Journal of Machine Learning Research 11 (2010) 1833-1863 Submitted 10/09; Revised 5/10; Published 6/10

Permutation Tests for Studying Classifier Performance

Markus Ojala MARKUS.OJALA@TKK.FI
Helsinki Institute for Information Technology
Department of Information and Computer Science
Aalto University School of Science and Technology
P.O. Box 15400, FI-00076 Aalto, Finland

Gemma C. Garriga GEMMA.GARRIGA@LIP6.FR
Université Pierre et Marie Curie
Laboratoire d'Informatique de Paris 6
4 place Jussieu, 75005 Paris, France

Editor: Xiaotong Shen

Abstract

We explore the framework of permutation-based *p*-values for assessing the performance of classifiers. In this paper we study two simple permutation tests. The first test assess whether the classifier has found a real class structure in the data; the corresponding null distribution is estimated by permuting the labels in the data. This test has been used extensively in classification problems in computational biology. The second test studies whether the classifier is exploiting the dependency between the features in classification; the corresponding null distribution is estimated by permuting the features within classes, inspired by restricted randomization techniques traditionally used in statistics. This new test can serve to identify descriptive features which can be valuable information in improving the classifier performance. We study the properties of these tests and present an extensive empirical evaluation on real and synthetic data. Our analysis shows that studying the classifier performance 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

1. Introduction

Building effective classification systems is a central task in 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 instances. The detection and characterization of statistically 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, are characterized by high

o	x	x	x	x	x	x	x	+
x	x	o	x	x	x	x	o	+
x	x	x	x	o	o	x	x	+
x	x	x	x	x	x	x	o	+
x	x	o	x	o	o	o	x	+
x	x	x	x	x	x	x	o	+
x	o	o	x	o	x	x	x	+
x	x	x	x	o	x	x	o	+
o	o	o	x	x	o	o	o	−
o	o	o	o	o	o	o	o	−
x	o	x	o	o	o	o	o	−
x	o	x	o	o	x	o	o	−
o	o	x	o	o	o	o	o	−
o	o	o	o	o	o	x	o	−
x	o	o	o	o	o	o	o	−
o	o	o	x	o	o	o	o	−

Data Set D_1

x	x	x	o	x	x	x	x	+
x	x	x	x	o	x	x	x	+
x	x	x	x	x	x	x	x	+
x	o	x	x	x	x	x	x	+
o	o	o	o	o	o	o	x	+
x	o	o	o	o	o	o	o	+
o	o	o	o	o	x	o	o	+
o	o	o	o	o	o	o	o	+
x	x	x	x	o	o	o	x	−
x	x	x	x	x	o	o	o	−
x	x	o	x	o	o	o	o	−
x	x	x	x	o	o	o	o	−
o	o	o	o	x	x	x	x	−
o	o	o	o	x	x	x	x	−
o	x	o	o	x	x	x	o	−
o	o	o	x	x	x	x	x	−

Data Set D_2

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

dimensionality (thousands of features) and small number of data points (tens of rows). An important question is whether we should believe in the classification accuracy obtained 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, provides an interval for the expected error of the classifier. The error estimate itself is the best statistics when different classifiers are compared against each other (Hsing et al., 2003). However, it has been argued that evaluating a single classifier with an error measurement is ineffective for small amount of data samples (Braga-Neto and Dougherty, 2004; Golland et al., 2005; Isaksson et al., 2008). Also classical generalization bounds are not directly appropriate when the dimensionality of the data is too high; for these reasons, some recent approaches using filtering and regularization alleviate this problem (Rossi and Villa, 2006; Berlinet et al., 2008). 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?

For instance, consider the small toy example in Figure 1. There are 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 data sets have a clear separation into the two given classes, + and –. However, it seems at first sight that the structure within the classes for data set D_1 is much simpler than for data set D_2. If we train a 1-Nearest Neighbor classifier on the data sets 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? It turns out that the good classification result in D_1 is explained purely by the different value distributions inside the classes whereas in D_2 the interdependency between the features is important in classification. This example will be analyzed in detail later on in Section 3.3.

In recent years, a number of papers have suggested to use permutation-based p-values for assessing the competence of a classifier (Golland and Fischl, 2003; Golland et al., 2005; Hsing et al., 2003; Jensen, 1992; Molinaro et al., 2005). Essentially, the permutation test procedure measures how likely the observed accuracy would be obtained by chance. A p-value represents the fraction of random data sets under a certain null hypothesis where the classifier behaved as well as or better than in the original data.

Traditional permutation tests suggested in the recent literature 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 data set. This corresponds also to the most traditional statistical methods (Good, 2000), where the results on a control group are compared against the results on a treatment group. This simple test has been proven effective already for selecting relevant genes in small data samples (Maglietta et al., 2007) or for attribute selection in decision trees (Frank, 2000; Frank and Witten, 1998). However, the related literature has not performed extensive experimental studies for this traditional test in more general cases.

The goal of this paper is to study permutation tests for assessing the properties and performance of the classifiers. We first study the traditional permutation test for testing whether the classifier has found a real class structure, that is, a real connection between the data and the class 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 even if the class structure is weak.

We then propose a test for studying whether the classifier is exploiting dependency between some features for improving the classification accuracy. This second test is inspired by restricted randomization techniques traditionally used in statistics (Good, 2000). We study its relation to the traditional method both analytically and empirically. This new test can serve as a method for obtaining descriptive properties for classifiers, namely whether the classifier is using the feature dependency in the classification or not. For example, many existing classification algorithms are like black boxes whose functionality is hard to interpret directly. In such cases, indirect methods are needed to get descriptive information for the obtained class structure in the data.

If the studied data set is known to contain useful feature dependencies that increase the class separation, this new test can be used to evaluate the classifier against this knowledge. For example, often the data is gathered by a domain expert having deeper knowledge of the inner structure of the data. If the classifier is not using a known useful dependency, the classifier performance could be improved. For example, with medical data, if we are predicting the blood pressure of a person based on the height and the weight of the individual, the dependency between these two features is important in the classification as large body mass index is known to be connected with high blood pressure. However, both weight and height convey information about the blood pressure but the dependency between them is the most important factor in describing the blood pressure. Of course, in this case we could introduce a new feature, the body mass index, but in general, this may not be practical; for example, introducing too many new features can make the classification ineffective or too time consuming.

If nothing is known previously from the structure of the data, Test 2 can give some descriptive information for the obtained class structure. This information can be useful as such for understanding

the properties of the classifier, or it can guide the search towards an optimal classifier. For example, if the classifier is not exploiting the feature dependency, there might be no reason to use the chosen classifier as either more complex classifiers (if the data contains useful feature dependencies) or simpler classifiers (if the data does not contain useful feature dependencies) could perform better. Note, however, that not all feature dependencies are useful in predicting the class labels. Therefore, in the same way that traditional permutation tests have already been proven useful for selecting relevant features in some contexts as mentioned above (Maglietta et al., 2007; Frank, 2000; Frank and Witten, 1998), the new test can serve for selecting combinations of relevant features to boost the classifier performance for specific applications.

The idea is to provide users with practical p-values for the analysis of the classifier. The permutation tests provide useful statistics about the underlying reasons for the obtained classification result. Indeed, no test is better than the other, but all provide us with information about the classifier performance. Each p-value is a statistic 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).

The remaining of the paper is organized as follows. In Section 2, we give the background to classifiers and permutation-test p-values, and discuss connections with previous related work. In Section 3, we describe two simple permutation methods and study their behavior on the small toy example in Figure 1. In Section 4, we analyze in detail the properties of the different permutations and the effect of the tests for synthetic data on four different classifiers. In Section 5, we give experimental results on various real data sets. Finally, Section 6 concludes the paper.[1]

2. 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, for example, "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,

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

1. A shorter version of this paper appears in the proceedings of the IEEE International Conference on Data Mining (Ojala and Garriga, 2009). This is an improved version based on valuable comments by reviewers which includes: detailed discussions and examples, extended theoretical analysis of the tests including statistical power in special case scenarios, related work comparisons and a thorough experimental evaluation with large data sets.

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.

It has been recently argued that evaluating the classifier with an error measurement is ineffective for small amount of data samples (Braga-Neto and Dougherty, 2004; Golland et al., 2005; Hsing et al., 2003; Isaksson et al., 2008). Also classical generalization bounds are inappropriate 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 $e(f,D)$ in order to understand better the behavior of the classifier. For example, even if a consistent algorithm produces a classifier with low error, the data itself may have no structure.

Recently, a number of papers have suggested to use permutation-based *p*-values for assessing the competence of a classifier. Essentially, the permutation test procedure is used to obtain a *p*-value statistic from a null distribution of data samples, as described in Definition 1. In Section 3.1 we will introduce two different null hypotheses for the data.

Definition 1 (Permutation-based *p*-value) *Let $\widehat{D}$ be a set of k randomized versions D' of the original data D sampled from a given null distribution. The empirical p-value for the classifier f is calculated as follows (Good, 2000),*[2]

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

The empirical *p*-value of Definition 1 represents the fraction of randomized samples where the classifier behaved better in the random data than in the original data. Intuitively, it 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. Therefore, if the *p*-value is small enough—usually under a certain threshold, for example, $\alpha = 0.05$—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, that is, the null hypothesis is rejected.

Ideally the entire set of randomizations of *D* should be used to calculate the corresponding permutation-based *p*-value. This is known as the *exact randomization test*; unfortunately, this is computationally infeasible in data that goes beyond toy examples. Instead, we will sample from the set of all permutations to approximate this *p*-value. It is known that the Monte Carlo approximation of the *p*-value has a standard deviation of $\sqrt{\frac{p(1-p)}{k}}$, see, for example, Efron and Tibshirani (1993) and Good (2000), where p is the underlying true *p*-value and k is the number of samples used. Since p is unknown in practice, the upper bound $\frac{1}{2\sqrt{k}}$ is typically used to determine the number of samples required to achieve the desired precision of the test, or the value of the standard deviation in the critical point of $p = \alpha$ where α is the significance level. Alternatively, a sequential probability ratio test can be used (Besag and Clifford, 1991; Wald, 1945; Fay et al., 2007), where we sample randomizations of *D* until it is possible to accept or reject the null hypothesis. With these tests, often already 30 samples are enough for statistical inference with significance level $\alpha = 0.05$.

2. Notice the addition of 1 in both the denominator and the numerator of the definition. This adjustment is an standard procedure to compute empirical *p*-values and it is justified by the fact that the original database *D* is as well a randomized version of itself.

We will specify with more details in the next section how the randomized versions of the original data D are obtained. Indeed, this is an important question as each randomization method entails a certain null distribution, that is, which properties of the original data are preserved in the randomization test, directly affecting the distribution of the error $e(f, D')$. In the following, we will assume that the number of samples k is determined by any of the standard procedures just described here.

2.1 Related Work

As mentioned in the introduction, using permutation tests for assessing the accuracy of a classifier is not new, see, for example, Golland and Fischl (2003), Golland et al. (2005), Hsing et al. (2003) and Molinaro et al. (2005). The null distribution in those works is estimated by permuting labels from the data. This corresponds also to the most traditional statistical methods (Good, 2000), where the results on a control group are compared against the results on a treatment group. This traditional null hypothesis is typically used to evaluate one single classifier at a time (that is, one single model) and we will call it as Test 1 in the next section where the permutation tests are presented.

