4th International Conference on Computational Systems-Biology and Bioinformatics, CSBio2013

An efficient and effective wrapper based on paired t-test for learning naive Bayes classifiers from large-scale domains

Chanju Kim\textsuperscript{a}, Honglan Li\textsuperscript{a}, Soo-Yong Shin\textsuperscript{b}, Kyu-Baek Hwang\textsuperscript{a,*}

\textsuperscript{a}School of Computer Science and Engineering, Soongsil University, Seoul 156-743, Korea
\textsuperscript{b}Department of Clinical Epidemiology & Biostatistics, Asan Medical Center & University of Ulsan College of Medicine, Seoul 138-736, Korea

Abstract

Feature selection is one of the crucial steps in supervised learning, which influences the entire subsequent classification (or regression) process. The approaches to this task can largely be divided into two categories: filter-based and wrapper-based methods. Generally, the latter produces better results than the former with regard to given learning methods, though it consumes more computational resources for searches over the feature subset space. In this paper, we propose an Efficient wRapper based on a Paired \(t\)-Test (ERPT) for choosing features from large-scale data consisting of thousands of variables, such as microarrays. Statistical tests are a reasonable option when the number of features is very large because they have more predictable behavior and can be more efficient than most search methods. The proposed method consists of two phases: decrement phase and increment phase. In the decrement phase, it selects \textit{strongly relevant} features. In the increment phase, it adds \textit{weakly relevant} features, given the previously selected features. Our method, combined with naive Bayes classifiers, has been tested in an extensive set of experiments on University of California Irvine (UCI) Machine Learning Repository data. The results showed that the performance of the proposed method is comparable to that of the backward search-based wrapper and superior to that of the forward search-based wrapper. Furthermore, it demonstrated much better performance than the forward search-based wrapper when applied to three microarray data sets, for which the backward search-based wrapper was impractical because of the computational burden involved. The proposed method has the following three merits: (1) it is applicable to data sets having thousands of variables, (2) it provides a theoretically sound and controllable criterion for thresholding features, and (3) it finds feature subsets for the maximizing of classification performance on sparse domains.

© 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of the Program Committee of CSBio2013

Keywords: feature selection; wrappers; Naive Bayes classifiers; microarray data

\* Corresponding author. Tel.: +82-2-820-0925 ;
E-mail address: kbhwang@ssu.ac.kr

© 2013 The Authors. Published by Elsevier B.V. Open access under CC BY-NC-ND license.
Selection and peer-review under responsibility of the Program Committee of CSBio2013
doi:10.1016/j.procs.2013.10.014
1. Introduction

Feature selection is the process of reducing the dimensionality of data by removing irrelevant features. It is one of the most important steps in pattern recognition, machine learning, and data mining. Feature selection strongly affects classification, regression, and clustering performance and has been actively applied in diverse areas, such as bioinformatics, text categorization and classification, image retrieval, customer retention, and intrusion detection. The various purposes of feature selection can be summarized as follows: (1) to improve model performance and avoid overfitting, (2) to build a simple and fast model and (3) to gain insight into data and facilitate data visualization. Of these three, the focus of this study is the enhancement of model performance.

Feature selection methods are categorized into filter, wrapper, and embedded methods. Filter methods select feature subsets independent of induction algorithms. Wrapper methods use induction algorithms to select features. Embedded methods refer to the feature selection processes included in classifier learning algorithms, e.g., decision tree learning.

Wrapper methods select features that are biased to a specific induction algorithm, and their performance is known to generally be better than that of filter methods. Most wrappers rely on search heuristics to find optimal feature subsets. The search space is exponential in terms of the number of features. Thus, a greedy search is usually adopted. Two types of greedy searches are forward selection, which starts from an empty set, and backward elimination, which starts with all features. Forward search is usually faster but less accurate than backward search.

Applications in large-scale domains such as microarrays or text classification are challenging because of the curse of dimensionality. Furthermore, many microarray data sets have a small number of examples (usually less than 100), which makes them prone to overfitting. As a consequence, filter methods have been widely adopted for microarray classification. Wrapper methods have been employed in restricted forms, e.g., a population-based probabilistic search suited for dealing with a relatively small number of genes, and forward greedy searches. Some wrappers with greedy search are impractical when the number of features exceeds the thousands. For example, backward elimination wrappers are inapplicable to microarray data on normal desktop computers, even though they are expected to deliver better performance than other wrappers and filters.

