Feature Selection Filters Based on the Permutation Test

Predrag Radivojac [1], Zoran Obradovic [2], A. Keith Dunker [1], Slobodan Vucetic [2]

[1] Center for Computational Biology and Bioinformatics, Indiana University, Indianapolis, IN 46202, U. S. A.
[2] Center for Information Science and Technology, Temple University, Philadelphia, PA 19122, U. S. A.

Abstract. We investigate the problem of supervised feature selection within the filtering framework. In our approach, applicable to the two-class problems, the feature strength is inversely proportional to the p-value of the null hypothesis that its class-conditional densities, $p(X \mid Y = 0)$ and $p(X \mid Y = 1)$, are identical. To estimate the p-values, we use Fisher's permutation test combined with the four simple filtering criteria in the roles of test statistics: sample mean difference, symmetric Kullback-Leibler distance, information gain, and chi-square statistic. The experimental results of our study, performed using naive Bayes classifier and support vector machines, strongly indicate that the permutation test improves the above-mentioned filters and can be used effectively when sample size is relatively small and number of features relatively large.

1 Introduction

The increasing trend of high dimensional data collection and problem representation calls for the use of feature selection algorithms in many machine learning tasks. Real-life datasets are often characterized by a large number of irrelevant features that, if not properly isolated, may significantly hamper model accuracy and learning speed. In addition, feature selection algorithms can be handy in explaining and visualizing data.

Traditionally, methods for selecting subsets of features that provide best performance results are divided into wrappers and filters [1]. Wrappers utilize the learning machine as a fitness (evaluation) function and search for the best features in the space of all feature subsets. This formulation of the problem allows the use of standard optimization techniques, with an additional complication that the fitness function has a probabilistic nature. Despite their simplicity and often having the best performance results, wrappers highly depend on the inductive principle of the learning model and may suffer from excessive computational complexity since the problem itself is NP-hard.

In contrast to wrappers, filters are typically based on selecting the best features in one pass, although more complex approaches have also been studied [2]. Filters are model independent and are applied as a preprocessing step to model selection and learning. In domains such as text categorization or gene selection filters are still dominant [3]. Evaluating one feature at a time, filters estimate its usefulness for the prediction process according to various metrics [4-7].

Besides wrappers and filters, some authors distinguish embedded methods as a separate category of feature selection algorithms [3, 8]. Embedded methods are incorporated into the learning procedure, and hence are also dependent on the model. Many such algorithms have been proposed in learning logical expressions or in the PAC setting. In fact, almost any learner can be considered some form of an embedded feature selection algorithm, where the particular estimation of features' usefulness may result in their weighting [9], elimination [10] or construction of new features [11].

In this paper we focus on the problem of supervised feature selection. We extend the work of Frank and Witten [12] and present a permutation-based filtering framework which is effective when the number of available examples is relatively small. In particular, we adopt a statistical approach with the goals of testing the null hypothesis that the class-conditional densities of individual features are equal and of using the test p-value for their comparison. The features are ranked in the increasing order of their estimated p-values and the top-ranked features are used for classification. To estimate p-values we used Fisher's permutation test. A simple procedure for feature ranking is then proposed that introduces only a moderate computational overhead over the standard filters. Finally, we performed extensive experimentation to evaluate the proposed permutation-based methodology.

The rest of the paper is organized as follows. Section 2 describes previous work relevant to our approach. In Section 3 we present our framework in detail. Section 4 gives experimental setting and most important results of our study. Finally, concluding remarks are contained in Section 5.

2 Related Work

A significant body of research has been produced in the feature selection area. Excellent surveys, systematizations, and journal special issues on feature selection algorithms have been presented in the recent past [1, 3, 8, 13]. Searching for the best features within the wrapper framework, Kohavi and John [1] define an optimal subset of features with respect to a particular classification model. The best feature subset given the model is the one that provides the highest accuracy. Numerous wrappers have been proposed to date. These techniques include heuristic approaches such as forward selection, backward elimination [14], hill-climbing, best-first or beam search [15], randomized algorithms such as genetic algorithms [16] or simulated annealing [17], as well as their combinations [18]. In general, wrappers explore the power set of all features starting with no features, all features, or a random selection thereof.

Optimality criterion was also tackled within the filtering framework. Koller and Sahami select the best feature subset based strictly on the joint probability distributions [19]; a feature subset $\boldsymbol{Z} \subseteq \boldsymbol{X}$ is optimal if $p(Y|\boldsymbol{X}) = p(Y|\boldsymbol{Z})$. The difference between the probability distributions was measured by the relative entropy or Kullback-Leibler distance. This problem formulation naturally leads to the backward elimination search strategy. Since the true conditional distribution $p(Y|\boldsymbol{X})$ is generally unknown, Koller and Sahami proposed an efficient approximation algorithm based on Markov blankets.

