Applying permutation tests for assessing the statistical significance of wrapper based feature selection

Antti Airola, Tapio Pahikkala, Jorma Boberg, Tapio Salakoski
Department of Information Technology
University of Turku, Turku Centre for Computer Science
Turku, Finland
Email: firstname.lastname@utu.fi

***Abstract*—Feature selection is commonly used in bioinformatics applications, such as gene selection from DNA microarray data. Recently, wrapper methods have been proposed as an improvement over traditionally used filter based feature selection methods. In wrapper methods, the goodness of a feature set is often measured using the cross-validation performance of a machine learning method trained with the features. This can lead to overfitting, meaning that the cross-validation performance on the final selected feature set may be high even in cases when the selected features in fact are not informative. Evaluating the statistical significance of gained results is therefore of major concern.**

Non-parametric permutation tests have been previously used as a univariate filter for selecting individual features. In contrast, we propose using such tests to measure the statistical significance of the whole selection process, which is carried out by a wrapper method. We achieve computational efficiency by using a regularized least-squares based wrapper method, which combines a state-of-the-art classifier with matrix calculus based computational shortcuts for greedy forward feature selection. Permutation tests prove to be a practical tool for estimating the significance of gained results, as shown in simulations and experiments on two DNA microarray data sets.

I. Introduction

Feature selection has in the recent years gained prominence in bioinformatics. A wide range of applications have been proposed in areas such as sequence analysis, microarray analysis, mass spectrometry and biomedical text mining ([1]). Feature selection has the potential to deepen the understanding of the studied biological phenomenon by identifying relevant features, and to lead to more accurate, or computationally efficient models by eliminating irrelevant features. In this work, we consider as a case study the task of gene selection from DNA microarray data (see e.g. [2]). However, the introduced approach is general, being applicable to many types of supervised learning tasks, where feature selection needs to be performed.

We assume the standard binary classification setting, where a training set containing the feature representations and class labels of a number of examples is supplied. The class labels determine whether a training example belongs to the positive or to the negative class. For example, in cancer classification the examples correspond to patients, the features to gene expression level measurements and the class labels to the diagnosis, whether the patient has cancer or not. The aim is to learn a model which correctly predicts the class label of any new example given its feature values.

Further, we consider the wrapper model for feature selection ([3]), where features are selected through interaction with a classifier training method. In the wrapper approach, the power set of features is searched over. A new classifier is trained during each search step using the corresponding feature subset and an estimate of its classification error is used to measure the quality of the subset. As suggested by [3], we use leave-one-out (LOO) cross-validation (CV) (see e.g. [4]) for estimating classification error.

So far, the most popular feature selection approaches in bioinformatics have been univariate filter methods. Here each feature is assessed separately with respect to its ability to discriminate between the classes. The downside compared to the the wrapper approach is, that when each feature is considered in isolation, possible interactions between them are lost. It has been recently experimentally shown that on the DNA microarray domain the wrapper approach tends to provide better performance than filter methods [5]. This improvement is however coupled with considerable increase in computational cost, when using the classifier inside the wrapper as a black box method.

In many biological applications a typical property of the data is small sample size (tens of examples), and high dimensionality (thousands of dimensions). Further, the level of signal in the data may be low, or even non-existent. In such settings it is likely that strong patterns between the features and the labels will arise in the training set simply by random chance. It is a well documented result in the literature, further supported by experimental evidence presented in this article, that this can lead to serious overfitting in feature selection [6], [7], [8], [9]. A model constructed from apparently informative features may have low CV error, and yet fail to generalize beyond the training set. Nested CV, where on each round of CV the selection process is re-run, has been proposed for error estimation in this setting (see e.g. [9]). While this estimator is almost unbiased, the high variance of CV still remains a problem on small data sets

(see e.g. [10], [11]). On small data sets the CV error alone may not provide enough information to decide whether the features found by the selection method truly are informative.