This simple traditional test has already been proven effective for selecting relevant genes in small data samples (Maglietta et al., 2007) or for attribute selection in decision trees (Frank, 2000; Frank and Witten, 1998). Particularly, the contributions by Frank and Witten (1998) show that permuting the labels is useful for testing the significance of attributes at the leaves of the decision trees, since samples tend to be small. Actually, when discriminating attributes for a decision tree, this test is preferable to a test that assumes the chi-squared distribution.

In the context of building effective induction systems based on rules, permutation tests have been extensively used by Jensen (1992). The idea is to construct a classifier (in the form of a decision tree or a rule system) by searching in the space of several models generated in an iterative fashion. The current model is tested against other competitors that are obtained by local changes (such as adding or removing conditions in the current rules). This allows to find final classifiers with less over-fitting problems. The evaluation of the different models in this local search strategy is done via permutation tests, using the framework of multiple hypothesis testing (Benjamini and Hochberg, 1995; Holm, 1979). The first test used corresponds to permuting labels—that is, Test 1—while the second test is a conditional randomization test. Conditionally randomization tests permute the labels in the data while preserving the overall classification ability of the current classifier. When tested on data with a conditionally randomized labelling, the current model will achieve the same score as it does with the actual labelling, although it will misclassify different observations. This conditionally randomization test is effective when searching for models that are more adaptable to noise.

The different tests that we will contribute in this paper could be as well used in this process of building an effective induction system. However, in general our tests are not directly comparable to the conditional randomization tests of Jensen (1992) in the context of this paper. We evaluate the classifier performance on the different randomized samples, and therefore, creating data set samples that preserve such performance would only produce always p-values close to one.

The restricted randomization test that we will study in detail later, can be used for studying the importance of dependent features for the classification performance. Related to this, group variable selection is a method for finding similarities between the features (Bondell and Reich, 2008). In that approach, similar features are grouped together for decreasing the dimensionality and improving the classification accuracy. Such methods are good for clustering the features while

doing classification. However, our aim is to test whether the dependency between the features is essential in the classification and not to reduce the dimensionality and similarities, thus differing from the objective of group variable selection.

As part of the related work we should mention that there is a large amount of statistical literature about hypothesis testing (Casella and Berger, 2001). Our contribution is to use the framework of hypothesis testing for assessing the classifier performance by means of generating permutation-based p-values. How the different randomizations affect these p-values is the central question we would like to study. Also sub-sampling methods such as bootstrapping (Efron, 1979) use randomizations to study the properties of the underlying distribution, but this is not used for testing the data against some null model as we intend here.

3. 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. The questions for which the two statistical tests supply answers can be summarized as follows:

Test 1: Has the classifier found a significant class structure, that is, a real connection between the data and the class labels?

Test 2: Is the classifier exploiting a significant dependency between the features to increase the accuracy of the classification?

Note, that these tests study whether the classifier is using the described properties and not whether the plain data contain such properties. For studying the characteristics of a population represented by the data, standard statistical test could be used (Casella and Berger, 2001).

Let π be a permutation of n elements. 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 $X = [X^1, X^2, \ldots, X^m]$.

3.1 Two Simple Permutation Methods

The first permutation method is the standard permutation test used in statistics (Good, 2000). 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 data set and let π be a permutation of n elements. 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, that is, obtaining a small p-value, rejects the null hypothesis that the features and the labels are independent, meaning that there is no difference between the classes. Let us now study this by considering the following case analysis. If the original data contains a real (i.e., not a random effect) dependency between data points and labels, then: (1) a significant classifier f will use such information to achieve a good classification accuracy and this will result in a small p-value (because the randomized samples do not contain such dependency

by construction); (2) if the classifier f is not significant in the sense of Test 1 (that is, f was not able to use the existing dependency between data and labels in the original data), then the p-value would tend to be high because the error in the randomized data will be similar to the error obtained in the original data. Finally, if the original data did not contain any real dependency between data points and labels, that is, such dependency was similar to randomized data sets, then all classifiers tend to have a high p-value. However, as a nature of statistical tests, about α of the results will be incorrectly regarded as significant.

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 additionally some dependency structure in the data that is not checked by Test 1. Indeed, it is very often the case that the p-values obtained from Test 1 are very small on real data because a classifier is easily regarded as significant even if the class structure is weak. We will provide more evidence about this fact in the experiments.

An important point is in fact, that a good classifier can be using other types of dependency if this exists in the data, for example the dependency between the features. From this perspective, Test 1 does not generate the appropriate randomized data sets to test such hypotheses. Therefore, we propose a new test whose aim is to check for the dependency between the attributes and how classifiers use such information.

The second null hypothesis assumes that the columns in X are mutually independent inside a class, thus $p(X(c)) = p(X(c)^1) \cdots p(X(c)^m)$, where $X(c)$ represents the submatrix of X that contains all the rows having the class label $c \in \mathcal{Y}$. This can be stated also using conditional probabilities, that is, $p(X \mid y) = p(X^1 \mid y) \cdots p(X^m \mid y)$. Test 2 is inspired by the restricted randomizations from statistics (see, e.g., Good, 2000).

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 of X in class label c, that is, $X(c) = \{X_i \mid y_i = c\}$ of size $l_c \times m$.*
- *Let $\pi_1, \ldots, \pi_m$ be m independent permutations of l_c elements.*
- *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 $X' = \{X(c)' \mid c \in Y\}$ 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. Notice that we are not testing the data but the classifier against the null hypothesis corresponding to Test 2. The classification result is significant with Test 2 only if the classifier exploits the interdependency between the features, if such interdependency exists. If the dependency is not used, there might be no reason to use a complicated classifier, as simpler and faster methods, such as Naive Bayes, could provide similar accuracy results for the same data. On

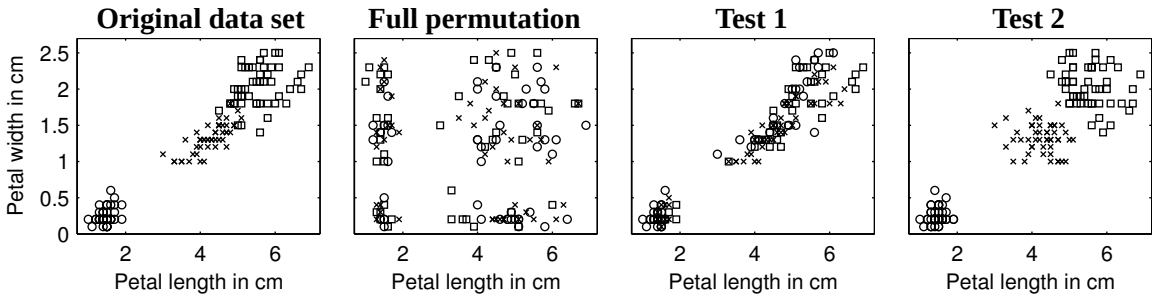


Figure 2: Scatter plots of original Iris data set and randomized versions for full permutation of the data and for Tests 1 and 2 (one sample for each test). The data points belong to three different classes denoted by different markers, and they are scattered against petal length and width in centimeters.

the other hand, this observation can lead us to find a classifier that can exploit the possibly existing dependency and thus improve the classification accuracy further, as discussed in the introduction.

There are three important properties of the permutation-based p-values and the two tests proposed here. The first one is that the number of missing values, that is, the number of entries in D that are empty because they do not have measured values, will be distributed equally across columns in the original data set D and the randomized data sets D'; this is necessary for a fair p-value computation. The second property is that the proposed permutations are always relevant regardless of the data domain, that is, values are permuted always within the same column, which does not change the domain of the randomized data sets. Finally, we have that unbalanced data sets, that is, data sets where the distribution of class labels is not uniform, remain equally unbalanced in the randomized samples.

In all, with permutation tests we obtain useful statistics about the classification result. No test is better than the other, but all provide us with information about the classifier. Each p-value is a statistic 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).

In Figure 2, we give as an example one randomization for each test on the well-known Iris data set. We show here the projection of two features, before and after randomizations according to each one of the tests. For comparison, we include a test corresponding to full permutation of the data where each column is permuted separately, breaking the connection between the features and mixing the values between different classes. Note how well Test 2 has preserved the class structure compared to other tests. To provide more intuitions, in this case a very simple classifier, which predicts the class by means of one single of these two features would suffice in reaching a very good accuracy. In other words, the dependency between the two features is not significant as such, so that a more complex classifier making use of such dependency would end up having a high p-value with Test 2. We will discuss the Iris data more in the experiments.

3.2 Handling Instability of the Error

A related issue for all the above presented tests concerns the variability of the error estimate returned by a classifier. Indeed, applying the same classifier several times over the original data set D can

return different error estimates $e(f,D)$ if, for example, 10-fold cross-validation is used. So the question is, how can we ensure that the p-values given by the tests are stable to such variance?

The empirical p-value depends heavily on the correct estimation of the original classification accuracy, whereas the good estimation of the classification errors of the randomized data sets is not so important. However, exactly the same classification procedure has to be used for both the original and randomized data for the p-value to be valid. Therefore, we propose the following solution to alleviate the problem of having instable test statistic: We train the classifier on the original data r times, thus obtaining r different error estimates $E = \{e_1(f,D),\ldots,e_r(f,D)\}$ on D. Next, we obtain k randomized samples of D according to the desired null hypothesis and compute the p-value for each one of those original errors $e \in E$. We obtain therefore r different p-values by using the same k randomized data sets for each computation. We finally output the average of those r different p-values as the final empirical p-value.

Note that in total we will compute the error of the classifier $r+k$ times: r times on the original data and one time for each of the k randomized data sets. Of course, the larger the k and the larger the r, the more stable the final averaged p-value would be. A larger r decreases the variance in the final p-value due to the estimation of the classification error of the original data set whereas a larger k decreases the variance in the final p-value due to the random sampling from the null distribution. In practice, we have observed that a value of $r = 10$ and $k = 100$ produce sufficiently stable results.

This solution is closely related to calculating the statistic ρ, or calculating the test statistic U of the Wilcoxon-Mann-Whitney two-sample rank-sum test (Good, 2000). However, it is not valid to apply these approaches in our context as the r classification errors of the original data are not independent of each other. Nevertheless, the proposed solution has the same good properties as the ρ and U statistics as well as it generalizes the concept of empirical p-value to instable results.

A different solution would be to use a more accurate error estimate. For example, we could use leave-one-out cross-validation or cross-validation with 100 folds instead of 10-fold cross-validation. This will decrease the variability but increase the computation time dramatically as we need to perform the same slow classification procedure to all k randomized samples as well. However, it turns out that the stability issue is not vital for the final result; our solution produces sufficiently stable p-values in practice.

3.3 Example

We illustrate the concept of the tests by studying the small artificial example presented in the introduction in Figure 1. Consider the two data sets D_1 and D_2 given in Figure 1. The first data set D_1 was generated as follows: in the first eight rows corresponding to class +, each element is independently sampled to be x with probability 80% and o otherwise; in the last eight rows the probabilities are the other way around. Note that in the data set D_1 the features are independent given the class since, for example, knowing that $X_i^{j_1} =$ x inside class + does not increase the probability of $X_i^{j_2}$ being x. The data set D_2 was generated as follows: the first four rows contain x, the second four rows contain o, the third four rows contain x in the first four columns and o in the last four columns, and the last four rows contain o in the first four columns and x in the last four columns; finally, 10% of noise was added to the data set, that is, each x was flipped to o with probability of 10%, and vice versa.