In addition to the optimality criterion, many authors also discuss relevancy of features [1, 8, 20]. John et al. [20] define relevancy in the following way: feature j is strongly relevant to the target concept and distribution *D* iff there exists a pair of examples $\mathbf{x}_1$ and $\mathbf{x}_2$ in the instance space with non-zero probability over *D*, such that $\mathbf{x}_1$ and $\mathbf{x}_2$ differ only in their assignment to feature *j* and have different class labels. Additionally, feature *j* is weakly relevant to the target concept and distribution *D*, if there is a subset of features that can be removed such that feature *j* becomes strongly relevant. The fact that a feature is irrelevant according to some criterion, however, does not automatically imply that it is not useful in the model construction process. An example of such a situation is XOR task where none of the relevancy definitions can detect all truly relevant features [1]. Hence, relevance and optimality do not imply each other.

Naturally, in cases of high-dimensional datasets containing thousands of features, filters are preferred to wrappers. The domains most commonly considered within the filtering framework are text categorization [3] and gene expression [21, 22]. A significant difference between the two models is that text categorization systems typically contain both a large number of features and a large number of examples; while gene expression data usually contain a limited number of examples pushing the problem toward statistically underdetermined. Most commonly used filters are based on information-theoretic or statistical principles. For example, information gain or χ^2 goodness-of-fit tests have become baseline methods. However, both require discretized features and are difficult to "normalize" when features are multi-valued. Kononenko explored biases of several feature selection filters and showed that even normalized versions of the information gain and χ^2-test are significantly biased against the features with fewer values [4]. In addition, χ^2-test requires >30 examples in each field of the contingency table in order for the probability distribution to be well approximated by the χ^2 distribution. Thus, in many cases the application of both methods is limited. Several other approaches frequently used are Relief [23, 24], gini-index [11], relevance [25], average absolute weight of evidence [26], bi-normal separation [6] etc. Various benchmarking studies across several domains are provided in [5-7].

Some examples of embedded methods are decision tree learners such as ID3 [27] and CART [11] or the support vector machine approaches of Guyon et al. [10] and Weston et al. [28]. For example, in the recursive feature elimination approach of Guyon et al. [10], an initial model is trained using all the features. Then, features are iteratively removed in a greedy fashion until the largest margin of separation is reached. Good surveys of embedded techniques are given by Blum and Langley [8] and Guyon and Elisseeff [3].

2.1 Statistical Tests in Feature Selection

Statistical tests are straightforward choices for the feature selection algorithms, especially filters. Methods such as the F-test, measuring the ratio of standard deviations between the classes, or the t-test, in cases when the distribution of features is near Gaussian, have been used for feature selection in the statistical community. Several other approaches were also proposed as splitting criteria in decision tree training. For example, White and Liu [29] used a χ^2-goodness-of-fit test, while Martin [30] used an exact probability of a contingency table. In a study by Frank and Witten [12] it was shown that Fisher's permutation test [31] used on contingency tables can be effectively implemented as a decision-tree pre-pruning strategy. This algorithm represents a

Monte Carlo approximation of Freeman and Halton's exact test and its significance lies in the bias correction of the exact probability of the contingency table. Frank and Witten also traced that the first use of the permutation test in machine learning was probably by Gaines [32], who used Fisher's exact test in rule learning. Excellent coverage of permutation tests and other non-parametric methods is provided by Efron and Tibshirani [31].

2.2 Classification Models in Evaluating Feature Selection Algorithms

Speed of the learning process in real-life conditions often dictates that the classification model be pre-selected. Thus, in the applications for high-dimensional datasets filtering methods are often combined with the simplest or fastest learners. This is due to the fact that learning parameters usually involved in choosing complex learners may make the selection process infeasible or may result in overfitting [33]. Consequently, most frequently used models have historically been naive Bayes [34], K-nearest neighbor models or decision trees, typically C4.5 [35]. Recently, embedded methods and increases in computational power have also enabled the choice of support vector machines [36].

3 Methods

We constrain our discussion to a class of binary classification problems. Let $D = \{(\mathbf{x}_i, y_i) \mid i = 1\ldots n\}$ be the set of n labeled data points, where $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k})^T$ represents an assignment of values to a set of k features $\boldsymbol{X} = (X_1, X_2, \ldots, X_k)^T$ and $y_i \in \{0, 1\}$ is the realization of the class label Y. Let $\mathcal{X}_j$ be the domain for feature X_j. As real-life datasets may contain a large number of irrelevant features, the task of estimating the relevance of a feature in a single pass is gaining importance.

Definition 1. The relevance of feature X is measured as the difference between class-conditional distributions $p(X \mid Y = 0)$ and $p(X \mid Y = 1)$.