Non-parametric permutation tests have been previously proposed for measuring the statistical significance of CV results for classification [12], [13]. In this work we explore the suitability of the approach for assessing the significance of feature selection. The null hypothesis is that the classification method cannot reliably learn to predict the labels from the selected set of features. The alternative hypothesis is that a classifier with a low error rate can be trained from the selected features. The test is done by repeatedly permuting the labels of the training set, performing the feature selection process, and evaluating the CV performance of the model trained on the final set of selected features. The p-value is simply the relative frequency of such runs that result in as good as or better CV performance than the CV performance of the model trained on the non-permuted labels.

In feature selection, permutation tests have been previously used as filter methods (see e.g. [14], [15]). In this approach, the permutation test is used together with a univariate statistic to calculate a p-value for each feature separately. The p-value is for the null hypothesis that the class conditional densities for a feature are equal for both classes. This approach provides no estimate for the statistical significance of the whole selection process itself, but simply acts as a criterion for selecting individual features. In contrast, our approach of combining the permutation test with wrapper based feature selection allows us to capture interactions between different features, and provides a tool for estimating the statistical significance of the whole selection process.

One of the main challenges in applying permutation tests for state-of-the-art wrapper based feature selection methods is the computational cost. These methods are typically computationally intensive, requiring the re-training of a machine learning method for each tested feature set, and each round of CV. Repeating this process for, say, 1000 permutations of the labels, is typically not feasible. However, using a wrapper method based on the regularized least-squares (RLS) classifier ([16]), this process can be efficiently carried out. RLS has been a popular algorithm for gene selection from microarray data due to both the good classification performance, and the availability of computational shortcuts for CV (see e.g. [17], [15]). Recently, we have proposed the greedy RLS wrapper method, which combines the RLS CV shortcuts with efficient updating when adding new features, leading to linear time and space complexities in training ([18]). Using greedy RLS, we achieve the computational efficiency necessary for combining permutation tests with wrapper based feature selection.

II. Methods

A. Regularized Least-Squares

Let $\mathbb{R}^m$ and $\mathbb{R}^{n\times m}$, where $n, m \in \mathbb{N}$, denote the sets of real valued column vectors and $n \times m$-matrices, respectively. To denote real valued matrices and vectors, we use bold capital letters and bold lower case letters, respectively. Moreover, index sets are denoted with calligraphic capital letters. For index set $\mathcal{R} \subseteq \{1, \ldots, n\}$, we denote the submatrix of $\mathbf{M} \in \mathbb{R}^{n\times m}$ having the rows of $\mathbf{M}$ indexed by $\mathcal{R}$ as $\mathbf{M}_\mathcal{R}$. We use an analogous notation also for column vectors, that is, $\mathbf{v}_\mathcal{R}$ refers to a vector consisting of the entries of the vector $\mathbf{v} \in \mathbb{R}^n$ indexed by $\mathcal{R}$.

Let $\mathbf{X} \in \mathbb{R}^{n\times m}$ be the data matrix containing the feature representations of the training examples, where n is the total number of features and m is the number of training examples. The i, jth entry of $\mathbf{X}$ contains the value of the ith feature in the jth training example. Moreover, let $\mathbf{y} \in \{-1, 1\}^m$ be a vector containing the labels of the training examples.

In this paper, we consider linear predictors of type

$$f(\mathbf{x}) = \mathbf{w}^\mathrm{T}\mathbf{x}_\mathcal{S},$$

where $\mathcal{S} \subseteq \{1, \ldots, n\}$, $\mathbf{w}$ is the $|\mathcal{S}|$-dimensional vector representation of the learned predictor and $\mathbf{x}_\mathcal{S}$ can be considered as a mapping of the data point $\mathbf{x}$ into $|\mathcal{S}|$-dimensional feature space. Note that the vector $\mathbf{w}$ only contains entries corresponding to the features indexed by $\mathcal{S}$. The rest of the features are not used in the prediction phase. The class label predicted for a new data point $\mathbf{x}$ is determined by the sign of the real valued prediction.