In this paper, we propose a two-step wrapper that does not use a search method, but rather uses a statistical test to resolve the above issue. The proposed method is called ERPT (Efficient wRapper based on a Paired t-Test). The method is based on the definitions of the strong and weak relevances of features. ERPT chooses strongly relevant features for classification and then adds weakly relevant features on the basis of the information on the strongly relevant features. Statistical tests with cross validation are applied to the task of distinguishing significantly relevant features from those that are less so. The use of statistical tests instead of searches offers the following advantages. First, the computational complexity of tests is linearly proportional to the number of features to be examined. Hence, the running time of statistical test-based methods is predictable in principle. In contrast, the running time of most search heuristics increases on much more than a linear scale as the number of features grows and is hardly predictable. The second merit is that the computational resources that are saved by avoiding a search can be put to use segregating the real relevances from false positives by means of statistical tests. This advantage is particularly great when the given data is sparse and noisy. Thus, the proposed method is highly appropriate when it is deployed for sparse data with a large number of noisy variables. The third advantage of statistical tests is that they provide a theoretically sound threshold for the selection of features, such as significance levels.

We demonstrated the performance of our method in two kinds of experiments. First, using 18 UCI Machine Learning Repository data sets, we showed that our method achieves classification performance in terms of the area under the ROC curve (AUC) that is comparable to that of backward elimination, and superior to that of forward selection. Second, the appropriateness of ERPT in sparse domains consisting of thousands of variables was demonstrated via the experiments using three real-world microarray data sets. In this case, our approach substantially outperformed filter-based feature selection and the forward search-based wrappers. The backward search-based wrappers were not applicable to these huge data sets. All the experiments were performed with naive Bayes classifiers used as an induction algorithm because of their simple and efficient learning and evaluation properties. However, other learning algorithms can also be used with ERPT.
This paper is organized as follows. In Section 2, we give a detailed description of the proposed model. Next, we present the experimental results of our method in terms of its performance in Section 3. Finally, concluding remarks are presented in Section 4.

2. An Efficient Wrapper Based on a Paired \( t \)-Test

In this section, the concept of feature relevance is introduced and then our method for distilling relevant features is described.

2.1. Feature Relevance

In general, the degree of relevance of a feature to the target variable is stratified by the following three definitions. Let us suppose that \( F = \{F_1, F_2, \ldots, F_n\} \) is a set of \( n \) feature variables. \( F_i \) denotes the \( i \)-th feature and \( F_i' = F \setminus \{F_i\} \). Also, \( F_i' \) is a subset of \( F_i \). \( C \) is the class variable and \( P(C \mid F) \) is the conditional probability distribution of class given the feature set \( F \).

Definition 1: Strong relevance

A feature \( F_i \) is strongly relevant iff

\[
P(C \mid F_i, F_i') \neq P(C \mid F_i').
\]

Definition 2: Weak relevance

A feature \( F_i \) is weakly relevant iff

\[
P(C \mid F_i, F_i') = P(C \mid F_i') \quad \text{and} \quad \exists F \subset F_i \quad \text{such that} \quad P(C \mid F_i, F_i') \neq P(C \mid F_i').
\]

Definition 2: Irrelevance

A feature \( F_i \) is irrelevant iff

\[
\forall F \subseteq F_i, \quad P(C \mid F_i, F_i') = P(C \mid F_i').
\]

Weakly relevant features are further divided into redundant and non-redundant ones. An optimal classifier uses all strongly relevant features and non-redundant weakly relevant features if the true underlying distribution is known. Fig. 1 illustrates the feature relevance hierarchy and an optimal feature subset.

![Fig. 1. Relevance and redundancy of feature.](image)
2.2. ERPT

As shown in the previous subsection, feature relevance is defined in terms of the difference of probability distributions. Such probability distributions are estimated from data by a machine learning algorithm in practice. In other words, the true underlying distribution is inaccessible in most cases. As a result, even relevant features could degrade the classification performance of a learned classifier since the estimation problem is NP-hard in many cases and data usually contains noise\textsuperscript{11}. Thus, the previous definitions of feature relevance should be adapted for practical situations. Because wrapper-based feature selection methods consider the performance of an induction algorithm given a feature set, performance measures can be utilized for such an adaptation. In our approach, the AUC (area under the ROC curve)\textsuperscript{20} is adopted as such a measure due to its popularity in pattern recognition. It has also been suggested as a better measure than accuracy\textsuperscript{21}.

The AUC score obtained from a cross validation test on a data set $D$ with an induction algorithm $I$ for a feature set $F$ is denoted as $\text{AUC}(F, I, D)$. Then, $\text{AUC}(F, I, D) \neq \text{AUC}(F, I, D)$ with high probabilities if $F$ is strongly relevant as previously defined. Since some strongly relevant features might worsen the classification performance, we define positively strongly-relevant features as the features which are strongly relevant and can improve the AUC score. More precisely, a feature $F$ is positively strongly-relevant if

$$\text{AUC}(F, I, D) \neq \text{AUC}(F, I, D).$$

(1)

When $\text{AUC}(F, I, D) = \text{AUC}(F, I, D)$, $F$ is highly probable to be weakly relevant or irrelevant. Among such features, we distinguish positively weakly-relevant features as follows. Let $S$ be the set of positively strongly-relevant features. A feature $F$ is positively weakly-relevant if

$$\text{AUC}(S, I, D) < \text{AUC}(S \cup \{F\}, I, D) \quad (F \notin S).$$

(2)