We consider the following distance measures between $p(X \mid Y = 0)$ and $p(X \mid Y = 1)$:

1. The difference in sample means
$$r_M(X) = E[X \mid Y = 0] - E[X \mid Y = 1]\,. \tag{1}$$
2. Symmetric variant of the Kullback-Leibler distance, called J-measure [37]
$$r_J(X) = \sum_x \left\{ p(X = x \mid Y = 0) - p(X = x \mid Y = 1) \right\} \cdot \log_2 \frac{p(X = x \mid Y = 0)}{p(X = x \mid Y = 1)}\,. \tag{2}$$
3. Information gain
$$r_{IG}(X) = \sum_x \sum_y p(X = x, Y = y) \cdot \log_2 \frac{p(X = x, Y = y)}{p(X = x) \cdot p(Y = y)}\,. \tag{3}$$
4. χ^2 statistics-based measure
$$r_{CHI}(X) = \sum_x \sum_y \frac{\left\{ p(X = x, Y = y) - p(X = x) \cdot p(Y = y) \right\}^2}{p(X = x) \cdot p(Y = y)}\,. \tag{4}$$

While r_{IG} and r_{CHI} have been often and successfully used for feature selection we consider r_M and r_J as interesting alternatives; r_M as a simple measure for selection of real-valued attributes, and r_J as an information-based measure of the difference between two probability distributions. It should be noted that r_M measure is suitable for real-valued or binary features while r_J, r_{IG}, r_{CHI} are suitable for categorical features. Therefore, if r_M is used, categorical features with $C > 2$ categories are transformed into C binary features. On the other hand, if r_J, r_{IG}, r_{CHI} are used, real-valued features are discretized and transformed into categorical attributes. Note that when $p(X|Y=0) = p(X|Y=1)$ all the measures result in zero values.

r_{CHI} measure has an interpretation particularly relevant to our study. The quantity $n{\cdot}r_{CHI}$ represents χ^2-statistic that is used in the χ^2 goodness-of-fit test of independence between a feature X and target Y, or, equivalently, of the equality between distributions $p(X|Y=0)$ and $p(X|Y=1)$. It can be shown that, for a sufficiently large sample size n, the χ^2-statistic follows the χ^2 distribution with $|\mathcal{X}| - 1$ degrees of freedom. This allows easy estimation (using a lookup table) of the *p-value*, defined as the lowest significance level at which the null hypothesis H_0: $p(X|Y=0) = p(X|Y=1)$ can be rejected. Therefore, r_{CHI} is proportional to the p-value of the hypothesis test H_0, i.e. the feature relevance is inversely proportional to its p-value. We extend this notion to the novel framework for feature selection filters.

Let us define $D_j^0 = \{x_{i,j} \,|\, y_i = 0\}$ and $D_j^1 = \{x_{i,j} \,|\, y_i = 1\}$ as samples containing values of the j-th feature of all data points with class 0 and 1, respectively. Let us also denote $\theta(D_j^0, D_j^1)$ as a test statistic obtained as a function of samples D_j^0 and D_j^1, and π_j as the p-value of the hypothesis test H_0: $p(X_j\,|Y = 0) = p(X_j\,|Y = 1)$ based on $\theta(D_j^0, D_j^1)$.

Definition 2. The relevance of feature X_j is inversely proportional to the p-value π_j of the hypothesis test H_0: $p(X_j|\,Y=0) = p(X_j|\,Y=1)$ based on the test statistic $\theta(D_j^0, D_j^1)$.

Our goal is to empirically test whether Definition 2 results in a successful feature selection procedure. It should be observed that the measures in (1-4) can be interpreted as test statistics θ since they can be estimated from D_j^0 and D_j^1. Similarly, one could think of a number of different test statistics that could be used within the flexible framework provided by Definition 2.

As already mentioned, the test statistic r_{CHI} applied to large samples allows easy calculation of the p-value. However, for the small sample size and/or for an arbitrary test statistic θ, it is difficult to derive its distribution under the null hypothesis. Therefore, to estimate the p-value we use Fisher's permutation test. This reduces the feature selection algorithm to the application of a standard statistical tool with the test statistic chosen according to the criteria from Definition 1. In Section 3.1 we describe the permutation test used for feature selection according to Definition 2.