Given training data and a set of feature indices $\mathcal{S}$, we find $\mathbf{w}$ by minimizing the RLS risk. This can be expressed as:

$$\operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^{|\mathcal{S}|}} \left\{((\mathbf{w}^\mathrm{T}\mathbf{X}_\mathcal{S})^\mathrm{T} - \mathbf{y})^\mathrm{T}((\mathbf{w}^\mathrm{T}\mathbf{X}_\mathcal{S})^\mathrm{T} - \mathbf{y}) + \lambda\mathbf{w}^\mathrm{T}\mathbf{w}\right\}.$$

The first term, called the empirical risk, measures how well the prediction function fits to the training data. The second term called the regularizer controls the tradeoff between the loss on the training set and the complexity of the prediction function. The computational complexity of learning a linear RLS predictor with $|\mathcal{S}|$ features and m training examples is $O(\min\{|\mathcal{S}|^2 m, |\mathcal{S}|m^2\})$ (see e.g. [16]).

B. Greedy Forward Feature Selection for RLS

Here, we consider greedy forward feature selection for RLS with LOO criterion (for a description of LOO error, see [4]). A high level pseudo code of greedy RLS is presented in Algorithm 1. The outermost loop adds one feature at a time to the set of selected features $\mathcal{S}$ until the size of the set has reached the desired number of selected features k. The inner loop goes through every feature that has not yet been added to the set of selected features. For each feature available for addition, the LOO error of the RLS predictor trained on both the previously chosen features, and the new

feature, is evaluated. The feature whose addition provides the lowest LOO error is then chosen. The function

$$l : \bigcup_{i=1}^{n} (\mathbb{R}^{i \times m} \times \{-1, +1\}^m \times \mathbb{R}) \mapsto \mathbb{R} \quad (1)$$

maps a data matrix $\mathbf{X}_{\mathcal{R}}$, a label vector $\mathbf{y}$, and a regularization parameter λ to the LOO classification error $l(\mathbf{X}_{\mathcal{R}}, \mathbf{y}, \lambda)$ of RLS trained with the features indexed by $\mathcal{R}$.

Algorithm 1: Pseudo code of greedy RLS

```
Input: X ∈ R^{n×m}, y ∈ R^m, k, λ
Output: S
S ← ∅;
while |S| < k do
    e ← ∞;
    b ← 0;
    foreach i ∈ {1, ..., n} \ S do
        R ← S ∪ {i};
        e_i ← l(X_R, y, λ);
        if e_i < e then
            e ← e_i;
            b ← i;
        end
    end
    S ← S ∪ {b};
end
```

In the standard wrapper approach for feature selection (see e.g. [3]), RLS is used as a black-box method meaning that a new RLS predictor is trained for each tested feature set and for each CV round. However, given that the computation of LOO classification error requires m retrainings, that the forward selection goes through $O(n)$ features in each iteration, and that k features are chosen, the process would become computationally costly. Namely, the overall computational complexity of the greedy forward selection with LOO criterion is $O(\min\{k^3m^2n, k^2m^3n\})$. Even worse, the whole process would have to be repeated for each considered permutation when assessing statistical significance with permutation tests.

In [18], we have proposed an algorithm for learning sparse predictors, called greedy RLS, whose computational complexity is $O(kmn)$, that is, it scales linearly with the desired number of features to be selected, the size of the training set, and with the overall number of features in the training set. Moreover, we have shown that the predictor trained with greedy RLS is exactly the same as the one obtained via the standard wrapper approach with LOO criterion. For a detailed technical description of the method, we refer the reader to [18].

C. Permutation Tests

Here, we consider permutation tests similar to those used for classification (see e.g. [12]). In permutation tests, the labels of the training data are shuffled randomly and a new predictor is then constructed and evaluated with the data with permuted labels. The shuffling, training, and performance evaluation is repeated many times and the evaluation results are used for computing the statistical significance of the result obtained with the original training data.

Algorithm 2: Permutation test

```
Input: X ∈ R^{n×m}, y ∈ R^m, k, λ, r
Output: p
S ← GreedyRLS(X, y, k, λ);
γ ← l(X_S, y, λ);
c ← 0;
repeat r times
    y' ← π(y);
    S' ← GreedyRLS(X, y', k, λ);
    γ' ← l(X_S', y', λ);
    if γ' ≤ γ then c ← c + 1
end
p ← c/r;
```

Following [12], the performance of a trained predictor is evaluated via a statistic $\mathcal{T}$ measuring the similarity between the sets of positive $\{(x_i, y_i) \mid y_i = 1\}$ and negative $\{(x_i, y_i) \mid y_i = -1\}$ training examples. The null hypothesis assumes the independence of the data and the labels, that is, the data would contain no signal related to the class labels.