Our wrapper method aims at distilling positively strongly-relevant and weakly-relevant features from data containing thousands of variables with a relatively small number of instances. In specific, the one-sided paired $t$-test is utilized for finding the features satisfying Equations (1) and (2) in a statistically significant way. The paired $t$-test is a standard method used for (dis)confirming the fact that the mean values of two paired sets are different from each other. There are several options for splitting data when applying a statistical test such as resampling, cross validation, and average-over-folds\textsuperscript{22}. Among such methods, a combination of random split and $n$-fold cross validation, i.e., average-over-folds, was adopted since it has been shown to be economic and having acceptable power when used with paired $t$-tests. In the average-over-folds approach, cross validation is repeated with random division. Then, each cross validation trial is averaged over its folds and considered as one sample value (see $\text{CVEvaluate}()$ in Fig. 2).

To summarize, our method, ERPT (Efficient wRapper based on a Paired $t$-Test) is composed of decrement and increment phases. In the decrement phase, it extracts positively strongly-relevant features from all given features. Then, in the increment phase, it adds positively weakly-relevant features based on the selected features in the previous phase. Fig. 2 details the ERPT algorithm.

3. Experimental Evaluation

Two kinds of experiments were conducted to compare ERPT with other feature selection methods. In the first experiment, we used 18 data sets from the UCI Machine Learning Repository\textsuperscript{23}. In the second experiment, we used three microarray data sets having thousands of attributes and less than two hundred instances.

The compared search-based wrappers included FSS (forward sequential selection), BSE (backward sequential elimination), BFF (best-first forward selection), and BFB (best-first backward elimination)\textsuperscript{11}. In addition, two famous filters, i.e., information gain (IG)\textsuperscript{24} and Relief\textsuperscript{25}, were compared with our method. The naive Bayes classifier was employed for the experiments due to its computational efficiency. However, BSE and BFB were not applicable to the microarray data sets even with the efficient naive Bayes classifier.
Input: significance level $\alpha$; cross validation fold $k$;  
number of ERPT repetitions $N$; initial feature set $F$;  
induction algorithm $I$; training data $D$  
Output: selected feature subset $S$  
$\text{ERPT}(\alpha, k, N, F, I, D)$  
Decrement: 
\begin{align*}
\text{for } j &= 1 \rightarrow N \\
\quad \text{AUC}_{\text{baseline}} &= \text{CVEvaluate}(j, k, F, I, D) \\
\quad \text{for } F_i \text{ in } F \\
\quad \quad d_i &= \text{AUC}_{\text{baseline}} - \text{CVEvaluate}(j, k, F_i, I, D) \\
\quad \text{endfor}
\end{align*}
\begin{align*}
\text{mean}_d &= \text{mean of } \{d_1, d_2, \ldots, d_N\} \\
\text{std}_d &= \text{standard deviation of } \{d_1, d_2, \ldots, d_N\} \\
S &= \text{all } F_i \text{'s satisfying } \frac{\text{mean}_{d_i}}{\text{std}_{d_i}} > t_{N-1, \alpha} \quad \text{// one-side paired t-test}
\end{align*}
Increment: 
\begin{align*}
\text{for } j &= 1 \rightarrow N \\
\quad \text{AUC}_{\text{baseline}} &= \text{CVEvaluate}(j, k, S, I, D) \\
\quad \text{for } F_i \text{ in } F - S \\
\quad \quad d_i &= \text{AUC}_{\text{baseline}} - \text{CVEvaluate}(j, k, S \cup \{F_i\}, I, D) \\
\quad \text{endfor}
\end{align*}
\begin{align*}
\text{mean}_d &= \text{mean of } \{d_1, d_2, \ldots, d_N\} \\
\text{std}_d &= \text{standard deviation of } \{d_1, d_2, \ldots, d_N\} \\
S &= \text{all } F_i \text{'s in } F - S \text{ satisfying } \frac{\text{mean}_{d_i}}{\text{std}_{d_i}} < t_{N-1, \alpha} \quad \text{// one-side paired t-test}
\end{align*}
Return $S$
End

$\text{CVEvaluate}(j, k, F, I, D)$: Return average of the AUC scores of $I$  
using $F$ on $D$ over $k$-fold cross validation 
with $j$ as seed for random splitting.

Fig. 2. The ERPT algorithm. Here, $t_{N-1,\alpha}$ means the t-statistics value with degrees of freedom $N - 1$ for one-sided region $\alpha(1-\alpha)$.

The WEKA API\cite{weka} was used for implementing ERPT and adopting the other wrappers and the filters in our experiments. All the wrapper-based methods including ERPT adopted the weighted AUC score as the criterion function. All the other parameters followed the default setting in WEKA. Continuous features were discretized prior to learning and evaluation. A minimum description length (MDL) principle-based discretization method proposed by Fayyad and Irani\cite{fayyad} was applied to training data. Then, the acquired boundaries for each continuous feature were applied to test data for discretization. For the filter-based methods, the default threshold value (zero) was adopted.
3.1. Experimental Results on the Data Sets from the UCI ML Repository

Table 1 shows the description of the data sets used in the experiments. The number of features ranges from 4 to 60. Sparsity was calculated by dividing the number of features by the number of instances.