3.1 Permutation Test

The algorithm begins by choosing and calculating a test statistic θ, e.g. difference of means, J-measure, information gain, χ^2-statistic, for a given dataset. Without loss of generality, we consider only the cases where θ is non-negative. Then, assuming the null hypothesis H_0 is true, i.e. that samples D_j^0 and D_j^1 of lengths l and m were generated according to the same probability distribution, they are concatenated into a single sample $W = (D_j^0, D_j^1)$ of size $l + m$. There are $(l + m)! / (l! \cdot m!)$ possible splits of W into two parts of sizes l and m, each of which is equally likely under H_0. The achieved significance level, or the p-value, of the statistical test is defined to be the probability that the test statistic of a random permutation is at least as large as θ. However, due to the sizes of samples D_j^0 and D_j^1, the exact p-value, calculated by evaluating all possible permutations, cannot be computed in most practical situations. In such cases, it is estimated using a fixed number of permutations (B) of the combined sample W, as shown in Figure 1. In each step b ($b = 1 \ldots B$), W is randomly shuffled and split into two parts $U^*(b)$ and $V^*(b)$ of lengths l and m. The test statistic $\theta^*(b)$ is calculated for each pair $U^*(b)$ and $V^*(b)$, and the p-value is finally estimated as the fraction of times $\theta^*(b) \geq \theta$. The actually observed permutation W is included as the iteration $B+1$.

```
Input: $D = \{(\mathbf{x}_i, y_i) \mid i = 1 \ldots n\}$; $B$ – number of permutations

for each feature $j \in \{1, 2, \ldots k\}$
  split feature $j$ into $D_j^0$ and $D_j^1$, where $D_j^0 = \{\mathbf{x}_{i,j} \mid y_i = 0\}$
    and $D_j^1 = \{\mathbf{x}_{i,j} \mid y_i = 0\}$ and $l = |D_j^0|$ and $m = |D_j^1|$
  calculate test statistic $\theta(D_j^0, D_j^1)$
  $counter = 1$
  for $b = 1$ to $B$
    randomly shuffle feature $j$ and split it into two parts
      $U^*(b)$ and $V^*(b)$, where $|U^*(b)| = l$ and $|V^*(b)| = m$
    calculate $\theta(U^*(b), V^*(b))$
    if $\theta(U^*(b), V^*(b)) \geq \theta(D_j^0, D_j^1)$
      $counter = counter + 1$
    end
  end
  $\hat{\pi}_j = counter / (B + 1)$
end

Output: $\hat{\pi}_1, \hat{\pi}_2, \ldots, \hat{\pi}_k$ – estimated p-values for features $1 \ldots k$
```

Figure 1. Steps of the permutation-test based algorithm applied to the feature selection.

After the p-values of all k features are calculated, the features are ranked according to the ascending level of their estimated p-values. Therefore, the most important features are the ones whose probability distributions of the components having different class labels are least likely to be identical.

3.2 Practical Implementation of the Permutation Test

Given the actual p-value π_j and the number of permutations B, the number of times n_j the test statistic of the random permutation is at least as large as $\theta(D_j^0, D_j^1)$ is a random variable with a binomial distribution $b(B, \pi_j)$. Therefore, to estimate π_j within 10% of its true value, the coefficient of variation $CV = ((1-\pi_j)/(\pi_j \cdot B))^{1/2}$ should be set to 0.1. As a result, for an accurate estimate of the p-value $\pi_j = 0.05$, approximately $B =$

2,000 random permutations are required. However, for highly relevant features, π_j could be extremely small which would, in turn, require a very large number of permutations.

To reduce computational cost, we limit the total number of permutations, e.g. to B = 2,000, and propose the following procedure for feature ranking. Feature X_j is classified as *weak* if $n_j \geq 0.05B$ and its p-value is estimated as $\hat{\pi}_j = (n_j + 1)/(B + 1)$. Alternatively, if $n_j < 0.05B$, feature X_j is classified as *strong*. For the strong features, n_j is close or equal to zero and their p-value estimates $\hat{\pi}_j$ are highly unreliable. For such features, we calculate Z-scores $Z_j = \{\theta(D_j^0, D_j^1) - mean(\theta(U^*, V^*))\} / std(\theta(U^*, V^*))$, where Z_j measures the distance between $\theta(D_j^0, D_j^1)$ and the distribution of the test statistic of a random permutation. Finally, strong features are ranked in the upper tier based on their estimated Z-scores, while the weak features are ranked in the lower tier based on their estimated p-values. Note that for most irrelevant features the coefficient of variation may quickly reach the pre-specified threshold in which case much less than 2,000 permutations may be required.

The computational cost of the permutation test is $O(k \cdot n)$, linear in the number of features and data points.

4 Experiments and Results

4.1 Datasets

Our experiments were performed on eleven datasets summarized in Table 1. The first nine were downloaded from the UCI repository [38], with dataset HOUSING converted into a binary classification problem according to the mean value of the target. Datasets MAMMOGRAPHY and OIL were constructed in [39] and [40], respectively, and provided to us by the authors.

Table 1. Datasets: basic characteristics. *NF* and *CF* indicate the number of numerical and categorical features, respectively.