Observe that both D_1 and D_2 have a clear separation into the two given classes, + and –. However, the structure inside the data set D_1 is much simpler than in the data set D_2. For illustration

1-**Nearest Neighbor**					
	Orig.	Test 1		Test 2	
Data Set	Err.	Err. (Std)	p-val.	Err. (Std)	p-val.
D_1	0.00	0.53 (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

Table 1: Average error and p-value for Test 1 and Test 2 when using the 1-Nearest Neighbor classifier to data sets of Figure 1.

purposes, we analyze this with the 1-Nearest Neighbor classifier using the leave-one-out cross-validation given in Equation (1). Results for Test 1 and Test 2 are summarized in Table 1. The classification error obtained in the original data is 0.00 for both D_1 and D_2, which is expected since the data sets were generated to contain clear class structure.

First, we use the standard permutation test (i.e., permuting labels, Test 1) to understand the behavior under the null hypothesis where data points and labels are independent. We produce 1000 random permutations of the class labels for both the data sets D_1 and D_2, and perform the same leave-one-out cross-validation procedure to obtain a classification error for each randomized data set. On the randomized samples of data set D_1 we obtain an average classification error of 0.53, a standard deviation 0.14 and a minimum classification error of 0.13. For the randomized data from D_2 the corresponding values are 0.53, 0.14 and 0.19, respectively. These values result in two empirical p-values of both 0.001 on both the data sets D_1 and D_2. Thus, we can say that the classifiers are significant under the null hypothesis that data and labels are independent. That is, the connection between the data and the class labels is real in both data sets and the 1-Nearest Neighbor classifier is able to find that connection in both data sets, resulting into a good classification accuracy.

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. To get more information of the properties of the classifiers, we study next the performance of the classifiers by taking into account the inner structure of data sets D_1 and D_2 by applying Test 2. Again, we produce 1000 random samples of the data sets D_1 and D_2 by permuting each column separately inside each class. The same leave-one-out cross-validation procedure is performed for the randomized samples, obtaining for the data set D_1 the average classification error of 0.06, standard deviation of 0.06 and a minimum value of 0.00. For the data set D_2 the corresponding values are 0.62, 0.14 and 0.19, respectively. Therefore, under Test 2 the empirical p-values are 0.358 for the data set D_1 and 0.001 for the data set D_2.

We can say that, for Test 2, the 1-Nearest Neighbor classifier is significant for data set D_2 but not for data set D_1. Indeed, the data set 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. Note, however, that none of the features in the data set D_1 is sufficient alone to correctly classify all the samples due to the noise in the data set. Thus using a combination of multiple features for classification is necessary for obtaining a good accuracy, even though the features are independent of each other. For data set D_2 we have that the dependency between the columns inside the classes is essential for the good classification result, and in this case, the 1-Nearest Neighbor classifier has been able to exploit that information.

4. Analysis

In this section we analyze the properties of the tests and demonstrate the behavior of the different *p*-values on simulated data. First, we state the relationships between the different sets of permutations.

4.1 Connection between Test 1 and Test 2

Remember that the random samples from Test 1 are obtained by permuting the class labels and the samples from Test 2 by permuting the features inside each class. To establish a connection between these randomizations, we study the randomization where each data column is permuted separately, regardless of the class label. This corresponds to the full permutation presented in Figure 2 in Section 3.1 for Iris data set. It breaks the connection between the features, and furthermore, between the data and the class labels. The following result states the relationship between Test 1, Test 2 and the full permutation method.

Proposition 2 *Let $\Pi_l(D)$, $\Pi_c(D)$, $\Pi_{cc}(D)$ be the sets of all possible randomized data sets obtained from D via permuting labels (Test 1), permuting data columns (full permutation), or permuting data columns inside class (Test 2), respectively. The following holds,*

(1) $\Pi_l(D) \subset \Pi_c(D)$

(2) $\Pi_{cc}(D) \subset \Pi_c(D)$

(3) $\Pi_l(D) \neq \Pi_{cc}(D)$

Note that $\Pi_l(D)$, $\Pi_c(D)$ and $\Pi_{cc}(D)$ refer to sets of data matrices. Therefore, we have that permuting the data columns is the randomization method producing the most diverse samples, while permuting labels (Test 1) and permuting data within class (Test 2) produce different randomized samples.

Actually, the relationship stated by Proposition 2 implies the following property: the *p*-value obtained by permuting the data columns is typically smaller than both the *p*-values obtained from Test 1 and Test 2. The reason is that all the randomized data sets obtained by Test 1 and Test 2 can also be obtained by permuting data columns and the additional randomized data sets obtained by permuting the columns are, in general, even more random. Theoretically, permuting the data columns is a combination of Test 1 and Test 2, and thus, it is not a useful test. In practice, we have observed that the *p*-value returned by permuting the data columns is very close to the *p*-value of Test 1, which tends to be much smaller than the *p*-value of Test 2.

Considering Proposition 2, it makes only sense to restrict the randomization to classes by using Test 2, whenever Test 1 has produced a small *p*-value. That is, it is only reasonable to study whether the classifier uses feature dependency in separating the classes if it has found a real class structure.

4.2 Behavior of the Tests

To understand better the behavior of the tests, we study generated data where correlation is used as the dependency between the features. Consider the following simulated data, inspired by the data used by Golland et al. (2005): 100 data points are generated from two-dimensional normal distribution with mean vector (1,0), unit variances and covariance $\rho \in [-1,1]$. Another 100 data points are generated from similar normal distribution with mean $(-1,0)$, unit variances and same

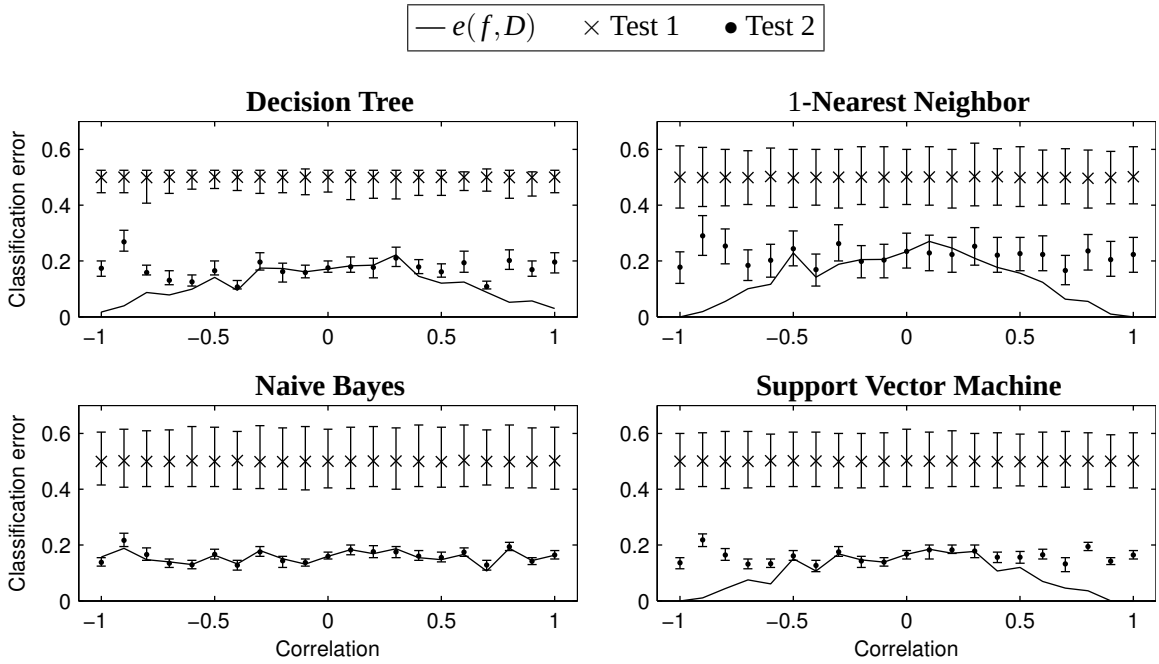


Figure 3: Average values of stratified 10-fold cross-validation error (y-axis) for varying values of correlation between the features per class (x-axis). The solid line shows the error on the original data, and symbols $\times$ and $\bullet$ represent the average of the error on 1000 randomized samples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

covariance ρ. The first 100 samples are assigned with class label $y = +1$ with probability $1-t$ and $y = -1$ with probability t. For the other 100 samples the probabilities are the opposite. The probability $t \in [0, 0.5]$ represents the noise level. When $t = 0.5$, there is no class structure at all. Note that the correlation between the features improves the class separation: if the correlation $\rho = 1$ and the noise $t = 0$, 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 data sets (with varying parameters of noise and correlation) we use as an error estimate 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 (Witten and Frank, 2005) with the default parameters of the implementations 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. Tuning the parameters of these algorithms is not in the scope of this paper; our objective is to show the behavior of the discussed p-values for some selected classifiers.

Figure 3 shows the behavior of the classifiers on data sets without class noise, $t = 0$, and with the correlation ρ between features inside classes varying from -1 (negative correlation) to 1 (positive correlation). The solid line corresponds to $e(f,D)$, that is, the error of the classifier in the original

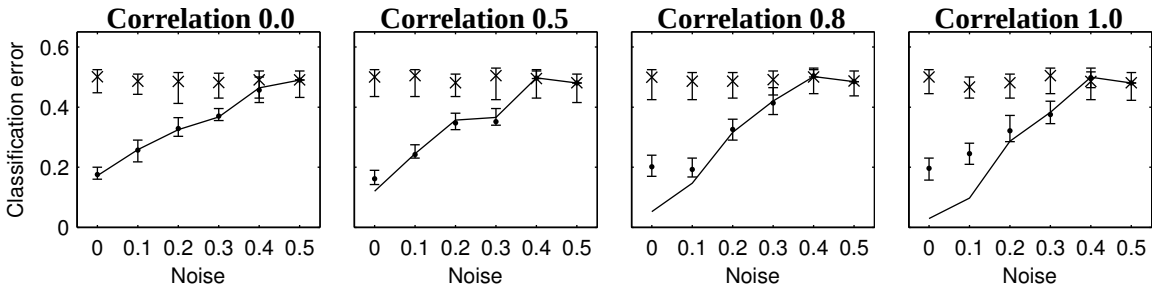


Figure 4: Average values of stratified 10-fold cross-validation error (*y*-axis) for the Decision Tree classifier when noise varies on the original data set (*x*-axis) with four fixed correlation values between the features inside the classes. The solid line shows the error on the original data, and symbols $\times$ and $\bullet$ show the average error on 1000 randomized samples from Test 1 and Test 2, respectively. Each average of the error on the randomized samples $\times$ and $\bullet$ is depicted together with the $[1\%, 99\%]$-deviation bar below which the associated null hypothesis is rejected with significance level $\alpha = 0.01$.

data. The symbols "$\times$" and "$\bullet$" represent the average error obtained by the classifier on 1000 randomized samples from Test 1 and Test 2, respectively. When the solid line of $e(f, D)$ falls below the $[1\%, 99\%]$-deviation bars, the corresponding associated null hypothesis is rejected with significance level $\alpha = 0.01$. Actually, the correspondence between the confidence intervals and hypotheses testing is only approximately true since the definition of empirical *p*-value contains the addition of 1 in both the numerator and denominator. However, the practical difference is negligible.