The experimental process was as follows: (1) Randomly split the original data set into training and test data sets of which ratio is 7 to 3. (2) Apply a feature selection method to the training data set. (3) Learn a classifier from the training data set with the selected features. (4) Evaluate the performance of the classifier with the selected features on the test data set. We repeated the above procedure 50 times for each data set.

Table 1. Description of the UCI ML Repository data sets. Sparsity is defined as $\frac{\text{# of features}}{\text{# of instances}}$.

| DataSet Name     | # of Instances | # of Features | Nominal Features | Continuous Features | # of Classes | Sparsity |
|------------------|----------------|--------------|------------------|--------------------|--------------|----------|
| Sonar            | 208            | 60           | 0                | 60                 | 2            | 0.2885   |
| Ionosphere       | 351            | 34           | 0                | 34                 | 2            | 0.0969   |
| SPECT heart      | 267            | 22           | 0                | 22                 | 2            | 0.0824   |
| Wine             | 178            | 13           | 0                | 13                 | 3            | 0.0730   |
| Horse-colic      | 368            | 22           | 15               | 7                  | 2            | 0.0598   |
| Wdbc             | 569            | 30           | 0                | 30                 | 2            | 0.0527   |
| Soybean-large    | 683            | 35           | 35               | 0                  | 19           | 0.0512   |
| Glass            | 214            | 9            | 0                | 9                  | 6            | 0.0421   |
| Vote             | 435            | 16           | 16               | 0                  | 2            | 0.0368   |
| Iris             | 150            | 4            | 0                | 4                  | 3            | 0.0267   |
| Crx              | 690            | 15           | 9                | 6                  | 2            | 0.0217   |
| Breast cancer    | 699            | 9            | 0                | 9                  | 2            | 0.0129   |
| Pima             | 768            | 8            | 0                | 8                  | 2            | 0.0104   |
| Segmentation     | 2,310          | 19           | 0                | 19                 | 7            | 0.0082   |
| Mushroom         | 8,124          | 22           | 22               | 0                  | 2            | 0.0027   |
| Pen digits       | 10,992         | 16           | 0                | 16                 | 10           | 0.0015   |
| Letter recog.    | 20,000         | 16           | 0                | 16                 | 26           | 0.0008   |
| Adult            | 48,842         | 14           | 8                | 6                  | 2            | 0.0003   |

The number of repetitions for the decrement and increment phases was set to 30. At each iteration of ERPT (see $CVEvaluate(j, k, \{F, I, D\}$ in Fig. 2) and each search step of the other wrappers, 5-fold cross validation was applied to weighted AUC score calculation. The significance level of the paired $t$-test was 0.05. Table 2 compares the average and standard deviation of weighted AUC scores from the 50 trials for each data set. In average, backward sequential elimination (BSE) was the best. ERPT was the second best, although it showed the best performance on only two data sets, i.e., ‘Ionosphere’ and ‘Letter recog.’ To summarize, ERPT was slightly worse than BSE, but better than the forward search-based methods (FSS and BFF), best first backward elimination (BFB), and the filters (IG and Relief). This result confirms the fact that ERPT is stable and comparable to other search-based wrappers and famous filters.

For the top four most sparse data sets, i.e., ‘Sonar’, ‘Ionosphere’, ‘SPECT heart’, and ‘Wine’, EPRT showed the best performance among the wrappers except ‘Sonar’ on which EPRT achieved the second best result. For the four least sparse data sets, ERPT showed the worst performance among the wrappers in ‘Mushroom’, ‘Pen digits’ and ‘Adult’ data sets. However, the difference of performance was not significant, and ERPT outperformed the filter methods.

Table 3 shows the average number of selected features. Among the search-based wrappers, BSE and BFB chose more features than FSS and BFF. EPRT selected more features than them. The filter-based methods chose much more features compared to the wrapper-based methods.
data sets extremely sparse. We used the following three microarray data sets: acute leukemia data 29, prostate cancer
the experimental cost, however, the number of examples is usually less than a hundred, making general microarray
Segmentation
Letter recog.
Horse-colic
Mushroom
Wdbc
Glass
Vote
Iris
Crx
Breast cancer
Pima
Segmentation
Mushroom
Pen digits
Letter recog.
Adult
Average
Table 3. Number of selected features of each method (mean ± standard deviation).