The pseudo code of the algorithm calculating the permutation test is given in Figure 2. The algorithm is provided with the original training data, the desired number of features to be selected k, the regularization parameter λ, and the number of permutation rounds r. First, the algorithm evaluates the predictor trained with greedy RLS and with the original labels. In the algorithm description, "GreedyRLS" denotes a function which performs the whole feature selection process via the greedy RLS algorithm. The permutation test algorithm then repeats r times the computation of the evaluation statistic for a predictor trained with permuted labels. The function providing shuffled label vectors is denoted by π. The algorithm returns the value p, which is the relative frequency of such evaluation results achieved with the permuted labels, which are as good as or better as the results gained with the true labels. Let α be the acceptable significance level. Then the null hypothesis can be rejected if $p < \alpha$.

As a statistic $\mathcal{T}$, we use the LOO classification error (1) of the predictor obtained with the greedy RLS algorithm. Note that since we use the LOO error also as a criterion in the feature selection process, it is likely that its value will be very low even with data sets containing no signal. However, the whole feature selection process is rerun for each permutation of the training labels, and hence it is as likely to get low LOO errors with the shuffled labels as with the original labels containing no signal. That is, the permutation test is able to capture the expressive power of greedy RLS and the overfitting of the LOO error measure during the whole feature selection process.

The overall computational complexity of running the test with greedy RLS is $O(rkmn)$, where r is the number

of permutations performed by the test. In order for the permutation test to be exact, we should go through every possible permutation of the class labels. The number of permutations grows exponentially with respect to the number of training examples, and hence computing the exact test is infeasible. However, reliable estimates of the p-value can be achieved with Monte Carlo sampling of permutations.

D. Number of Features to be Selected

A question still left open is how many features one should select. There are many possible approaches, each having their own advantages and disadvantages. The answer, of course, also depends on the background and characteristics of the feature selection task. We next consider three possible approaches.

First, we can decide the number of features to be selected in advance. One may use prior knowledge such as, for example, if it has been previously conjectured that the underlying concept depends on a certain number of features. Alternatively, the number of selected features may be constrained by the memory available for storing the predictor, or by real-time demands for the prediction speed. In this case, the technical limitations may dictate an upper bound on the number of features to be selected. A advantage of this approach is that the actual feature selection process has one free parameter less. An obvious disadvantage is that setting the number of selected features too high can lead to selecting a large number of non-relevant features, and setting it too low leads to missing important ones.

The second approach we test in our experiments is to stop the feature selection process when the LOO error stops decreasing. This method has also certain disadvantages. Firstly, it can get stuck to a local minima. This is the case, for example, in our the experiments with the breast cancer data. Moreover, it can not be guaranteed that the method will stop before the statistical significance is lost due to overfitting. In fact, according to the p-value curves of our experiments, the results are statistically significant only for feature sets of very small size. Nevertheless, in our experiments with real data, results provided by this approach are significant.

A third possibility is to use the permutation test itself for determining when to stop the selection. Namely, more features would be added to the set of selected features as long as the p-value returned by the permutation test is below the significance threshold. A downside of this approach is that it makes permutation test less reliable for computing the overall significance of the final result of the feature selection compared to the case in which the test is computed only once. This problem can in turn be solved by, for example, by adjusting the significance threshold via some sort of a correction for multiple-hypothesis testing, such as the Bonferroni correction.

III. Results

We perform experiments on an artificial non-signal data set, as well as two real world microarray data sets. First, we explore the degree to which performing feature selection on the whole data set biases the LOO error estimate. Second, we show how permutation tests can be used to detect whether the results of a feature selection method are significant.

The non-signal data consists of 50 examples with 2000 features each. The feature values are generated from a normal distribution with zero mean and unit variance. We assume a balanced class distribution, so that half of the examples are randomly assigned positive, and half of the examples negative class labels. Any classifier trained on this data will have expected error rate of 0.5 on new examples, since class labels are assigned with equal probability, and independently of the feature values.