Dataset	Size	Size of class 1	NF	CF
IONOSPHERE	351	225	34	0
VOTES	435	267	0	48
GLASS	214	163	9	0
HEART	303	139	6	7
LABOR	57	37	8	21
HOUSING	506	250	13	0
CREDIT	690	307	6	41
PIMA	768	268	9	0
ZOO	78	41	1	15
MAMMOGRAPHY	11,183	260	6	0
OIL	937	41	49	0

4.2 Data Preprocessing

The datasets were preprocessed such that each categorical feature with $C > 2$ categories was represented using C binary features. This resulted in datasets that allowed consistent comparison of different feature selection methods with naive Bayes and SVM algorithms. To allow application of r_J, r_{IG}, and r_{CHI} test statistics, prior to the feature selection process, real-valued features were discretized using the method by Fayyad and Irani [41]. This is a supervised, entropy-based algorithm that mimics recursive splitting of a decision tree construction. It automatically decides on the number of quantization intervals.

Given a real-valued feature X and a threshold t the set $S = \{(x_i, y_i) \mid i = 1 \ldots n\}$ is divided into two subsets $S_1 = \{(x_i, y_i) \mid x_i \in I_1\}$ and $S_2 = \{(x_i, y_i) \mid x_i \in I_2\}$, where $I_1 = (-\infty, t)$ and $I_2 = [t, \infty)$. The optimal threshold t is the one that maximizes the information gain defined as $G(S,t) = H(S) - E(S,t)$, where

$$H(S) = \sum_y p(y \mid x \in I) \cdot \log_2 \frac{1}{p(y \mid x \in I)}; \quad E(S,t) = \sum_{i=1}^{2} \frac{|S_i|}{|S|} \cdot H(S_i) . \tag{6}$$

The procedure is recursively repeated for each newly created interval $I \subset X$ until the stopping criterion is met. To determine whether a partition should be accepted, we use a simple stopping criterion based on the minimum description length [41]. That is, the split is allowed if

$$G(S,t) > \frac{\log_2(|S|-1)}{|S|} + \frac{\log_2(3^c - 2) - cH(S) + c_1 H(S_1) + c_2 H(S_2)}{|S|} , \tag{7}$$

where c, c_1, and c_2 are the numbers of distinct classes present in samples S, S_1, and S_2, respectively.

4.3 Evaluation Strategy and Learners

In order to evaluate different filters, we used 10 cross-validation. A dataset D is randomly divided into 10 equal-sized subsets D_1, D_2, … D_{10} and the following procedure is repeated 10 times. In each step i, $i = 1, 2, \ldots 10$, $D - D_i$ is used to rank the features and then to build a classifier, while D_i is used to test the accuracy of the classifier. The aggregate accuracy over the 10 steps represents the estimate of the accuracy. To eliminate the influence of a particular split on the performance results, we repeated the 10 cross-validation for 15 different partitions of D into 10 subsets. Finally, the average accuracy over 15 experiments is reported as the classification accuracy estimate.

Due to a considerable class-imbalance in some datasets, all models were trained and tested using balanced class sizes. During construction of the dataset D from the original data, in each of the 15 cross validation repetitions, all examples of the minority class were retained together with the same number of examples randomly selected from the majority class.

We examined two learning algorithms with different inductive biases: (i) naive Bayes and (ii) support vector machines. Naive Bayes is a MAP classifier based on the assumption that feature values are conditionally independent given the class label. In the case of categorical features, we estimated probabilities via the Laplace correction approach. For numerical features, we used the assumption of a normal distribution [7, 42]. Support vector classifiers are learning algorithms whose goal is to find class boundaries which maximize margin of class separation. For the experimental evaluation we used support vector machines with linear kernel and regularization parameter $C = 0.1$.

We examined eight feature selection methods; the first four of them are directly using measures r_M, r_J, r_{IG}, r_{CHI} for feature ranking, while the second four are ranking features according to the p-values of the hypothesis test that uses r_M, r_J, r_{IG}, r_{CHI} as test statistics, as described in Definition 2. Thus, the first four methods correspond to the traditional approach to feature ranking, while the second four correspond to the proposed framework.

For each of the eight feature selection filters, we trained naive Bayes and SVM classifiers using only the highest ranked features, then the two highest ranked features, up to the M highest ranked features, where $M = min(15, k/2)$ and k is the number of features in the dataset. This allowed us to provide a comprehensive evaluation of the proposed feature selection framework in which the datasets were similarly weighted. We observe that, to accomplish the proposed extensive evaluation, we trained $10 \cdot 15 \cdot 8 \cdot 2 \cdot M$ classifiers for each dataset.

4.4 Experimental Results

In Table 2 we show the maximum achieved accuracies for each dataset and feature selection algorithm. In each field of the table, accuracies of the two classifiers trained on $i = 1, 2, \ldots M$ highest ranked features were compared and the highest achieved accuracy is reported. These results provide a weak indication that the permutation-test based approaches result in higher accuracy. However, considering small differences between the methods and a possibility of overfitting, the results were not conclusive.