| Data Set   | ERPT Fwd (FSS) | ERPT Back (BSE) | Sequential Fwd (FSS) | Sequential Back (BSE) | Best First Fwd (BFF) | Best First Back (BFB) | Filter IG | Filter Relief |
|------------|----------------|-----------------|----------------------|-----------------------|----------------------|-----------------------|-----------|---------------|
| Sonar      | 8.58±2.57      | 12.86±2.52      | 8.42±2.14            | 13.14±3.27            | 17.18±3.04           | 17.12±3.02            |           |               |
| Ionsosphere| 10.02±2.06     | 13.38±3.53      | 10.18±1.87           | 11.90±2.38            | 31.10±1.02           | 30.88±1.08            |           |               |
| SPECT heart| 10.02±1.76     | 11.18±1.81      | 10.34±1.79           | 9.96±1.58             | 22.00±0.00           | 21.28±0.64            |           |               |
| Wine       | 8.46±1.50      | 9.28±1.31       | 8.72±1.39            | 8.70±1.02             | 13.00±0.00           | 13.00±0.00            |           |               |
| Horse-colic| 6.54±1.27      | 6.86±1.25       | 6.60±1.47            | 7.40±1.63             | 20.22±0.79           | 18.66±1.08            |           |               |
| Wdbc       | 8.86±2.02      | 11.52±1.99      | 8.40±2.14            | 10.96±1.60            | 26.24±0.72           | 26.24±0.72            |           |               |
| Soybean-large| 16.74±1.99    | 19.78±2.35      | 16.84±2.49           | 18.32±2.13            | 35.00±0.00           | 35.00±0.00            |           |               |
| Glass      | 5.83±0.88      | 5.21±1.23       | 5.38±1.18            | 5.46±1.03             | 6.62±0.73            | 6.62±0.73             |           |               |
| Vote       | 4.94±0.71      | 5.20±0.83       | 4.88±0.72            | 4.80±0.76             | 16.00±0.00           | 16.00±0.00            |           |               |
| Iris       | 2.58±0.70      | 2.48±0.61       | 2.54±0.61            | 2.58±0.64             | 4.00±0.00            | 4.00±0.00             |           |               |
| Crx        | 7.06±1.04      | 7.44±1.28       | 7.28±1.16            | 7.36±1.12             | 14.08±0.60           | 14.08±0.60            |           |               |
| Breast cancer| 5.58±0.76    | 5.64±0.63       | 5.68±0.94            | 5.46±0.73             | 9.00±0.00            | 9.00±0.00             |           |               |
| Pima       | 3.74±0.69      | 3.94±0.79       | 3.68±0.74            | 3.74±0.69             | 5.02±0.68            | 4.88±0.72             |           |               |
| Segmentation| 7.20±0.99     | 7.70±0.74       | 7.18±0.90            | 7.60±0.90             | 17.10±0.36           | 16.94±0.24            |           |               |
| Mushroom   | 6.38±1.14      | 12.46±2.03      | 5.72±1.14            | 11.26±1.66            | 21.00±0.00           | 21.00±0.00            |           |               |
| Pen digits | 12.46±1.31     | 13.66±0.69      | 12.98±1.20           | 13.32±1.04            | 16.00±0.00           | 16.00±0.00            |           |               |
| Letter recog. | 13.00±0.00  | 13.00±0.00      | 13.00±0.00           | 13.00±0.00            | 15.00±0.00           | 15.00±0.00            |           |               |
| Adult      | 9.06±0.24      | 9.06±0.27       | 9.04±0.20            | 9.06±0.24             | 13.86±0.24           | 13.06±0.24            |           |               |
| Average    | 8.14±1.22      | 9.51±1.31       | 8.16±1.23            | 9.11±1.23             | 16.76±0.45           | 16.60±0.50            |           |               |

3.2. Experimental Results on Microarray Data Sets

Microarray technology has become routinely used for gene expression profiling. Each gene of an organism corresponds to a feature of microarray data. Thus, the number of features amounts to more than thousands. Due to the experimental cost, however, the number of examples is usually less than a hundred, making general microarray data sets extremely sparse. We used the following three microarray data sets: acute leukemia data, prostate cancer data, and colon cancer data. Table 4 describes the characteristics of the three microarray data sets. The acute leukemia and prostate cancer data sets consist of separate training and test data sets. All the data sets have been used...
for binary classification. The colon and prostate cancer data sets have been used for discriminating tumor from normal samples. The acute leukemia data set has been used for discriminating between two disease subtypes, i.e., acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). These data sets are extremely sparse compared to the UCI ML Repository data sets shown in Table 1.