First, note that the Decision Tree, 1-Nearest Neighbor and Support Vector Machine classifiers have been able to exploit the dependency between the features, that is, 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 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 magnitude of 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 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 data sets, which results in a very high *p*-value. Naive Bayes classifier is an example of such classifiers which are not able to use the dependency between the features at all. Thus applying Test 2 for Naive Bayes classifier will practically always produce a high *p*-value irrespective of the data.

Finally, Figure 4 shows the behavior of the Decision Tree classifier when the noise $t \in [0, 0.5]$ is increased on the x-axis. We also vary the correlation ρ between the features per class and show the results on four cases: zero correlation, 0.5, 0.8 and total correlation. We observe that as the noise increases the p-values tend to be larger. Therefore, it is more difficult to reject the null hypothesis on very noisy data sets, that is, when the class structure is weak. This is true for both Test 1 and Test 2. However, Test 1 rejects the null hypothesis even if there is 30% of noise. This supports the fact already observed in related literature (Golland et al., 2005), that even a weak class structure is easily regarded as significant with Test 1. Compared to this, Test 2 gives more conservative results.

4.3 Power Analysis of Test 2

The *power* of a statistical test is the probability that the test will reject the null hypothesis when the alternative hypothesis is true. The power of the test depends on how much or how clearly the null hypothesis is false. For example, in our case with Test 2, a classifier may rely solely on a strong dependency structure between some specific features in the classification, or it may use a weak feature dependency to slightly improve the classification accuracy. Rejecting the null hypothesis of Test 2 is much easier in the former than in the latter case. Note, however, that a strong dependency between the features is not always useful in separating the classes, as seen in Figure 2 with Iris data set. So, the question with Test 2 is whether the classifier is exploiting some of the dependency structure between the features in the data and how important such feature dependency is for the classification of the data.

In general, the power of the test can only be analyzed in special cases. Nevertheless, such analysis can give some general idea of the power the test. Next, we present a formal power analysis in the particular case where we vary the correlation between the features that is useful in separating the classes from each other. Note, however, that there exist also other types of dependency than correlation. The amount of correlation is just easy to measure, thus being suitable for formal power analysis.

We present the power analysis on similar data as studied in Section 4.2. The results in the previous subsection can be seen as informal power analysis. In summary, we observed that when the magnitude of the correlation in the data studied in Section 4.2 was larger than about 0.5 and the classifier was exploiting the feature dependency, that is, a classifier different from Naive Bayes, Test 2 was able to reject the null hypothesis. However, based on the data it is clear that even smaller correlations increased the class separation and were helpful in classifying the data but Test 2 could not regard such improvement as significant. The following analysis supports these observations.

Let the data set X consist of n points with two features belonging to two classes, $+1$ and -1. Let a point $x \in X$ be in class $y = +1$ with probability 0.5 and in class $y = -1$ with probability 0.5. Let the point $x \in X$ be sampled from two-dimensional normal distribution with mean $(0,0)$, unit variances and covariance $y\rho$ where $\rho \in [0,1]$ is a given parameter. Thus, in the first class, $y = +1$, the correlation between the two features is positive and in the second class, $y = -1$, it is negative. Compared to the data sets in Section 4.2, now the covariance changes between the classes, not the mean vector. An optimal classifier assigns a point $x \in X$ to class $y = +1$ if $x_1 x_2 > 0$ and to class $y = -1$ if $x_1 x_2 < 0$, where x_i is the i-th feature of the vector x.

The null hypothesis of Test 2 is that the classifier is not exploiting the dependency between the features in classification. To alleviate the power analysis, we assume that the classifier is able to find the optimal classification, that is, it assigns the point x to class $\text{sgn}(x_1 x_2)$ where $\text{sgn}(\cdot)$ is

the signum function. If the classifier is not optimal, it will just decrease the power of the test. The nonoptimality of the classifier could be taken into account by introducing a probability t for reporting a nonoptimal class label; this approach is used in the next subsection for power analysis of Test 1 but is left out here for simplicity in the analysis. Under this optimality scenario, the probability of correctly classifying a sample is

$$\begin{aligned}\Pr(\mathrm{sgn}(x_1x_2)=y) &= \frac{1}{2}\Pr(x_1x_2>0 \mid y=+1)+\frac{1}{2}\Pr(x_1x_2<0 \mid y=-1)\\ &= \Pr(x_1x_2>0 \mid y=+1) = 2\int_0^\infty\int_0^\infty \Pr(x_1,x_2)\,\mathrm{d}x_1\mathrm{d}x_2\\ &= 2\int_0^\infty\int_0^\infty \frac{1}{2\pi\sqrt{1-\rho^2}}\exp\left[-\frac{x_1^2-2\rho x_1x_2+x_2^2}{2(1-\rho^2)}\right]\mathrm{d}x_1\mathrm{d}x_2\\ &= \frac{1}{2}+\frac{1}{\pi}\arcsin\rho, \end{aligned} \tag{2}$$

where $\Pr(x_1,x_2)$ is just the standardized bivariate normal distribution. The null hypothesis corresponds to the case where the correlation parameter is zero, $\rho=0$, that is, no feature dependency exists. In that case, the probability of correctly classifying a sample is $1/2$.

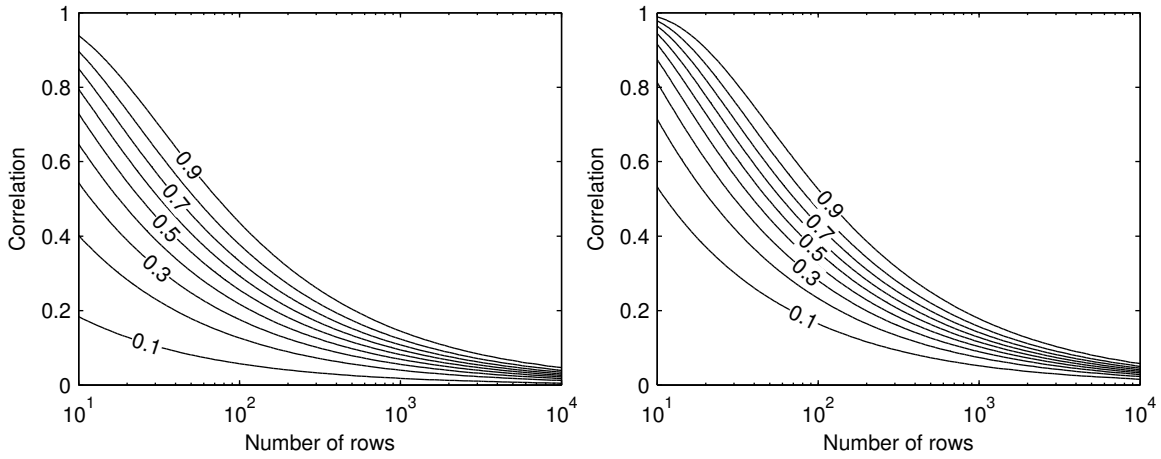
In our randomization approach, we are using classification error as the test statistic. Since we assume that the optimal classifier is given, we use all the n points of the data set X for testing the classifier and calculating the classification error. Under the null hypothesis H_0 and under the alternative hypothesis H_1 of Test 2, the classification errors $e(f \mid H_0)$ and $e(f \mid H_1)$ are distributed as follows:

$$\begin{aligned} n\cdot e(f\mid H_0) &\sim \mathrm{Bin}\left(n,\frac{1}{2}\right)\approx \mathcal{N}\left(\frac{n}{2},\frac{n}{4}\right),\\ n\cdot e(f\mid H_1) &\sim \mathrm{Bin}\left(n,\frac{1}{2}-\frac{1}{\pi}\arcsin\rho\right)\approx \mathcal{N}\left(\frac{n}{2}-\frac{n}{\pi}\arcsin\rho,\frac{n}{4}-\frac{n}{\pi^2}\arcsin^2\rho\right),\end{aligned}$$

where $\frac{1}{2}-\frac{1}{\pi}\arcsin\rho$ is the probability of incorrectly classifying a sample by Equation (2). The normal approximation $\mathcal{N}(np,np(1-p))$ of a binomial distribution $\mathrm{Bin}(n,p)$ holds with good accuracy when $np>5$ and $n(1-p)>5$. In our case, the approximation is valid if $n(\frac{1}{2}-\frac{1}{\pi}\arcsin\rho)>5$. This holds, for example, if $n\geq 20$ and $\rho\leq 0.7$.

Now the power of Test 2 for this generated data is the probability of rejecting the null hypothesis H_0 of $\rho=0$ with significance level α when the alternative hypothesis H_1 is that the correlation $\rho>0$. Note that we are implicitly assuming that the classifier is optimal, that is, we are excluding the classifier quality from the power analysis. Thus, the power is the probability that $e(f \mid H_1)$ is smaller than $1-\alpha$ of the errors $e(f \mid H_0)$ under the alternative hypothesis H_1:

$$\begin{aligned}\mathrm{Power} &= \Pr\left(e(f\mid H_1)<F^{-1}_{e(f|H_0)}(\alpha)\right)\\ &\approx \Pr\left(\frac{1}{2}-\frac{1}{\pi}\arcsin\rho+\sqrt{\frac{1}{4n}-\frac{1}{n\pi^2}\arcsin^2\rho}\cdot Z<\frac{1}{2}+\frac{1}{2\sqrt{n}}\Phi^{-1}(\alpha)\right)\\ &= \Phi\left(\frac{2\sqrt{n}\arcsin\rho+\pi\Phi^{-1}(\alpha)}{\sqrt{\pi^2-4\arcsin^2\rho}}\right),\end{aligned} \tag{3}$$



(a) $\alpha = 0.05$ (b) $\alpha = 0.01$

Figure 5: Contour plots of the statistical power of Test 2 as a function of the number of rows n in the generated data set and the correlation parameter ρ. Each solid line corresponds to a constant value of the power that is given on top of the contour. The power values are calculated by Equation (3) for two different values of significance level α.

where $F_{e(f|H_0)}$ is the cumulative distribution function of $e(f \mid H_0)$, Z is a random variable following standard normal distribution and Φ is the cumulative distribution function of the standard normal distribution. Note that we are using exact p-value instead of empirical p-value, effectively leaving out the influence of variance by using k randomized samples; see Fay et al. (2007) for analysis of resampling risk of using k samples. However, this has little effect to the power of the test. When the correlation $\rho = 0$, the power is α, that is, when the null hypothesis is true, it is rejected incorrectly about α of the times. Therefore, α is really the significance level of the tests.

In Figure 5 we present contour plots of the statistical power in Equation (3) for different values of the two varying parameters. As expected, the higher the correlation ρ and the number of rows n are, the higher the statistical power of Test 2 is. For example, if the data set contains about 1000 rows, we can infer with 90% probability that the classifier is exploiting the feature dependency of approximately a correlation of 0.2 in the data. The results are also in line with the results from Section 4.2 although the studied data sets are slightly different. When the significance level used is $\alpha = 0.01$ we can infer that the classifier is exploiting the feature dependency of correlation larger than 0.4 approximately 90% of the times when the data set has 200 rows.