To provide a more detailed evaluation we used a win/loss counting method often applied by machine learning researchers, e.g. in [7]. The objective was to compare the performance of each traditional feature ranking measure with its permutation test counterpart. Since for each dataset we estimated accuracy of M naive Bayes and M SVM classifiers, this provided us with 103 comparisons for both classifiers. In Table 3 we list the number of wins and losses by the permutation-based method over its traditional counterpart, where the comparison is counted as a tie if the accuracies differed by less than 0.1%. It can be seen that permutation-based methods are consistently better for all four test statistics and for both classification algorithms. This provides a strong indication that the permutation-based approach is superior to traditional feature ranking.

Finally, we used a similar win/loss methodology to find the most successful among the eight feature ranking algorithms. Here, for each tuple (*dataset*, *learning method*, *number of selected features*) we compared a given feature ranking algorithm with the remaining ones and reported the number of wins (that could range from −7 to 7). The results shown in Table 4 indicate that each of the four permutation-based algorithms outperforms each of the four traditional algorithms. There is only a slight difference in the results obtained with naive Bayes and SVM classifiers. It is interesting to observe that while traditional use of r_M and r_J measures results in clearly inferior feature ranking as compared to the more popular r_{IG} and r_{CHI} measures, they appear to be superior if used as test statistics in a permutation-based approach.

Table 2. Maximum achieved accuracy [%], with the number of selected features (in parentheses), for the eight feature selection criteria. Notation: r_{IG} – information gain, r_{CHI} – chi-square statistic, r_M – sample mean difference, r_J – J-divergence. Corresponding methods based on the permutation test are denoted as $p\text{-}r_{IG}$, $p\text{-}r_{CHI}$, $p\text{-}r_M$, and $p\text{-}r_J$. Symbols $^{\blacksquare}$ and $^{\square}$ indicate that the maximum accuracy was achieved using naive Bayes classifier and SVM, respectively. The absence of any symbol indicates a tie between the two classification algorithms. Values in bold indicate the winning algorithm between each measure and its permutation-based variant.

Dataset	*Feature selection method*							
	r_{IG}	$p\text{-}r_{IG}$	r_{CHI}	$p\text{-}r_{CHI}$	r_M	$p\text{-}r_M$	r_J	$p\text{-}r_J$
IONOSPHERE	84.8 (15)$^{\blacksquare}$	**85.6** (11)$^{\blacksquare}$	83.9 (8)$^{\blacksquare}$	**84.8** (11)$^{\blacksquare}$	84.6 (13)$^{\square}$	**86.2** (4)$^{\blacksquare}$	87.2 (13)$^{\blacksquare}$	**87.7** (13)$^{\blacksquare}$
VOTES	96.0 (2)$^{\blacksquare}$	96.0 (2)$^{\blacksquare}$	96.0 (2)$^{\blacksquare}$	96.0 (2)$^{\blacksquare}$	96.0 (2)$^{\blacksquare}$	96.0 (2)$^{\blacksquare}$	96.0 (2)$^{\blacksquare}$	96.0 (2)$^{\blacksquare}$
GLASS	90.0 (2)$^{\blacksquare}$	**90.4** (4)$^{\square}$	90.1 (3)$^{\blacksquare}$	90.1 (3)	**91.0** (3)$^{\blacksquare}$	90.7 (4)$^{\square}$	89.7 (4)$^{\square}$	**90.3** (1)$^{\square}$
HEART	83.9 (9)$^{\square}$	83.9 (8)$^{\square}$	83.9 (9)$^{\square}$	83.9 (9)$^{\square}$	83.5 (11)$^{\blacksquare}$	**83.9** (9)$^{\square}$	83.8 (9)$^{\square}$	**83.9** (9)$^{\square}$
LABOR	**93.2** (8)$^{\blacksquare}$	93.0 (7)$^{\blacksquare}$	93.2 (8)$^{\square}$	93.2 (7)$^{\blacksquare}$	93.2 (13)$^{\square}$	**93.5** (9)$^{\blacksquare}$	93.0 (8)$^{\square}$	**93.2** (6)$^{\blacksquare}$
HOUSING	84.9 (4)$^{\square}$	**85.1** (4)$^{\square}$	84.9 (6)$^{\square}$	**85.0** (7)$^{\square}$	82.3 (5)$^{\square}$	**84.9** (7)$^{\square}$	85.0 (7)$^{\square}$	**85.1** (3)$^{\blacksquare}$
CREDIT	86.2 (1)	86.2 (1)	86.2 (1)	86.2 (1)	**86.6** (15)$^{\square}$	86.4 (15)$^{\square}$	86.2 (1)	86.2 (1)
PIMA	72.8 (4)$^{\square}$	**73.7** (3)$^{\blacksquare}$	72.7 (4)$^{\blacksquare}$	**73.7** (3)$^{\blacksquare}$	72.6 (4)$^{\square}$	**73.6** (3)$^{\blacksquare}$	72.8 (3)$^{\blacksquare}$	**73.6** (3)$^{\blacksquare}$
ZOO	100.0 (1)$^{\blacksquare}$	100.0 (3)$^{\blacksquare}$	100.0 (1)$^{\blacksquare}$	100.0 (3)$^{\blacksquare}$	100.0 (3)$^{\square}$	100.0 (1)$^{\blacksquare}$	**100.0** (1)$^{\blacksquare}$	99.8 (1)$^{\blacksquare}$
MAMMOGRAPHY	85.8 (3)$^{\blacksquare}$	**85.9** (3)$^{\blacksquare}$	85.8 (3)$^{\blacksquare}$	**85.9** (3)$^{\blacksquare}$	84.2 (3)$^{\square}$	**86.0** (3)$^{\blacksquare}$	85.9 (3)$^{\blacksquare}$	85.9 (3)$^{\blacksquare}$
OIL	81.5 (2)$^{\blacksquare}$	**81.7** (2)$^{\blacksquare}$	**81.8** (2)$^{\blacksquare}$	81.6 (2)$^{\blacksquare}$	80.4 (11)$^{\square}$	**81.3** (2)$^{\blacksquare}$	**80.8** (14)$^{\square}$	80.5 (2)$^{\square}$