Table 4. Description of the microarray data sets.

| DataSet      | # of Training | # of Test | # of Features | # of Classes | Classes       | Sparsity     |
|--------------|---------------|-----------|---------------|--------------|---------------|--------------|
| Acute leukemia | 38            | 34        | 7,129         | 2            | ALL, AML      | 99.01        |
| Prostate cancer | 102           | 34        | 12,600        | 2            | tumor, normal | 92.65        |
| Colon cancer  | 62            | N/A       | 2,000         | 2            | tumor, normal | 32.26        |

Because the colon cancer data set does not have a separate test set, we randomly split it into training and test sets of which ratio is 7 to 3. Then, the experiment was repeated 50 times. For the acute leukemia and prostate cancer data sets, the performance was evaluated on the separate test sets and the repetition number was ten due to the computational burden. The number of repetitions for the decrement and increment phases of ERPT was 30 and the significance level for the paired t-test was set to 0.05. For the weighted AUC score calculation at each iteration of ERPT and each search step of the other wrappers, 3-fold cross validation was applied since the size of the test data set produced by 5-fold division was too small for obtaining meaningful AUC scores. All the experiments were performed on a PC server with an Intel Xeon CPU (2GHz) and 32GB of RAM.

Table 5 summarizes the results on the acute leukemia, prostate cancer, and colon cancer data sets, respectively. With respect to the classification performance, ERPT was the best as it achieved the highest average AUC score on all of three microarray data sets. Furthermore, the results from ERPT were significantly better than those from the other feature selection methods in most cases. Table 6 shows the comparison results between ERPT and the others in terms of the statistical significance. Here, only one comparison case (ERPT vs. BFF on the colon cancer data set) showed statistically insignificant difference (p-value>0.05).

Table 5. Results on the microarray data sets. The backward search-based methods were not able to produce any results (Failed). Running time without feature selection (All) is zero.

|              | Acute leukemia | Prostate cancer | Colon cancer |
|--------------|----------------|-----------------|--------------|
|              | AUC Score      | # of Features   | Time (seconds) | AUC Score | # of Features | Time (seconds) | AUC Score | # of Features | Time (seconds) |
| ERPT         | 95.79 ±4.66   | 30.20 ±28.84   | 16.73 ±4.881 | 78.35 ±11.64 | 3.926 ±5.679 | 258.048 ±72.387 | 85.99 ±11.02 | 3.926 ±5.679 | 258.048 ±72.387 |
| Sequential   |               |                 |               |            |               |                |            |               |                |
| Fwd (FSS)    | 76.95 ±12.76  | 2.20 ±0.40     | 101.8 ±29.6   | 66.18 ±11.72 | 8.00 ±1.10   | 609.9 ±132.8   | 83.62 ±8.65  | 9.56 ±2.72   | 89.0 ±56.9     |
| Back (BSE)   | N/A            | N/A             | Failed        | N/A        | N/A           | Failed         | N/A        | N/A           | Failed         |
| Best First   |               |                 |               |            |               |                |            |               |                |
| Fwd (BEF)    | 72.59 ±10.26  | 2.00 ±0.00     | 233.3 ±56.7   | 68.49 ±8.87 | 9.00 ±2.37   | 1028 ±226.6    | 84.08 ±9.47  | 10.86 ±3.52  | 125.7 ±64.8   |
| Back (BFB)   | N/A            | N/A             | Failed        | N/A        | N/A           | Failed         | N/A        | N/A           | Failed         |
| Filter       |               |                 |               |            |               |                |            |               |                |
| IG           | 93.11 ±0.9    | 866 ±0.2       | 64.71 ±1.554  | 2.0 ±0.2    | 106.78 ±11.11 | ±37.41 ±0.0   | 83.05 ±11.11 | ±37.41 ±0.0   |
| Relief       | 93.11 ±1.9    | 866 ±0.4       | 57.35 ±1.482  | 8.9 ±2.6    | 106.78 ±11.11 | ±37.41 ±0.0   | 83.05 ±11.11 | ±37.41 ±0.0   |
| All          | 93.11 ±0.9    | 7,129 ±0.4     | 60.29 ±12.600 | 0 ±2.6      | 80.36 ±12.13  | 2,000 ±0.0     |

Backward sequential elimination (BSE) might be expected to perform better than ERPT as in the experiments on the UCI Machine Learning Repository data sets (see Table 2) However, it was not able to produce any results in proper time in the same computing environment. In our experiments, the filter methods were the fastest, taking less than few seconds in general. BFF and FSS usually took several minutes. ERPT required pretty longer time than them. In the case of the acute leukemia data set, the average running time of ERPT was about 4.64 hours. For the
prostate cancer data set, our method took about 71.68 hours in general. In the case of the colon cancer data set, the average execution time of ERPT was about 21 minutes. The actual running time of ERPT was influenced much by various factors such as the process scheduling of the operating system when the amount of necessary computation was huge. Thus, ERPT showed large variations in its running time for the acute leukemia and prostate cancer data sets. However, in the case of the relatively small colon cancer data set, the influence of such environmental factors was attenuated. For the colon cancer data set (Table 5), we were able to observe that the running time of ERPT is more stable and predictable than that of the forward search-based wrappers. Even though the running time of ERPT was much longer than most of the other methods, it should be emphasized that the backward search-based wrappers (BSE and BFB) failed to produce the results on the three microarray data sets. Thus, our method can be regarded as a reasonable choice for maximizing the classification performance when the backward search-based methods are inapplicable due to their computational burden.