Notice that if we had not considered an "optimal" classifier, that is, if we had introduced a probability t of assigning each observation to the incorrect label, then Equation (3) would depend on three parameters. In that case, the higher t, the smaller is the power of the test; however, for a fixed t we still would observe the same behaviour as in the contourplots above: the higher the correlation ρ and the larger the n, the higher is the statistical power of Test 2. The error parameter t is taken into account in the next section, where the power analysis of Test 1 does not depend on ρ between the features.

4.4 Power Analysis of Test 1

Let the data set X consist of n observations belonging to two different classes with equal probability. We assume that we have a classifier f whose error rate is $t \in [0,1]$, that is, the classifier assigns each observation to the correct class with probability $1-t$. Another way to see this is that the classifier f is optimal but the original class label of each point is erroneous with probability t. We perform power analysis of Test 1 for this general form of data.

Note that the results in Section 4.2 can be seen as informal power analysis of Test 1 on similar setting as studied here. The results in Figure 4 can be summarized as follows. When the error rate was smaller than $t < 0.4$, Test 1 was able to reject the null hypotheses. Note, however, that the error rate t used in this section is not directly comparable to the error rate used in Section 4.2.

The power analysis of Test 1 proceeds similarly as in the previous subsection for Test 2. Under the null hypothesis H_0 and under the alternative hypothesis H_1 of Test 1, the classification errors $e(f \mid H_0)$ and $e(f \mid H_1)$ are distributed as follows:

$$
\begin{aligned}
n \cdot e(f \mid H_0) &\sim \text{Bin}\left(n, \frac{1}{2}\right) \approx \mathcal{N}\left(\frac{n}{2}, \frac{n}{4}\right), \\
n \cdot e(f \mid H_1) &\sim \text{Bin}\,(n,t) \approx \mathcal{N}\left(nt, nt(1-t)\right).
\end{aligned}
$$

The null hypothesis H_0 assumes that there is no connection between the data and the class labels thus the probability of incorrect classification is $1/2$ as the classes are equally probable. Note that the null hypothesis corresponds to the case where the error rate of the classifier f is $t = 1/2$.

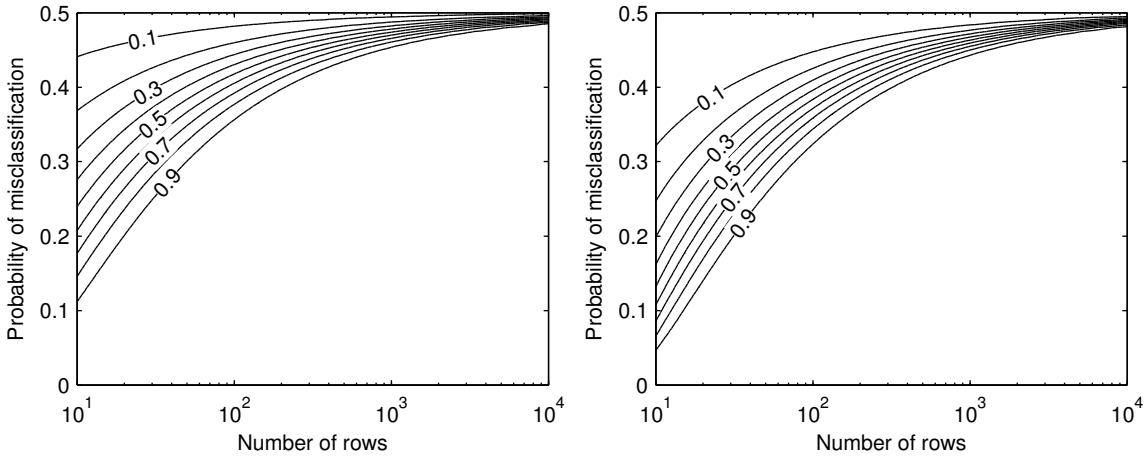
Now the power of Test 1 is the probability of rejecting the null hypothesis H_0 with significance level α when the alternative hypothesis H_1 is true, that is,

$$
\begin{aligned}
\text{Power} &= \Pr\left(e(f \mid H_1) < F^{-1}_{e(f|H_0)}(\alpha)\right) \\
&\approx \Pr\left(t + \sqrt{\frac{t(1-t)}{n}}\, Z < \frac{1}{2} + \frac{1}{2\sqrt{n}}\, \Phi^{-1}(\alpha)\right) \\
&= \Phi\left(\frac{(1-2t)\sqrt{n} + \Phi^{-1}(\alpha)}{2\sqrt{t(1-t)}}\right), \qquad (4)
\end{aligned}
$$

where the same notation as in the previous subsection is used. First, note that when the null hypothesis is true, that is, $t = 1/2$, the power of Test 1 calculated by Equation (4) equals the significance level α as it should.

In Figure 6 we present contour plots of the statistical power of Test 1 calculated by Equation (4) for different values of parameters. As expected, when the number of observations n increases or the error rate t decreases, the power increases. Furthermore, the larger the significance level α is, the larger the power of Test 1 is. When the parameter values are $\alpha = 0.01$, $n = 200$ and $t = 0.4$, the power of Test 1 is about 0.7 that is comparable to the results in Section 4.2.

In this section, we analyzed the behaviour and the power of the tests. Note that although we used correlation as the only type of dependency between features in this section, there exist also other forms of dependency that the classifier can exploit. As conclusions from the power analysis, the more rows the data set has, the easier we can infer that the classifier is using the feature dependency or some other properties in the data.



(a) $\alpha = 0.05$ (b) $\alpha = 0.01$

Figure 6: Contour plots of the statistical power of Test 1 as a function of the number of rows n in the generated data set and the probability of misclassification t. Each solid line corresponds to a constant value of the power that is given on top of the contour. The power values are calculated by Equation (4) for two different values of significance level α.

5. Empirical Results

In this section, we give extensive empirical results on 33 various real data sets from UCI machine learning repository (Asuncion and Newman, 2007). Basic characteristics of the data sets are described in Table 2. The data sets are divided into three categories based on their size: small, medium and large. Some data sets contain only nominal or numeric features whereas some data sets contain both kind of features (mixed). About one-third of the data sets contain also missing values. Notice that in most data sets the features are measured in different scales, thus it is not sensible to swap the values between different features. This justifies why it is only reasonable to consider column-wise permutations, and why some recent data mining randomization methods (Gionis et al., 2007; Ojala et al., 2009; Chen et al., 2005) are not generally applicable in assessing classification results.

In the experiments we use Weka 3.6 data mining software (Witten and Frank, 2005) that contains open source Java implementations of many classification algorithms. We use four different types of classification algorithms with the default parameters: Decision Tree, Naive Bayes, 1-Nearest Neighbor and Support Vector Machine classifier. The Decision Tree classifier is similar to C4.5 algorithm. The default kernel used with Support Vector Machine is linear. Missing values and the combination of nominal and numerical values are given as such as the input for the classifiers; the default approaches in Weka of the classifiers are used to handle these cases. Notice that tuning the parameters of these algorithms is not in the scope of this paper; our objective is to show the behavior of the discussed p-values for some selected classifiers on various data sets.

We use different classification procedures and the number of randomized data sets for each of the different size categories of the data sets (small, medium and large). For small data sets, we use stratified 10-fold cross-validation error as the statistic and 1000 randomized data sets for calculating the empirical p-values. For medium-sized data sets, we use the same stratified 10-fold cross-validation error and 100 randomized data sets. Finally, for large data sets, we divide the data

	Data Set	Rows	Features	Classes	Missing	Domain
Small	Audiology	226	70	24	2.0%	nominal
	Autos	205	25	6	1.2%	mixed
	Breast	286	9	2	0.3%	nominal
	Glass	214	9	6	No	numeric
	Hepatitis	155	19	2	5.7%	mixed
	Ionosphere	351	34	2	No	numeric
	Iris	150	4	3	No	numeric
	Lymph	148	18	4	No	mixed
	Promoters	106	57	2	No	nominal
	Segment	210	19	7	No	numeric
	Sonar	208	60	2	No	numeric
	Spect	267	22	2	No	nominal
	Tumor	339	17	21	3.9%	nominal
	Votes	435	16	2	5.6%	nominal
	Zoo	101	17	7	No	mixed
Medium	Abalone	4177	8	28	No	mixed
	Anneal	898	38	5	65.0%	mixed
	Balance	625	4	3	No	numeric
	Car	1728	6	4	No	nominal
	German	1000	20	2	No	mixed
	Mushroom	8124	22	2	1.4%	nominal
	Musk	6598	166	2	No	numeric
	Pima	768	8	2	No	numeric
	Satellite	6435	36	6	No	numeric
	Spam	4601	57	2	No	numeric
	Splice	3190	60	3	No	nominal
	Tic-tac-toe	958	9	2	No	nominal
	Yeast	1484	8	10	No	numeric
Large	Adult	48842	15	2	0.9%	mixed
	Chess	28056	6	18	No	mixed
	Connect-4	67557	42	3	No	nominal
	Letter	20000	16	26	No	numeric
	Shuttle	58000	9	7	No	numeric

Table 2: Summary of 33 selected data sets from UCI machine learning repository (Asuncion and Newman, 2007). The data sets are divided into three categories based on their size: small, medium and large.

set into training set with 10 000 random rows and to test set with the rest of the rows. We use 100 randomized data sets for calculating the empirical p-values with large data sets. The reason for the smaller number of randomized samples for medium and large data sets is mainly computation time. However, 100 samples is usually enough for statistical inference. Furthermore, as seen in Section 4 the power of the tests is greater when the data sets have more rows, that is, with large data sets it is easier to reject the null-hypotheses, supporting the need of fewer randomized samples in hypothesis testing.

Since the original classification error is not a stable result due to the randomness in training the classifier and dividing the data set into test and train data, we perform the same classification procedure ten times for the original data sets and calculate an empirical p-value for each of the ten results. This was described in Section 3.2. We give the average value of these empirical p-values as well as the average value and the standard deviation of the original classification errors.

As we are testing multiple hypotheses simultaneously, we need to correct for multiple comparisons. We apply the approach by Benjamini and Hochberg (1995) to control the false discovery rate (FDR), that is, the expected proportion of results incorrectly regarded as significant. In the experiments, we restrict the false discovery rate below $\alpha = 0.05$ separately for Test 1 and Test 2. In the Benjamini-Hochberg approach, if $p_1, \ldots, p_m$ are the original empirical p-values in increasing order, the results $p_1, \ldots, p_l$ are regarded as significant where l is the largest value such that $p_l \leq \frac{l}{m}\alpha$.

The significance testing results for the Decision Tree classifier are given in Table 3, for Naive Bayes in Table 4, for 1-Nearest Neighbor classifier in Table 5 and finally for Support Vector Machine classifier in Table 6. The mean and the standard deviation of the 10 original classification errors are given as well as the mean and standard deviation of the errors on the 1000 or 100 randomized samples with Test 1 and Test 2. The empirical p-values corresponding to nonsignificant results, when the false discovery rate is restricted below 0.05, are in boldface in the tables. With all classifiers, the largest significant empirical p-value was 0.01. The smallest non-significant p-values were 0.03 with Decision Tree and 1-Nearest Neighbor classifiers, 0.08 with Naive Bayes classifier and 0.19 with Support Vector Machine classifier.

The results for the traditional permutation method Test 1 show that the classification errors with most data sets are regarded as significant. These results show that the data sets contain clear class structure. However, they do not give any additional insight for understanding the class structure in the data sets.

