0.966**
Splice	0.06	0.60 (0.01)	0.001	0.07 (0.01)	0.002
Tic-tac-toe	0.15	0.35 (0.01)	0.001	0.30 (0.01)	0.001
Tumor	0.58	0.82 (0.02)	0.001	0.60 (0.02)	**0.138**
Votes	0.03	0.42 (0.02)	0.001	0.03 (0.01)	**0.791**
Zoo	0.07	0.64 (0.03)	0.001	0.07 (0.01)	**0.593**

Table III
CLASSIFICATION ERRORS AND EMPIRICAL p-VALUES FOR 1-NEAREST NEIGHBOR UNDER TEST 1 AND TEST 2. THE EMPIRICAL p-VALUES ARE CALCULATED OVER 1000 RANDOMIZED SAMPLES. BOLD p-VALUES CORRESPOND TO NONSIGNIFICANT RESULTS.

1-Nearest Neighbor

	Orig.	Test 1		Test 2	
Dataset	Err.	Err. (Std)	p-val.	Err. (Std)	p-val.
Anneal	0.05	0.40 (0.02)	0.001	0.08 (0.01)	0.001
Audiology	0.26	0.86 (0.03)	0.001	0.32 (0.03)	**0.030**
Autos	0.26	0.77 (0.03)	0.001	0.45 (0.03)	0.001
Balance	0.20	0.56 (0.02)	0.001	0.35 (0.02)	0.001
Breast	0.31	0.41 (0.03)	0.007	0.32 (0.03)	**0.324**
German	0.28	0.42 (0.02)	0.001	0.33 (0.02)	0.002
Glass	0.30	0.74 (0.04)	0.001	0.42 (0.03)	0.001
Hepatitis	0.19	0.33 (0.04)	0.002	0.14 (0.03)	**0.970**
Ionosphere	0.13	0.46 (0.03)	0.001	0.26 (0.01)	0.001
Iris	0.05	0.66 (0.05)	0.001	0.02 (0.01)	**0.962**
Lymph	0.18	0.53 (0.04)	0.001	0.20 (0.03)	**0.307**
Mushroom	0.00	0.50 (0.01)	0.001	0.01 (0.00)	0.001
Pima	0.29	0.46 (0.02)	0.001	0.27 (0.02)	**0.866**
Promoters	0.19	0.50 (0.06)	0.001	0.26 (0.04)	**0.083**
Segment	0.14	0.86 (0.03)	0.001	0.15 (0.02)	**0.266**
Sonar	0.13	0.50 (0.04)	0.001	0.27 (0.03)	0.001
Spect	0.24	0.32 (0.04)	0.011	0.18 (0.02)	**0.970**
Splice	0.24	0.61 (0.01)	0.001	0.30 (0.01)	0.001
Tic-tac-toe	0.21	0.44 (0.07)	0.001	0.38 (0.02)	0.001
Tumor	0.66	0.88 (0.02)	0.001	0.62 (0.02)	**0.860**
Votes	0.08	0.47 (0.03)	0.001	0.01 (0.00)	**1.000**
Zoo	0.03	0.75 (0.05)	0.001	0.04 (0.02)	**0.333**

Machine classifiers reject the null hypothesis. Thus these classifiers produce significant results under Test 2 when the features are highly correlated.

Finally, observe the behavior of Naive Bayes classifier for Test 2: the null hypothesis can never be rejected. This is because Naive Bayes classifier explicitly assumes by default that the features are independent, thus it always performs similarly on the original data and the randomized datasets, which results into a very high p-value.

V. EMPIRICAL RESULTS

In this section we give empirical results on 22 various real datasets from UCI machine learning repository [12]. The datasets contain nominal or/and numeric features as well as missing values. In most datasets the features are measured in different scales, thus it is only reasonable to consider column-wise permutations, leaving out of consideration some recent data mining randomization methods [13], [14]. We use stratified 10-fold cross-validation error as the statistic. In all cases, we calculate the empirical p-values over 1000 randomized samples and use the threshold of $\alpha = 0.01$ to determine the significance of the classification result. Since the original classification error is not a stable result due to the randomness in forming the folds and training the classifier, we perform the cross-validation ten times for the original datasets and calculate an empirical p-value for each of the ten results. In all the tables, we give the average value of these empirical p-values as well as the average value of the original classification error.

The significance testing results for the Decision Tree classifier are given in Table II and for 1-Nearest Neighbor classifier in Table III. The original cross-validation error is given as well as the mean and standard deviation of the errors on the 1000 randomized samples with Test 1 and Test 2. With Naive Bayes classifier the classification results were regarded on all datasets as significant with Test 1 and as nonsignificant with Test 2, as expected by the analysis in Section IV. The results with Support Vector Machine were similar to results with Decision Tree classifier.

The results for Test 1 show that the classification errors with most datasets are regarded as significant with threshold $\alpha = 0.01$. These results show that the datasets contain clear class structure; however, they do not give any additional insight for understanding the class structure in the datasets.

Next, we consider the results for permuting the features inside each class (Test 2). There are actually now more nonsignificant results than significant ones. Thus, in most datasets the original structure inside the classes is pretty simple, or it is not used by the classification algorithm. That is, the classes differ from each other mainly due to their

different value distributions and not due to some dependence between the features. Thus, in most of the datasets the class structure is explained by considering features independently of each other. The 1-Nearest Neighbor classifier has been able to use the dependency of features the most, i.e., containing the most of small, significant p-values with Test 2.