The two real world data sets are a colon cancer data set ([19]), with 62 examples and 2000 features, and a breast cancer data set ([20]) with 44 examples and 7129 features. The classification problem is that of performing cancer diagnosis on basis of gene expression level profiles measured on DNA microarray. Previous work on the data, further supported by our results, indicates a high probability of real signal being present in the data sets.

In the initial experiments the test protocol is as follows. First, a number of features to be selected is fixed. Then, for each fixed number of features greedy RLS is run starting from empty feature set until the given number has been selected with the true labels, as well as with 1000 permutations of the labels. For the non-signal data we consider only the permutation distribution, since there is no correct true labeling. We repeat the procedure for all the feature set sizes ranging from 1 to 100, and calculate at each point the LOO error with the true labels, the mean and variance of LOO error for the permutation distribution, and the p-values. We set $\lambda = 1$ on all the experiments.

In Figure 1 are the mean LOO-errors for the permutation distributions, as well as the LOO-errors with the true labels when available. When studying the results for the non-signal data, the strength of the overfitting phenomenon becomes clear. The true error rate of each model is 0.5. However, performing feature selection on the full data set allows the learners to overfit to the LOO-criterion, so that the mean of the LOO-error can be as low as 0.04. Thus it is clear that studying the LOO-performance of a classifier alone is not sufficient to determine whether the selected feature set is informative, when the whole data set is used in the selection process. The same overfitting occurs on the real world data sets. However, the error of the classifier trained on the true labels is still clearly lower than the mean of the permutation distribution.

In Figure 2 are the variances of the LOO-errors for the permutation distributions. It can be seen that the variance

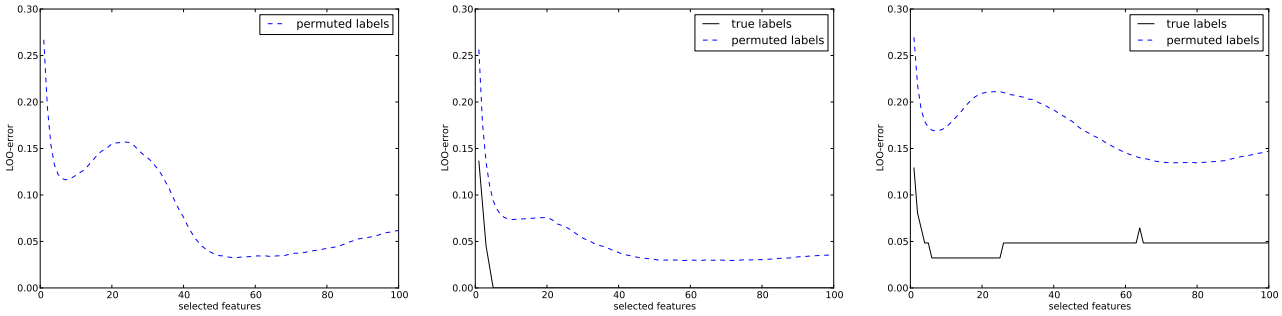

Figure 1. Mean LOO-errors over 1000 permutations, and the LOO errors for true labels. Non-signal (left), colon cancer (middle), breast cancer (right)

grows quite fast as the feature selection proceeds, though it seems to level off once enough features have been selected. One conclusion that can be drawn from studying the variances is that for small high-dimensional data sets one should aim to select only a handful of features. For too large set of selected features the variance will become so high that it will become impossible to detect whether the selection procedure was successful. The p-values presented in Figure 2 support this conclusion, the uncertainty grows quite high very soon as the feature selection process continues.