Table 6. P-values from the one-sided student’s t-test (vs. ERPT).

|          | Acute leukemia | Prostate cancer | Colon cancer |
|----------|----------------|-----------------|--------------|
| FSS      | 0.0003         | 0.020           | 0.030        |
| BFF      | 3.092E-06      | 0.029           | 0.068        |
| IG       | N/A            | N/A             | 5.43E-05     |
| Relief   | N/A            | N/A             | 5.43E-05     |

Table 5 also compares ERPT with the other methods in the viewpoint of the number of selected features (column name: # of features). From the acute leukemia data set, ERPT chose a larger number of features than FSS and BFF, but a smaller number of features compared to IG and Relief in average. These results coincide with those on the UCI Machine Learning Repository data sets (see Table 3). In the cases of the prostate and colon cancer data sets, however, the average number of features selected by ERPT is the largest. This is caused by the cases of selecting all the features. On the prostate cancer data set, three out of the ten trials (30%) by ERPT resulted in selecting all 12,600 features. In the case of the colon cancer data set, ERPT chose all 2,000 features in ten out of the 50 repetitions (20%). Table 7 shows the distribution of the number of selected features from the three microarray data sets.

Table 7. Details on the number of selected features (Min: minimum, Max: maximum, and Med: median values).

|          | Acute leukemia | Prostate cancer | Colon cancer |
|----------|----------------|-----------------|--------------|
| ERPT     | Min 7          | Max 115         | Med 24       |
|          | Min 109        | Max 12,600      | Med 286.5    |
|          | Min 15         | Max 2,000       | Med 43.5     |
| FSS      | 2              | 3               | 2            |
| BFF      | 6              | 10              | 8            |
| IG       | 2              | 2               | 2            |
| Relief   | 7              | 15              | 8            |
| All      | 7,129          | 12,600          | 2,000        |

If we exclude the extreme cases choosing all features from the prostate and colon cancer data sets, ERPT usually selected more features than the forward search-based wrappers (FSS and BFF) and less features than the filters (IG and Relief). In order to investigate the relationship among the selected features by different methods, we picked up the cases in which the number of selected features is similar to the median value (Table 7). Fig. 3 illustrates the overlap among the selected features by ERPT, FSS, and IG from the chosen runs in our experiments. In the acute leukemia and prostate cancer data sets, all the features chosen by ERPT and FSS were included in the feature subset produced by IG. In the colon cancer data set, most features selected by ERPT and FSS were included in the feature subset by IG. Because ERPT tries to select both the strongly and weakly relevant features, the features chosen only by FSS or IG are highly probable to be the irrelevant ones (see Fig. 1). Information gain is not a good measure for selecting the features relevant to the given classification problem. On the acute leukemia data set, among the top 26 (which is the size of the feature subset by ERPT) features having highest information gain values, only ten were included in the feature subset produced by ERPT. On the prostate cancer data set, 168 of the top 233 features having highest information gain values were chosen by ERPT. On the colon cancer data set, 32 of the top 43 features...
having highest information gain values were included in the subset by ERPT. Interestingly, not all the features selected by FSS were included in the feature subset by ERPT, even though the number of features selected by FSS is usually much smaller than the number of features selected by ERPT. On the sparse and noisy microarray data sets, FSS seems to have picked up some irrelevant features and ended up on a local maximum due to its greedy strategy for search.

![Image](image-url)

Fig. 3. Overlap of the selected features by different methods from the (a) acute leukemia, (b) prostate cancer, and (c) colon cancer data sets.

4. Conclusion

Large-scale sparse domains impose several challenges on classification and feature selection. In such cases, the heavy use of statistical tests instead of searches could be suitable for alleviating problems such as overfitting. Also, some search heuristics, e.g., backward sequential search, are impractical for the analysis of data having thousands of variables. We have proposed an approach that is able to choose feature subsets from large-scale sparse data in order to maximize the classification performance. An extensive set of experiments on benchmark data sets showed that the proposed method achieves a level of performance comparable to that of backward search-based wrappers and superior to that of forward search-based wrappers and several filters with respect to classification accuracy. In particular, our method produced substantially more accurate classification results than forward search-based methods and filters when applied to real world microarray data sets, which are a typical example of large-scale sparse domains that discourage the use of backward search-based wrappers. One direction for further research would be an explicit consideration of feature redundancy issues. Hybridization between our method and a filtering approach could be utilized for such a purpose. Another plausible direction would be the adaptation of our method for a specific learning method.

Acknowledgements

This work was supported by Basic Science Research Program (2013054061) through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning.