There are two reasons why the simple permutation test, Test 1, regards the class structure of the data sets as significant. Firstly, most of the data sets that are publicly available, as all the data sets used in this paper, have already passed some quality checks, that is, someone has already found some interesting structure in them. Secondly, and as a more important reason, the traditional permutation tests easily regard the results as significant even if there is only a slight class structure present because in the corresponding permuted data sets there is no class structure, especially if the original data set is large.

Furthermore, the few results which were regarded as nonsignificant with Test 1 are with such classifiers that have not performed well on the data. That is, the other classifiers have produced smaller classification errors on the same data sets, and, in contrast, these results are regarded as significant.

Next, we consider the results for permuting the features inside each class, that is, Test 2. The results show that there are actually now almost equal amount of nonsignificant and significant results with respect to Test 2. This means that in many data sets the original structure inside the classes is

		Decision Tree				
		Original	Test 1		Test 2	
	Data Set	Err. (Std)	Err. (Std)	*p*-val.	Err. (Std)	*p*-val.
Small	Audiology	0.22 (0.01)	0.82 (0.03)	0.001	0.23 (0.02)	**0.482**
	Autos	0.19 (0.01)	0.76 (0.04)	0.001	0.38 (0.04)	0.001
	Breast	0.26 (0.01)	0.30 (0.00)	0.001	0.29 (0.02)	**0.116**
	Glass	0.33 (0.02)	0.72 (0.03)	0.001	0.34 (0.03)	**0.457**
	Hepatitis	0.22 (0.02)	0.23 (0.02)	**0.319**	0.15 (0.03)	**0.955**
	Ionosphere	0.10 (0.01)	0.38 (0.02)	0.001	0.07 (0.01)	**0.964**
	Iris	0.05 (0.01)	0.67 (0.03)	0.001	0.05 (0.01)	**0.765**
	Lymph	0.22 (0.02)	0.51 (0.05)	0.001	0.23 (0.04)	**0.437**
	Promoters	0.21 (0.04)	0.50 (0.06)	0.002	0.22 (0.05)	**0.377**
	Segment	0.13 (0.02)	0.86 (0.03)	0.001	0.17 (0.02)	**0.132**
	Sonar	0.27 (0.02)	0.49 (0.03)	0.001	0.27 (0.03)	**0.507**
	Spect	0.19 (0.01)	0.22 (0.01)	0.004	0.15 (0.02)	**0.966**
	Tumor	0.58 (0.01)	0.82 (0.02)	0.001	0.60 (0.02)	**0.138**
	Votes	0.03 (0.00)	0.42 (0.02)	0.001	0.03 (0.01)	**0.791**
	Zoo	0.07 (0.01)	0.64 (0.03)	0.001	0.07 (0.01)	**0.593**
Medium	Abalone	0.79 (0.01)	0.89 (0.00)	0.01	0.67 (0.01)	**1.00**
	Anneal	0.07 (0.01)	0.24 (0.00)	0.01	0.13 (0.01)	0.01
	Balance	0.22 (0.01)	0.55 (0.02)	0.01	0.29 (0.02)	0.01
	Car	0.08 (0.00)	0.30 (0.00)	0.01	0.26 (0.01)	0.01
	German	0.29 (0.01)	0.32 (0.01)	0.01	0.28 (0.01)	**0.66**
	Mushroom	0.00 (0.00)	0.50 (0.01)	0.01	0.01 (0.00)	0.01
	Musk	0.03 (0.00)	0.16 (0.00)	0.01	0.09 (0.00)	0.01
	Pima	0.25 (0.01)	0.35 (0.01)	0.01	0.24 (0.01)	**0.67**
	Satellite	0.14 (0.00)	0.81 (0.00)	0.01	0.07 (0.00)	**1.00**
	Spam	0.07 (0.00)	0.40 (0.00)	0.01	0.06 (0.00)	**1.00**
	Splice	0.06 (0.00)	0.60 (0.01)	0.01	0.07 (0.01)	0.01
	Tic-tac-toe	0.15 (0.01)	0.36 (0.01)	0.01	0.30 (0.01)	0.01
	Yeast	0.44 (0.01)	0.76 (0.01)	0.01	0.47 (0.01)	**0.03**
Large	Adult	0.00 (0.00)	0.24 (0.00)	0.01	0.00 (0.00)	**1.00**
	Chess	0.46 (0.00)	0.89 (0.00)	0.01	0.77 (0.00)	0.01
	Connect-4	0.25 (0.00)	0.34 (0.00)	0.01	0.33 (0.00)	0.01
	Letter	0.16 (0.01)	0.96 (0.00)	0.01	0.38 (0.01)	0.01
	Shuttle	0.00 (0.00)	0.21 (0.00)	0.01	0.01 (0.00)	0.01

Table 3: Classification errors and empirical *p*-values obtained with Decision Tree classifier for Test 1 and Test 2. The empirical *p*-values are calculated over 1000 randomized samples for small data sets and over 100 randomized samples for medium and large data sets. Classification on the original data is repeated ten times. In the table, the average values and standard deviations of the classification errors are given. Bold *p*-values correspond to nonsignificant results when the false discovery rate is restricted below 0.05 with Benjamini and Hochberg (1995) approach.

Naive Bayes

		Original	Test 1		Test 2	
	Data Set	Err. (Std)	Err. (Std)	*p*-val.	Err. (Std)	*p*-val.
Small	Audiology	0.27 (0.00)	0.79 (0.03)	0.001	0.26 (0.01)	**0.869**
	Autos	0.43 (0.01)	0.79 (0.04)	0.001	0.22 (0.02)	**1.000**
	Breast	0.27 (0.01)	0.33 (0.02)	0.001	0.24 (0.02)	**0.959**
	Glass	0.52 (0.02)	0.81 (0.05)	0.001	0.45 (0.02)	**0.994**
	Hepatitis	0.16 (0.01)	0.30 (0.05)	0.001	0.09 (0.02)	**1.000**
	Ionosphere	0.17 (0.00)	0.46 (0.03)	0.001	0.01 (0.01)	**1.000**
	Iris	0.05 (0.01)	0.67 (0.05)	0.001	0.01 (0.01)	**0.999**
	Lymph	0.16 (0.01)	0.53 (0.05)	0.001	0.11 (0.02)	**0.995**
	Promoters	0.08 (0.01)	0.50 (0.06)	0.001	0.07 (0.02)	**0.746**
	Segment	0.21 (0.01)	0.86 (0.03)	0.001	0.13 (0.01)	**1.000**
	Sonar	0.32 (0.01)	0.50 (0.04)	0.001	0.13 (0.02)	**1.000**
	Spect	0.21 (0.01)	0.25 (0.03)	**0.077**	0.07 (0.01)	**1.000**
	Tumor	0.50 (0.01)	0.81 (0.02)	0.001	0.49 (0.02)	**0.751**
	Votes	0.10 (0.00)	0.44 (0.02)	0.001	0.00 (0.00)	**1.000**
	Zoo	0.03 (0.00)	0.81 (0.05)	0.001	0.03 (0.01)	**0.541**
Medium	Abalone	0.76 (0.00)	0.88 (0.01)	0.01	0.56 (0.01)	**1.00**
	Anneal	0.35 (0.01)	0.36 (0.04)	**0.65**	0.31 (0.01)	**1.00**
	Balance	0.09 (0.00)	0.54 (0.02)	0.01	0.24 (0.01)	0.01
	Car	0.14 (0.00)	0.30 (0.00)	0.01	0.24 (0.01)	0.01
	German	0.25 (0.00)	0.33 (0.01)	0.01	0.23 (0.01)	**1.00**
	Mushroom	0.04 (0.00)	0.50 (0.01)	0.01	0.00 (0.00)	**1.00**
	Musk	0.16 (0.00)	0.34 (0.06)	0.01	0.02 (0.00)	**1.00**
	Pima	0.24 (0.00)	0.37 (0.01)	0.01	0.22 (0.01)	**0.99**
	Satellite	0.20 (0.00)	0.80 (0.02)	0.01	0.00 (0.00)	**1.00**
	Spam	0.20 (0.00)	0.49 (0.05)	0.01	0.10 (0.00)	**1.00**
	Splice	0.05 (0.00)	0.53 (0.01)	0.01	0.03 (0.00)	**1.00**
	Tic-tac-toe	0.30 (0.00)	0.35 (0.01)	0.01	0.28 (0.01)	**1.00**
	Yeast	0.42 (0.00)	0.71 (0.01)	0.01	0.42 (0.01)	**0.36**
Large	Adult	0.02 (0.00)	0.24 (0.01)	0.01	0.01 (0.00)	**0.96**
	Chess	0.66 (0.00)	0.84 (0.00)	0.01	0.70 (0.00)	0.01
	Connect-4	0.28 (0.00)	0.34 (0.00)	0.01	0.29 (0.00)	**0.19**
	Letter	0.36 (0.00)	0.96 (0.00)	0.01	0.26 (0.00)	**1.00**
	Shuttle	0.10 (0.01)	0.47 (0.24)	0.01	0.04 (0.01)	**1.00**

Table 4: Classification errors and empirical *p*-values obtained with Naive Bayes classifier for Test 1 and Test 2. The empirical *p*-values are calculated over 1000 randomized samples for small data sets and over 100 randomized samples for medium and large data sets. Classification on the original data is repeated ten times. In the table, the average values and standard deviations of the classification errors are given. Bold *p*-values correspond to nonsignificant results when the false discovery rate is restricted below 0.05 with Benjamini and Hochberg (1995) approach.

1-**Nearest Neighbor**

		Original	Test 1		Test 2	
	Data Set	Err. (Std)	Err. (Std)	p-val.	Err. (Std)	p-val.
Small	Audiology	0.26 (0.01)	0.86 (0.03)	0.001	0.32 (0.03)	**0.030**
	Autos	0.26 (0.01)	0.77 (0.03)	0.001	0.45 (0.03)	0.001
	Breast	0.31 (0.02)	0.41 (0.03)	0.007	0.32 (0.03)	**0.324**
	Glass	0.30 (0.01)	0.74 (0.04)	0.001	0.42 (0.03)	0.001
	Hepatitis	0.19 (0.01)	0.33 (0.04)	0.002	0.14 (0.03)	**0.970**
	Ionosphere	0.13 (0.00)	0.46 (0.03)	0.001	0.26 (0.01)	0.001
	Iris	0.05 (0.00)	0.66 (0.05)	0.001	0.02 (0.01)	**0.962**
	Lymph	0.18 (0.02)	0.53 (0.04)	0.001	0.20 (0.03)	**0.307**
	Promoters	0.19 (0.02)	0.50 (0.06)	0.001	0.26 (0.04)	**0.083**
	Segment	0.14 (0.01)	0.86 (0.03)	0.001	0.15 (0.02)	**0.266**
	Sonar	0.13 (0.01)	0.50 (0.04)	0.001	0.27 (0.03)	0.001
	Spect	0.24 (0.02)	0.32 (0.04)	0.011	0.18 (0.02)	**0.970**
	Tumor	0.66 (0.02)	0.88 (0.02)	0.001	0.62 (0.02)	**0.860**
	Votes	0.08 (0.01)	0.47 (0.03)	0.001	0.01 (0.00)	**1.000**
	Zoo	0.03 (0.01)	0.75 (0.05)	0.001	0.04 (0.02)	**0.333**
Medium	Abalone	0.80 (0.00)	0.90 (0.00)	0.01	0.68 (0.01)	**1.00**
	Anneal	0.05 (0.00)	0.40 (0.02)	0.01	0.08 (0.01)	0.01
	Balance	0.20 (0.01)	0.57 (0.02)	0.01	0.35 (0.02)	0.01
	Car	0.22 (0.01)	0.41 (0.05)	0.01	0.29 (0.01)	0.01
	German	0.28 (0.01)	0.42 (0.02)	0.01	0.33 (0.02)	0.01
	Mushroom	0.00 (0.00)	0.50 (0.01)	0.01	0.01 (0.00)	0.01
	Musk	0.04 (0.00)	0.26 (0.00)	0.01	0.53 (0.01)	0.01
	Pima	0.29 (0.00)	0.45 (0.02)	0.01	0.27 (0.02)	**0.88**
	Satellite	0.10 (0.00)	0.81 (0.01)	0.01	0.01 (0.00)	**1.00**
	Spam	0.09 (0.00)	0.48 (0.01)	0.01	0.09 (0.00)	**0.31**
	Splice	0.24 (0.01)	0.61 (0.01)	0.01	0.30 (0.01)	0.01
	Tic-tac-toe	0.21 (0.02)	0.44 (0.07)	0.01	0.38 (0.02)	0.01
	Yeast	0.47 (0.01)	0.78 (0.01)	0.01	0.52 (0.01)	0.01
Large	Adult	0.02 (0.00)	0.36 (0.00)	0.01	0.01 (0.00)	**1.00**
	Chess	0.48 (0.00)	0.90 (0.00)	0.01	0.80 (0.00)	0.01
	Connect-4	0.34 (0.00)	0.50 (0.00)	0.01	0.43 (0.00)	0.01
	Letter	0.06 (0.00)	0.96 (0.00)	0.01	0.46 (0.00)	0.01
	Shuttle	0.00 (0.00)	0.36 (0.00)	0.01	0.02 (0.00)	0.01