Based on these considerations, we perform a further set of experiments. We study whether the greedy RLS can successfully find relevant features on the two real world data sets, using permutation tests. In this test we use an adaptive stopping criterion for greedy RLS. In the first experiment, we terminate selection once selecting a new feature no longer lowers the LOO-error. Naturally the same termination criterion is used both when running greedy RLS with the true labels, and with the permuted labels. Here, we test for statistical significance at $p < 0.05$. In the second experiment, we test the use of the permutation test itself as a stopping criterion. Namely, we stop the feature selection process at the point when adding a new feature would rise the p-value over the significance threshold. Then, we select the feature set having the lowest LOO error from among the sets seen during the previous steps of the greedy forward selection. If there are several such sets, we select the one with least number of features. Note that when we use the p-value as a stopping criterion, its reliability for computing the statistical significance of the final result would be questionable.

The resulting feature set sizes, LOO-errors, and for the first experiment also the corresponding p-values, are presented in Table I. Further, histograms for the permutation distributions in the first experiment are presented in Figure 3. First, we consider the results for the first experiment. The selected number of features is quite small. For colon cancer data 5 features are selected, for breast cancer 4. The LOO-errors are also very small, with 0 error for colon cancer, and 0.048 error for breast cancer. Both results are significant. Interestingly enough, while with the cancer data the resulting error is 0, the p-value 0.042 is much higher than that of the

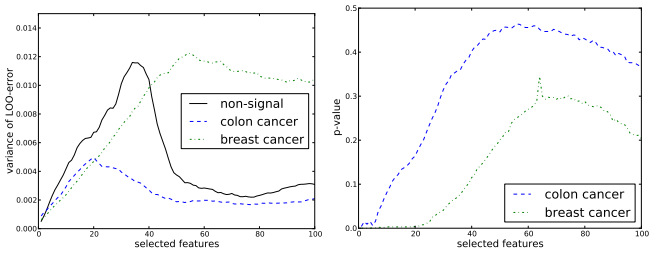

Figure 2. Variances of the permutation distributions (left). p-values for the experiment (right).

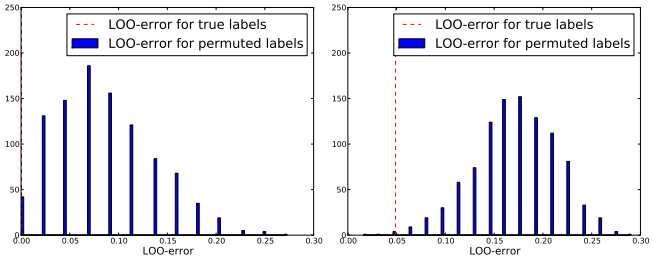

Figure 3. Histograms of the LOO-errors for the permutation test, calculated at the point where feature selection terminates. Error for true labels presented as a dashed red line. Colon cancer (left), breast cancer (right). For colon cancer, error with true labels is 0.00.

breast cancer data (0.006). The reason for this can be seen from the histograms. For the colon cancer data it can be seen that 0 error is achieved also for a number of permutations, indicating the possibility that the good result might have arisen simply by luck. However, the breast cancer result, as seen, is more of an outlier, with only very few of the permutations resulting in as good as, or better performance. In the second experiment, the feature selection process with the colon cancer data stops after selecting 5 features, since the LOO error becomes zero and the p-value is below the threshold. With the breast cancer data, the process stops when it has reached the LOO error 0.032 after selecting 6 features.

IV. Conclusion

The experiments further verify the need for significance testing in wrapper based feature selection, as such methods

Table I
EXPERIMENT 1: THE FEATURE SELECTION PROCESS IS STOPPED WHEN THE LOO ERROR STOPS DECREASING. EXPERIMENT 2: THE SMALLEST FEATURE SET HAVING THE LOWEST STATISTICALLY SIGNIFICANT LOO ERROR IS SELECTED.

	Exp1			Exp2	
data set	selected	LOO	p-value	selected	LOO
colon cancer	5	0.000	0.042	5	0.000
breast cancer	4	0.048	0.006	6	0.032

possess a notable risk of overfitting on small high dimensional samples. The permutation test captures the expressive power of the wrapper methods, allowing one to detect whether the results are significant or not. However, due to the computational costs of performing the test, the used wrapper method must be efficient. One suitable choice is the greedy RLS algorithm. Though we have limited our considerations to binary classification, the presented approach is applicable also for other types of learning tasks where feature selection is needed, such as learning to rank [21].

ACKNOWLEDGEMENTS

This work has been supported by the Academy of Finland.