Table 3. Relative comparisons between four feature selection filters and their permutation-test versions. The number of wins and losses are calculated using pairwise comparisons between each pair of filters. The score is calculated as the number of wins minus the number of losses.

Filters	*Naive Bayes Classifier*			*Support Vector Machine*		
	Score	*Wins*	*Losses*	*Score*	*Wins*	*Losses*
$p\text{-}r_{IG}$ vs. r_{IG}	+33	50	17	+28	44	16
$p\text{-}r_{CHI}$ vs. r_{CHI}	+20	44	24	+30	46	16
$p\text{-}r_M$ *vs.* r_M	+79	88	9	+52	69	17
$p\text{-}r_J$ vs. r_J	+42	54	12	+31	51	20

Table 4. Relative comparisons between eight feature selection filters. Each filter below is ranked according to its score, i.e. the number of wins – losses in pairwise comparisons to all other methods over all datasets and numbers of retained features.

Filters	*Naive Bayes Classifier*			Filters	*Support Vector Machine*		
	Score	*Wins*	*Losses*		*Score*	*Wins*	*Losses*
$p\text{-}r_J$	396	744	348	$p\text{-}r_M$	476	798	322
$p\text{-}r_{IG}$	354	700	346	$p\text{-}r_J$	180	618	438
$p\text{-}r_M$	268	744	476	$p\text{-}r_{CHI}$	156	566	410
$p\text{-}r_{CHI}$	180	624	444	$p\text{-}r_{IG}$	146	556	410
r_{CHI}	–8	524	532	r_{IG}	–156	408	564
r_{IG}	–30	518	548	r_{CHI}	–218	380	598
r_J	–178	482	660	r_J	–292	374	666
r_M	–982	196	1178	r_M	–292	464	756

5 Conclusions

In this study we investigated feature selection filters and the effects of the permutation test on various filtering metrics. Our experimental results suggest that sample mean difference and symmetric Kullback-Leibler distance (J-measure) can be used effectively for the feature selection and that, overall, they achieve better performance than information gain or χ^2-test. The permutation test was shown to improve all four metrics, but the improvement comes at a cost of increased computational complexity. The success of the proposed method on the diverse datasets selected for this study strongly suggests the applicability of the permutation-based framework to a wide range of real-life problems.

Acknowledgement

This study was supported by NIH grant 1R01 LM07688 awarded to AKD and ZO and by NSF grant IIS-0219736 awarded to ZO and SV. We thank Nitesh V. Chawla and Robert C. Holte for the datasets and Natascha A. Karlova for proofreading the manuscript.