References

1. Dash M, Liu H. Feature selection for classification. *Intell. Data Analysis* 1997; 1:131-156.
2. Guyon I, Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* 2003;3:1157-1182.
3. Xing EP, Jordan MI, Karp RM. Feature selection for high-dimensional genomic microarray data. *Proc. 18th Int. Conf. Mach. Learn. (ICML)*; 2001. p. 601-608.
4. Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. *Bioinformatics* 2007;23(19):2507-2517.
5. Yang Y, Pedersen JO. A comparative study on feature selection in text categorization. *Proc. 14th Int. Conf. Mach. Learn. (ICML)*;1997. p. 412-420.
6. Forman G. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.* 2003;3:1157-1182.
7. Swets DL, Weng JJ. Efficient content-based image retrieval using automatic feature selection. *Proc. Int. Symp. Computer Vision*; 1995. p. 85-90.

8. Ng K, Liu H. Customer retention via data mining. *Artif. Intell. Rev.* 2000;14(6): 569-590.

9. Lee, W., Stolfo, S. J., Mok, K. W., 2000. Adaptive intrusion detection: a data mining approach, *Artif. Intell. Rev.,* Springer, Vol. 14(6), pp.533-567.

10. Blum AL, Langley P. Selection of relevant features and examples in machine learning. *Artif. Intell. 1997;*97(1-2):245-271.

11. Kohavi R, John GH. Wrappers for feature subset selection. *Artif. Intell. 1997;*97(1-2):273-324.

12. Hwang KB, Zhang BT. Bayesian model averaging of Bayesian network classifiers over multiple node-orders: application to sparse datasets. *IEEE Trans. Syst. Man. Cybern. B Cybern.* 2005;35(6):1302-1310.

13. Liu H, Motoda H. Computational methods of feature selection. Chapman & Hall/CRC; 2008.

14. Li L, Weinberg CR, Darden TA, Pedersen LG. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics* 2001;17(12):1131-1142.

15. Ooi C, Tan P. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data, *Bioinformatic 2003;*19(1):37-44.

16. Blanco R, Larañaga P, Inza I and Sierra B. Gene selection for cancer classification using wrapper approaches. *Intern. J. Pattern Recognit. Artif. Intell. 2004;*18(8):1373-1390.

17. Jirapech-Umpai T, Aitken S. Feature Feature selection and classification for microarray data analysis: evolutionary methods for identifying predictive genes. *BMC Bioinformatics* 2005;6:148.

18. Xiong M, Fang X, Zhao J. Biomarker identification by feature wrappers. *Genome Res.* 2001;11:1878-1887.

19. Inza I, Larañaga P, Blanco R, Cerrolaza AJ. Filter versus wrapper gene selection approaches in DNA microarray domains. *Artif. Intell. Med.* 2004;31(2):91-103.

20. Fawcett T. An introduction to ROC analysis. *Pattern Recognit. Lett.* 2006;27(8):861-874.

21. Ling CX, Huang J, Zhang H. AUC: a statistically consistent and more discriminating measure than accuracy. *Proc. 18th Intern. J. Conf. Artif. Intell. (IJCAI);* 2003. p.329-341.

22. Bouckaert RR. Estimating replicability of classifier learning experiments. *Pattern Recognit. Lett.* ACM,New York, NY, USA; 2004. p.15.

23. Asuncion A, Newman DJ. UCI machine learning repository. *University of California, Irvine, School of Information and Computer Sciences.* http://www.ics.uci.edu/~mlearn/MLRepository.html; 2007.

24. Mitchell T. Computational methods of feature selection. McGraw-Hill; 1997.

25. Fawcett T. An introduction to ROC analysis. *Pattern Recognit. Lett.* 2006;27(8):861-874.

26. Ling CX, Huang J, Zhang H. AUC: a statistically consistent and more discriminating measure than accuracy. *Proc. 18th Intern. J. Conf. Artif. Intell. (IJCAI);* 2003. p.329-341.

27. Bouckaert RR. Estimating replicability of classifier learning experiments. *Pattern Recognit. Lett.* ACM,New York, NY, USA; 2004. p.15.

28. Asuncion A, Newman DJ. UCI machine learning repository. *University of California, Irvine, School of Information and Computer Sciences.* http://www.ics.uci.edu/~mlearn/MLRepository.html; 2007.

29. Mitchell T. Computational methods of feature selection. McGraw-Hill; 1997.

30. Fawcett T. An introduction to ROC analysis. *Pattern Recognit. Lett.* 2006;27(8):861-874.

31. Ling CX, Huang J, Zhang H. AUC: a statistically consistent and more discriminating measure than accuracy. *Proc. 18th Intern. J. Conf. Artif. Intell. (IJCAI);* 2003. p.329-341.

32. Bouckaert RR. Estimating replicability of classifier learning experiments. *Pattern Recognit. Lett.* ACM,New York, NY, USA; 2004. p.15.

33. Asuncion A, Newman DJ. UCI machine learning repository. *University of California, Irvine, School of Information and Computer Sciences.* http://www.ics.uci.edu/~mlearn/MLRepository.html; 2007.

34. Mitchell T. Computational methods of feature selection. McGraw-Hill; 1997.

35. Fawcett T. An introduction to ROC analysis. *Pattern Recognit. Lett.* 2006;27(8):861-874.