REFERENCES

[1] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[2] E. Xing, "Feature selection in microarray analysis," in *A Practical Approach to Microarray Data Analysis*, D. Berrar, W. Dubitzky, and M. Granzow, Eds. Boston, MA: Kluwer Academic Publishers, 2003, ch. 6, pp. 110–131.

[3] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif Intell*, vol. 97, no. 1-2, pp. 273–324, 1997.

[4] P. A. Lachenbruch, "An almost unbiased method of obtaining confidence intervals for the probability of misclassification in discriminant analysis," *Biometrics*, vol. 23, no. 4, pp. 639–645, 1967.

[5] I. Inza, P. Larrañaga, R. Blanco, and A. J. Cerrolaza, "Filter versus wrapper gene selection approaches in DNA microarray domains," *Artif Intell Med.*, vol. 31, no. 2, pp. 91–103, 2004.

[6] C. Ambroise and G. J. McLachlan, "Selection bias in gene extraction on the basis of microarray gene-expression data," *P Natl Acad Sci USA*, vol. 99, no. 10, pp. 6562–6566, 2002.

[7] R. Simon, M. D. Radmacher, K. Dobbin, and L. M. McShane, "Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification," *J Natl Cancer Inst*, vol. 95, no. 1, pp. 14–18, 2003.

[8] J. Reunanen, "Overfitting in making comparisons between variable selection methods," *JMLR*, vol. 3, pp. 1371–1382, 2003.

[9] S. Varma and R. Simon, "Bias in error estimation when using cross-validation for model selection," *BMC Bioinformatics*, vol. 7, no. 91, 2006.

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

[11] A. Airola, T. Pahikkala, W. Waegeman, B. De Baets, and T. Salakoski, "A comparison of AUC estimators in small-sample studies," in *JMLR Workshop and Conference Proceedings: Machine Learning in Systems Biology*, S. Džeroski, P. Geurts, and J. Rousu, Eds., 2010, vol. 8, pp. 3–13.

[12] P. Golland, F. Liang, S. Mukherjee, and D. Panchenko, "Permutation tests for classification," in *Proceedings of COLT 2005*, ser. Lecture Notes in Computer Science, P. Auer and R. Meir, Eds., vol. 3559. Springer, 2005, pp. 501–515.

[13] M. D. Radmacher, L. M. McShane, and R. Simon, "A paradigm for class prediction using gene expression profiles," *J Comput Biol*, vol. 9, no. 3, pp. 505–511, 2002.

[14] P. Radivojac, Z. Obradovic, A. K. Dunker, and S. Vucetic, "Feature selection filters based on the permutation test," in *Proceedings of ECML/PKDD 2004*, 2004, pp. 334–346.

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

[16] R. Rifkin, "Everything old is new again: A fresh look at historical approaches in machine learning," Ph.D. dissertation, MIT, 2002.

[17] E. K. Tang, P. N. Suganthan, and X. Yao, "Gene selection algorithms for microarray data based on least squares support vector machine," *BMC Bioinformatics*, vol. 7, no. 95, 2006.

[18] T. Pahikkala, A. Airola, J. Boberg, and T. Salakoski, "Speeding up greedy forward selection for regularized least-squares," in *Proceedings of ICMLA 2010*. IEEE Press, 2010.

[19] U. Alon, N. Barkai, D. A. Notterman, K. Gishdagger, S. Ybarradagger, D. Mackdagger, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *P Natl Acad Sci USA*, vol. 96, no. 12, pp. 6745–6750, 1999.

[20] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. A. Olson, J. R. Marks, and J. R. Nevins, "Predicting the clinical status of human breast cancer by using gene expression profiles," *P Natl Acad Sci USA*, vol. 98, no. 20, pp. 11 462–11 467, September 2001.

[21] T. Pahikkala, A. Airola, P. Naula, and T. Salakoski, "Greedy RankRLS: a linear time algorithm for learning sparse ranking models," in *SIGIR 2010 Workshop on Feature Generation and Selection for Information Retrieval*, E. Gabrilovich, A. J. Smola, and N. Tishby, Eds., 2010.