References

1. Kohavi, R. and G. John, Wrappers for feature selection. *Artificial intelligence*, 1997. **97**(1-2): p. 273-324.
2. Almuallim, H. and T.G. Dietterich. Learning with many irrelevant features. *National Conference on Artificial Intelligence*. 1992: p. 547-552.
3. Guyon, I. and A. Elisseeff, An introduction to variable and feature selection. *Journal of Machine Learning Research*, 2003. **3**: p. 1157-1182.
4. Kononenko, I. On biases in estimating multi-valued attributes. *International Joint Conference on Artificial Intelligence*. 1995. p. 1034-1040.
5. Yang, Y. and J.P. Pedersen. A comparative study on feature selection in text categorization. *International Conference on Machine Learning*. 1997, p. 412-420.
6. Forman, G., An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 2003. **3**: p. 1289-1305.
7. Hall, M.A. and G. Holmes, Benchmarking attribute selection techniques for discrete class data mining. *IEEE Trans. Knowledge and Data Engineering*, 2003. **15**(6): p. 1437-1447.
8. Blum, A.L. and P. Langley, Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 1997. **97**(1-2): p. 245-271.
9. Littlestone, N., Learning quickly when irrelevant attributes abound: a new linear threshold algorithm. *Machine Learning*, 1988. **2**: p. 285-318.
10. Guyon, I., et al., Gene selection for cancer classification using support vector machines. *Machine Learning*, 2002. **46**(1-3): p. 389-422.
11. Breiman, L., Classification and regression trees. 1984, Belmont, CA.
12. Frank, E. and I.H. Witten. Using a permutation test for attribute selection in decision trees. *International Conference on Machine Learning*. 1998. p. 152-160.
13. Dash, M. and H. Liu, Feature selection for classification. *Intelligent Data Analysis*, 1997. **1**(3): p. 131-156.
14. Bishop, C.M., Neural networks for pattern recognition. 1995, Oxford University Press.

15. Caruana, R. and D. Freitag. Greedy attribute selection. *International Conference on Machine Learning*. 1994, p. 28-36.
16. Vafaie, H. and I.F. Imam. Feature selection methods: genetic algorithms vs. greedy like search. *International Conference on Fuzzy and Intelligent Control Systems*. 1994.
17. Doak, J., An evaluation of feature selection methods and their application to computer security. 1992, Techical Report CSE-92-18. University of California at Davis.
18. Brassard, G. and P. Bratley, Fundamentals of algorithms. 1996, Prentice Hall.
19. Koller, D. and M. Sahami. Toward optimal feature selection. *International Conference on Machine Learning*. 1996. p. 284-292.
20. John, G.H., R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. *International Conference on Machine Learning*. 1994. p. 121-129.
21. Li, J. and L. Wong, Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns. *Bioinformatics*, 2002. **18**(5): p. 725-734.
22. Bhattacharyya, C., et al., Simultaneous relevant feature identification and classification in high-dimensional spaces: application to molecular profiling data. *Signal Processing*, 2003. **83**: p. 729-743.
23. Kira, K. and L.A. Rendell. The feature selection problem: traditional methods and a new algorithm. *National Conference on Artificial Intelligence*. 1992, p. 129-134.
24. Kononenko, I. Estimating attributes: analysis and extension of RELIEF. *European Conference on Machine Learning*. 1994, p. 171-182.
25. Baim, P.W., A method for attribute selection in inductive learning systems. *IEEE Transactions on PAMI*, 1988. **10**: p. 888-896.
26. Michie, D., Personal models of rationality. *Journal of Statistical Planning and Inference*, 1990. **21**, p. 381-399.
27. Quinlan, J.R., Learning efficient classification procedures and their application to chess and games, Machine learning:an artificial intelligence approach 1983, Morgan Kaufmann.
28. Weston, J., et al. Feature selection for SVMs. *Neural Information Processing Systems*. 2000, p. 668-674.
29. White, A.P. and W.Z. Liu, Bias in information-based measures in decision tree induction. *Machine Learning*, 1994. **15**(3): p. 321-329.
30. Martin, J.K., An exact probability metric for decision tree splitting and stopping. *Machine Learning*, 1997. **28**(2-3): p. 257-291.
31. Efron, B. and R.J. Tibshirani, An introduction to the bootstrap. 1993, Chapman & Hall.
32. Gaines, B. An ounce of knowledge is worth a ton of data. *International Workshop on Machine Learning*. 1989, p. 156-159.
33. Dietterich, T.G., Overfitting and undercomputing in machine learning. *Computing Surveys*, 1995. **27**: p. 326-327.
34. Mitchell, T.M., Machine learning. 1997, McGraw-Hill.
35. Quinlan, J., C4.5: programs for machine learning. 1992, Morgan Kaufmann.
36. Burges, C.J.C., A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 1998. **2**(2): p. 121-167.
37. Lin, J., Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 1991. **37**(1): p. 145-151.
38. Blake, C.L. and C.J. Merz, UCI Repository of machine learning databases. 1998, UC Irvine.
39. Chawla, N.V., et al., SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 2002. **16**(2002): p. 321-357.
40. Kubat, M., R.C. Holte, and S. Matwin, Detection of oil spills in satellite radar images of sea surface. *Machine Learning*, 1998. 30: p. 195-215.
41. Fayyad, U.M. and K.B. Irani. Multiinterval discretisation of continuous-valued attributes. *International Joint Conference on Artificial Intelligence*. 1993. p. 1022-1029.
42. Langley, P., W. Iba, and K. Thompson. An analysis of Bayesian classifiers. *National Conference on Artificial Intelligence*. 1992. p.223-228.