Table 5: Classification errors and empirical p-values obtained with 1-Nearest Neighbor classifier for Test 1 and Test 2. The empirical p-values are calculated over 1000 randomized samples for small data sets and over 100 randomized samples for medium and large data sets. Classification on the original data is repeated ten times. In the table, the average values and standard deviations of the classification errors are given. Bold p-values correspond to nonsignificant results when the false discovery rate is restricted below 0.05 with Benjamini and Hochberg (1995) approach.

Support Vector Machine

		Original	Test 1		Test 2	
	Data Set	Err. (Std)	Err. (Std)	*p*-val.	Err. (Std)	*p*-val.
Small	Audiology	0.20 (0.01)	0.83 (0.03)	0.001	0.20 (0.02)	**0.443**
	Autos	0.30 (0.02)	0.73 (0.04)	0.001	0.26 (0.03)	**0.873**
	Breast	0.30 (0.01)	0.31 (0.01)	**0.191**	0.25 (0.02)	**0.970**
	Glass	0.42 (0.01)	0.65 (0.03)	0.001	0.43 (0.02)	**0.363**
	Hepatitis	0.14 (0.01)	0.21 (0.00)	0.001	0.08 (0.02)	**0.999**
	Ionosphere	0.12 (0.01)	0.37 (0.01)	0.001	0.08 (0.01)	**0.995**
	Iris	0.04 (0.01)	0.67 (0.05)	0.001	0.02 (0.01)	**0.990**
	Lymph	0.14 (0.01)	0.51 (0.05)	0.001	0.12 (0.03)	**0.686**
	Promoters	0.09 (0.01)	0.50 (0.06)	0.001	0.10 (0.03)	**0.455**
	Segment	0.12 (0.01)	0.86 (0.03)	0.001	0.12 (0.01)	**0.529**
	Sonar	0.23 (0.02)	0.49 (0.04)	0.001	0.10 (0.02)	**0.999**
	Spect	0.17 (0.01)	0.21 (0.00)	0.001	0.08 (0.02)	**1.000**
	Tumor	0.53 (0.01)	0.77 (0.01)	0.001	0.53 (0.02)	**0.406**
	Votes	0.04 (0.00)	0.39 (0.01)	0.001	0.01 (0.00)	**1.000**
	Zoo	0.04 (0.00)	0.66 (0.04)	0.001	0.04 (0.01)	**0.666**
Medium	Abalone	0.75 (0.00)	0.84 (0.00)	0.01	0.57 (0.01)	**1.00**
	Anneal	0.15 (0.00)	0.24 (0.00)	0.01	0.14 (0.01)	**0.78**
	Balance	0.12 (0.01)	0.54 (0.03)	0.01	0.25 (0.01)	0.01
	Car	0.06 (0.00)	0.30 (0.00)	0.01	0.25 (0.01)	0.01
	German	0.25 (0.00)	0.30 (0.00)	0.01	0.22 (0.01)	**1.00**
	Mushroom	0.00 (0.00)	0.50 (0.01)	0.01	0.00 (0.00)	0.01
	Musk	0.05 (0.00)	0.15 (0.00)	0.01	0.01 (0.00)	**1.00**
	Pima	0.23 (0.00)	0.35 (0.00)	0.01	0.21 (0.01)	**1.00**
	Satellite	0.13 (0.00)	0.77 (0.00)	0.01	0.00 (0.00)	**1.00**
	Spam	0.10 (0.00)	0.39 (0.00)	0.01	0.04 (0.00)	**1.00**
	Splice	0.07 (0.00)	0.48 (0.00)	0.01	0.06 (0.01)	**0.99**
	Tic-tac-toe	0.02 (0.00)	0.37 (0.01)	0.01	0.30 (0.01)	0.01
	Yeast	0.43 (0.00)	0.69 (0.01)	0.01	0.42 (0.01)	**0.72**
Large	Adult	0.00 (0.00)	0.24 (0.00)	0.01	0.00 (0.00)	**1.00**
	Chess	0.66 (0.01)	0.85 (0.00)	0.01	0.72 (0.00)	0.01
	Connect-4	0.24 (0.00)	0.45 (0.07)	0.01	0.29 (0.00)	0.01
	Letter	0.19 (0.01)	0.96 (0.00)	0.01	0.32 (0.00)	0.01
	Shuttle	0.04 (0.01)	0.21 (0.00)	0.01	0.04 (0.00)	**0.45**

Table 6: Classification errors and empirical *p*-values for the Support Vector Machine classifier for Test 1 and Test 2. The empirical *p*-values are calculated over 1000 randomized samples for small data sets and over 100 randomized samples for medium and large data sets. Classification on the original data is repeated ten times. In the table, the average values and standard deviations of the classification errors are given. Bold *p*-values correspond to nonsignificant results when the false discovery rate is restricted below 0.05 with Benjamini and Hochberg (1995) approach.

pretty simple, or it is not used by the classification algorithm. That is, the classes differ from each other, from the point of view of the classifiers, mainly due to their different value distributions of the features and not due to some dependency between the features. Thus, in many data sets the class structure is explained by considering the features independently of each other.

The results with Naive Bayes classifier are in line with the analysis in Section 4.2. That is, practically all of the results are nonsignificant with Naive Bayes with Test 2 as it explicitly assumes independence of the features. However, there are three data sets where the results are regarded as significant with Test 2: Balance, Car and Chess. These three data sets seem to contain a good balance between the features that makes the Naive Bayes classifier to perform better on the original data than on the randomized data sets. That is, each instance contains usually at least one feature which makes the classification easy whereas in the randomized data sets there are instances that do not have separating values in any of the features. Thus, applying Test 2 to Naive Bayes classifier does not tell whether the classifier uses the interdependency between the features but whether the data are such that usually at least one feature in each instance has a clear separating value.

Compared to the other three classifiers, Naive Bayes is having both better and worse performance with all kind of data sets. Surprisingly, however, Naive Bayes is performing better also in a few such cases where the other classifiers are exploiting the feature dependency. For example, with data sets Splice and Yeast the Naive Bayes classifier has the best accuracy although the Decision Tree and 1-Nearest Neighbor classifiers are significant with Test 2. Thus if a classifier is using the feature dependency in the classification, it does not directly imply that some other classifier could not do better without using the dependency. In such case, however, it is likely that neither of the classifiers are optimal and we could obtain even better performance by combining the good properties of both the classifiers.

In the rest of this section, we will consider only the three other classifiers, namely Decision Tree, 1-Nearest Neighbor and Support Vector Machine classifiers. There is a clear difference between the small and large data sets with these classifiers. The results with Test 2 for small data sets are almost all nonsignificant whereas the results for large data sets are almost all significant. Only the Adult data set from large data sets seems to contain simple class structure. Actually, the Decision Tree and Support Vector Machine classifiers are able to classify correctly all the test samples on the original Adult data set as well as on the randomized versions of the Adult data set of Test 2. The results with the studied small UCI data sets are understandable, as many of them are known to contain fairly simple structure.

The results with the three classifiers are close to each other with all tests. Surprisingly, however, 1-Nearest Neighbor classifier has been able to use the interdependency between the features the most, that is, it contains the most of small, significant p-values with Test 2. However, other more complex classifiers could be able to find more data sets where the dependency between the features is useful in classification.

Let us now study the results with Test 2 in more detail. Consider the well-known Iris data set 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 data set with Test 2, the classification error on the randomized samples is even smaller than in the original data set. 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 3.1. 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.

For most of the data sets where the empirical p-values are very high for the null hypothesis of Test 2, there are either outliers inside the classes or positive correlation between some features that is not used in the classification as it does not help in separating the classes. For example, the data set Votes contains congressional "yes" and "no" voting records from republicans and democrats. There are few voting cases where the opinion of the voter clearly reveals the political views. However, there are some outliers, that is, people who have behaved more like democrats although they are republicans, or vice versa, that vanish after randomization. Nevertheless, these reasons do not remove the fact that the features independently classify the voting records.

Finally, we discuss the results for the Balance data set. With all classifiers the classification results of the Balance data set are significant under the null hypothesis of Test 2, that is, the classifiers have exploited the dependency between the features. The structure of the data supports this: The 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 in balance. It is clear that the dependency between the features is necessary for correct classification result.

Note however, that understanding the structure inside the data sets where the classification results are regarded as significant under the null hypothesis of Test 2 requires more study, that is, we just know that the features do not explain the class structure independently. Analyzing the dependency structure of the features is then a further task. But as seen, the null hypothesis of Test 2 explains about half of the good classification results in the 33 data sets.

We conclude the experiments with a summary about the running times of the methods. We used MATLAB for producing the randomized data sets and Weka for performing the classification on a 2.2 GHz Opteron with 4 GB of main memory. The running times of producing one randomization of each data set and the running times of calculating the classification errors on the original data sets and on the randomized data sets are given in Table 7. The running times of producing the randomized data sets are negligible compared to the running times of calculating the classification errors of the data sets, that is, training and testing the classifiers. There is, however, a small difference between the running times of obtaining the classification errors on the original and the randomized data sets. Usually, the classification is a little bit faster on the original data set than on the randomized data sets. Furthermore, the classification on randomized data sets of Test 2 is usually faster than on randomized data sets of Test 1. The reason is that it is harder to teach a classifier on a randomized data set which has usually a weaker class structure than the original data set. Among the two randomization tests, Test 2 generally preserves the original class structure the most because it preserves some connection between the data and the class labels.