Let us now study the results with Test 2 in more detail. Consider the well-known Iris dataset that contains measurements of three different species of iris flowers from four features: the length and the width of sepal and petal. It turns out that the classes are almost linearly separable given the length of petal or given the width of petal. Although there is a high positive linear correlation between the length and width of petal, it is not important for the classification result as both features can explain the classes by themselves.

Actually, observe that for the Iris dataset with Test 2, the classification error on the randomized samples is even smaller than in the original dataset. This phenomenon is explained by the positive linear correlation between the length and the width of petal, which disappears after the randomizations, as seen in Figure 2 in Section III. Randomizations eliminate most of the rows containing extreme values for both of the features inside the classes. Thus the classifiers do not use the dependency between these two features, as their correlation does not help in classifying the Iris data. When this positive correlation is eliminated per classes, the separation between the classes increases, and therefore, the classification accuracy is improved.

Finally, we study the dataset Balance, which is considered significant under the null hypothesis of Test 2. This data contains four features of a balance scale: left-weight, left-distance, right-weight and right-distance. The scale is in balance if left-weight times left-distance equals right-weight times right-distance. There are three classes: the scale tips to the left, to the right, or is balanced. It is clear that the dependence of the features is important to the classification result.

Understanding the structure inside the datasets where the classification result is regarded as significant under Test 2 requires more study, i.e., we just know that the features do not explain the class structure independently. Analyzing the dependence structure of the features is then a further task. But as seen, the null hypothesis of Test 2 explains most of the good classification results in the 22 datasets.

VI. Conclusions

We have considered the problem of assessing the classifier performance with permutation tests. We have described two different null hypotheses and shown how samples can be produced from the corresponding null models by simple permutation methods. Each test provides an empirical p-value for the classifier performance; each p-value depends on the original data (whether it contains the type of structure tested) and the classifier (whether it is able to use the structure). The null hypotheses can be summarized as follows: (1) the data and the class labels are independent; and (2) the features are mutually independent inside a class.

Experiments showed that the traditional permutation test produces a small p-value even if there is only a weak class structure present. Compared to this, the new test proposed was able to evaluate the underlying reasons for the classifier performance on the real datasets. Surprisingly, however, in most of the studied real datasets the class structure looks fairly simple; the dependency between the features is not used in classifying the data with the four tested classifiers. In such cases, there might be no reason to use complicated classifiers, as simpler methods would suffice.

Future work should explore the use of our tests for selecting the best discriminant features for classifiers, as it has been used for decision trees and other biological applications [8], [9]. Also, it would be useful to extend the setting to unsupervised learning, such as clustering.

References

[1] U. Braga-Neto and E. Dougherty, "Is cross-validation valid for small-sample microarray classification?" *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.

[2] P. Golland, F. Liang, S. Mukherjee, and D. Panchenko, "Permutation tests for classification," in *COLT*, 2005, pp. 501–515.

[3] T. Hsing, S. Attoor, and E. Dougherty, "Relation between permutation-test p values and classifier error estimates," *Mach. Learn.*, vol. 52, no. 1-2, pp. 11–30, 2003.

[4] A. Isaksson, M. Wallman, H. Göransson, and M. Gustafsson, "Cross-validation and bootstrapping are unreliable in small sample classification," *Pattern Recogn. Lett.*, vol. 29, no. 14, pp. 1960–1965, 2008.

[5] D. Jensen, "Induction with randomization testing: decision-oriented analysis of large data sets," Ph.D. dissertation, Washington University, St. Louis, Missouri, USA, 1992.

[6] A. Molinaro, R. Simon, and R. Pfeiffer, "Prediction error estimation: a comparison of resampling methods," *Bioinformatics*, vol. 21, no. 15, pp. 3301–3307, 2005.

[7] P. Good, *Permutation tests: a practical guide to resampling methods for testing hypotheses; Springer series in statistics.* Springer, 2000, vol. 2nd.

[8] R. Maglietta, A. D'Addabbo, A. Piepoli, F. Perri, S. Liuni, G. Pesole, and N. Ancona, "Selection of relevant genes in cancer diagnosis based on their prediction accuracy," *Artif. Intell. Med.*, vol. 40, no. 1, pp. 29–44, 2007.

[9] E. Frank and I. Witten, "Using a permutation test for attribute selection in decision trees," in *ICML*, 1998, pp. 152–160.

[10] B. Efron, "Bootstrap methods: Another look at the jackknife," *The Annals of Statistics*, vol. 7, no. 1, pp. 1–26, 1979.

[11] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. Morgan Kaufmann, San Francisco, 2005.

[12] A. Asuncion and D. Newman, "UCI machine learning repository," http://www.ics.uci.edu/~mlearn/MLRepository.html.

[13] A. Gionis, H. Mannila, T. Mielikäinen, and P. Tsaparas, "Assessing data mining results via swap randomization," *ACM TKDD*, vol. 1, no. 3, 2007.

[14] M. Ojala, N. Vuokko, A. Kallio, N. Haiminen, and H. Mannila, "Randomization of real-valued matrices for assessing the significance of data mining results." in *SDM'08*, 2008, pp. 494–505.