6. Conclusions

We have considered the problem of assessing the classifier performance with permutation tests in the statistical framework of hypothesis testing. We have described two different null hypotheses and

		Rand.		Decision Tree			Naive Bayes			1-Near. Neighbor			Supp. Vect. Mach.		
	Data Set	T1	T2	Or.	T1	T2	Or.	T1	T2	Or.	T1	T2	Or.	T1	T2
Small	Audiology	0.0	0.0	1.9	2.0	1.5	1.8	1.8	1.8	1.8	1.5	1.8	39	36	36
	Autos	0.0	0.0	0.5	0.5	0.5	0.5	0.4	0.4	0.5	0.4	0.4	3.2	2.2	2.3
	Breast	0.0	0.0	0.5	0.4	0.4	0.4	0.4	0.4	0.5	0.5	0.4	1.1	0.9	0.7
	Glass	0.0	0.0	0.4	0.4	0.3	0.3	0.3	0.2	0.3	0.2	0.2	2.4	2.3	2.3
	Hepatitis	0.0	0.0	0.4	0.3	0.3	0.3	0.3	0.2	0.3	0.3	0.3	0.5	0.4	0.4
	Ionosphere	0.0	0.0	1.2	1.0	0.9	0.8	0.7	0.7	0.9	0.8	0.9	1.2	1.3	1.0
	Iris	0.0	0.0	0.2	0.2	0.1	0.2	0.1	0.1	0.2	0.1	0.1	0.5	0.6	0.4
	Lymph	0.0	0.0	0.4	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	1.2	1.2	1.1
	Promoters	0.0	0.0	0.8	0.6	0.7	0.6	0.6	0.6	0.6	0.6	0.6	1.0	0.9	0.8
	Segment	0.0	0.0	0.5	0.6	0.4	0.4	0.3	0.3	0.4	0.3	0.4	3.6	2.4	2.3
	Sonar	0.0	0.0	1.2	0.9	1.1	0.9	0.7	0.7	0.9	0.9	0.9	1.1	1.1	0.9
	Spect	0.0	0.0	0.7	0.6	0.6	0.6	0.8	0.6	0.8	0.6	0.6	0.8	0.8	0.7
	Tumor	0.0	0.0	0.9	0.8	0.8	0.8	0.7	0.6	0.8	0.8	0.8	30	21	26
	Votes	0.0	0.0	1.0	0.8	0.7	0.9	0.9	0.7	1.0	0.9	0.8	1.2	1.1	1.1
	Zoo	0.0	0.0	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	3.1	3.1	2.6
Medium	Abalone	0.0	0.0	7.0	8.9	6.7	3.1	3.0	3.0	7.6	7.8	7.8	54	78	60
	Anneal	0.0	0.0	2.9	2.9	2.9	2.3	2.3	2.3	3.1	3.1	3.2	4.7	9.3	4.6
	Balance	0.0	0.0	0.5	0.5	0.5	0.4	0.3	0.4	0.5	0.5	0.4	0.9	1.0	0.8
	Car	0.0	0.0	1.4	1.5	1.4	1.4	1.4	1.3	1.8	1.8	1.7	5.1	8.0	7.7
	German	0.0	0.0	1.8	1.5	1.5	1.6	1.2	1.5	2.0	2.1	2.1	9.4	6.3	9.8
	Mushroom	0.0	0.0	21	24	21	20	21	20	68	70	70	60	1197	26
	Musk	0.0	0.2	130	110	176	55	63	80	230	309	318	502	5816	86
	Pima	0.0	0.0	0.8	0.7	0.9	0.7	0.7	0.5	0.8	0.7	0.8	0.9	0.8	0.9
	Satellite	0.0	0.0	26	128	21	13	14	14	59	62	81	19	157	15
	Spam	0.0	0.1	32	16	23	14	14	14	39	56	56	21	38	17
	Splice	0.0	0.0	17	17	16	15	16	15	36	28	28	87	1922	95
	Tic-tac-toe	0.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.1	1.1	1.2	3.1	3.5	6.6
	Yeast	0.0	0.0	1.7	2.2	1.6	1.3	1.0	1.3	1.4	1.4	1.7	5.2	5.3	4.9
Large	Adult	0.0	0.4	55	60	58	56	62	67	315	325	304	77	134	63
	Chess	0.0	0.1	71	57	54	45	54	59	131	105	111	93	89	102
	Connect-4	0.0	0.9	379	380	292	387	391	291	1161	1297	1285	975	558	1066
	Letter	0.0	0.1	37	45	42	32	45	34	95	103	99	43	50	45
	Shuttle	0.0	0.2	176	139	142	141	143	138	339	419	319	149	170	140

Table 7: Average running times in seconds for obtaining one randomization version of each data set for Test 1 (T1) and Test 2 (T2), as well as running times for obtaining one classification error for the four studied classifiers on each original data set (Or.) and on each randomized version of each data set (T1, T2). The running times are the average values over all the samples produced. Note that the classification procedures for small, medium and large data sets differ from each other.

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 two null hypotheses can be summarized as follows: (1) the data and the class labels are independent; and (2) the features are mutually independent given the class label.

Each test evaluates whether a certain structure (label–class dependency or dependency between features inside a class) is present in the data, and whether the classifier can use such structure for obtaining good results. If the original data really contains the structure being tested, then a significant classifier should use such information and thus obtain a low *p*-value. If the classifier is not significant then it will not notice the structure from the original data and thus, get a high *p*-value. On the other hand, if the original data does not contain any structure at all, then all *p*-values should be very high.

We have performed extensive experiments both on synthetic and real data. Experiments showed that the traditional permutation test (i.e., data and class labels are independent) is not useful in studying real data sets as it produces a small *p*-value even if there is only a weak class structure present. Compared to this, the new test proposed, that is, permuting the features inside a class, was able to evaluate the underlying reasons for the classifier performance on the real data sets. Surprisingly, however, in about half of the studied real data sets 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 the chosen classifier. That is, either the same or even better performance could be obtained by using some simpler methods, or the classification performance could be improved further by taking some useful unused feature dependency into account by changing the classification algorithm.

Interpreting the descriptive information provided by Test 2 needs care. If the classifier is significant with Test 2, then the data really contains a feature dependency that the classifier is exploiting. However, if the classifier is not significant with Test 2, that is, we obtain a high *p*-value, there are three different possibilities: (1) there are no dependencies between the features in the data; (2) there are some dependencies between the features in the data but they do not increase the class separation; or (3) there are useful dependencies between the features in the data that increase the class separation but the chosen classifier is not able to exploit them. In the third case, we would like to find such a classifier that could use the feature dependency to improve the classification performance. However, in general, when a high *p*-value is obtained with Test 2, we cannot know which of these applies to the data and to the chosen classifier. Thus the best we can do is to continue the search for a better classifier by assuming that any of them could be true. That is, we try more complex classifiers that could use the possible existing feature dependency, as well as simpler classifiers that could perform better if no feature dependency exists. Nevertheless, the answer provided by Test 2 is definite, that is, it tells whether the chosen classifier uses feature dependency to improve the classification performance.

Future work should explore the use of Test 2 for selecting the best discriminant features for classifiers, in similar fashion as Test 1 has been used for decision trees and other biological applications (Frank, 2000; Frank and Witten, 1998; Maglietta et al., 2007). Also, it would be useful to extend the setting to unsupervised learning, such as clustering. In addition, more study is needed for exploiting the descriptive information provided by Test 2. Specifically, how should we proceed to improve and study the classification performance when a high *p*-value is obtained with Test 2?

Acknowledgments

The authors would like to thank the editor and the anonymous reviewers for their careful reading and for the useful comments that improved substantially this manuscript. Thanks also to Antti Ukkonen and Kai Puolamäki for discussions on the previous version of this paper.

References

Arthur Asuncion and David J. Newman. UCI machine learning repository, 2007. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995.

Alain Berlinet, Gérard Biau, and Laurent Rouvière. Functional supervised classification with wavelets. *Annales de l'ISUP*, 52:61–80, 2008.

Julian Besag and Peter Clifford. Sequential Monte Carlo p-values. *Biometrika*, 78(2):301–304, 1991.

Howard D. Bondell and Brian J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64:115–123, 2008.

Ulisses Braga-Neto and Edward R. Dougherty. Is cross-validation valid for small-sample microarray classification? *Bioinformatics*, 20(3):374–380, 2004.

George Casella and Roger L. Berger. *Statistical Inference*. Duxbury Resource Center, 2001.

Yuguo Chen, Persi Diaconis, Susan P. Holmes, and Jun S. Liu. Sequential Monte Carlo methods for statistical analysis of tables. *Journal of the American Statistical Association*, 100(469):109–120, 2005.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

Michael P. Fay, Hyune-Ju Kim, and Mark Hachey. On using truncated sequential probability ratio test boundaries for Monte Carlo implementation of hypothesis tests. *Journal of Computational and Graphical Statistics*, 16(4):946–967, December 2007.

Eibe Frank. *Pruning Decision Trees and Lists*. PhD thesis, University of Waikato, 2000.

Eibe Frank and Ian H. Witten. Using a permutation test for attribute selection in decision trees. In *International Conference on Machine Learning*, pages 152–160, 1998.

Aristides Gionis, Heikki Mannila, Taneli Mielikäinen, and Panayiotis Tsaparas. Assessing data mining results via swap randomization. *ACM Trans. Knowl. Discov. Data*, 1(3), 2007.

Polina Golland and Bruce Fischl. Permutation tests for classification: Towards statistical significance in image-based studies. In *International Conference on Information Processing and Medical Imaging*, pages 330–341, 2003.

Polina Golland, Feng Liang, Sayan Mukherjee, and Dmitry Panchenko. Permutation tests for classification. In *Annual Conference on Learning Theory*, pages 501–515, 2005.

Phillip I. Good. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses; Springer series in statistics.*, volume 2nd. Springer, 2000.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979.

Tailen Hsing, Sanju Attoor, and Edward R. Dougherty. Relation between permutation-test p values and classifier error estimates. *Mach. Learn.*, 52(1-2):11–30, 2003.

Anders Isaksson, Mikael Wallman, Hanna Göransson, and Mats G. Gustafsson. Cross-validation and bootstrapping are unreliable in small sample classification. *Pattern Recogn. Lett.*, 29(14): 1960–1965, 2008.

David Jensen. *Induction with Randomization Testing: Decision-Oriented Analysis of Large Data Sets*. PhD thesis, Washington University, St. Louis, Missouri, USA, 1992.

Rosalia Maglietta, Annarita D'Addabbo, Ada Piepoli, Francesco Perri, Sabino Liuni, Graziano Pesole, and Nicola Ancona. Selection of relevant genes in cancer diagnosis based on their prediction accuracy. *Artif. Intell. Med.*, 40(1):29–44, 2007.

Annette M. Molinaro, Richard Simon, and Ruth M. Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

Markus Ojala and Gemma C. Garriga. Permutation tests for studying classifier performance. In *Proceedings of the 9th IEEE International Conference on Data Mining*, pages 908–913, 2009.

Markus Ojala, Niko Vuokko, Aleksi Kallio, Niina Haiminen, and Heikki Mannila. Randomization methods for assessing data analysis results on real-valued matrices. *Statistical Analysis and Data Mining*, 2(4):209–230, 2009.

Fabrice Rossi and Nathalie Villa. Support vector machine for functional data classification. *Neurocomputing*, 69(7-9):730–742, 2006.

Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition edition, 2005.