Incremental Forward Feature Selection with Application to Microarray Gene Expression Data

Yuh-Jye Lee[*], Chien-Chung Chang[†], and Chia-Huang Chao[‡]

Department of Computer Science and Information Engineering

National Taiwan University of Science and Technology

Taipei, 106 Taiwan

March 2, 2008

Abstract

In this paper, we proposed a new feature selection scheme, the incremental forward feature selection, which is inspired by incremental reduced support vector machines. In our method, a new feature will be added into the current selected feature subset if it will bring in the most extra information. We measure this information by using the distance between the new feature vector and the column space spanned by current feature subset. The incremental forward feature selection scheme can exclude highly linear correlated features which provide redundant information and might

[*] corresponding author, yuh-jye@mail.ntust.edu.tw

[†] D9115009@mail.ntust.edu.tw

[‡] M9115016@mail.ntust.edu.tw

degrade the efficiency of learning algorithms. We compared our method with the weight score approach and the 1-norm support vector machine on two well-known microarray gene expression data sets, the acute leukemia and colon cancer data sets. These two data sets have a very few observations but huge number of genes. We applied the linear smooth support vector machine to the feature subsets selected by these three schemes respectively and obtained a slightly better classification results in the 1-norm support vector machine and incremental forward feature selection. Finally, we claimed that the rest of genes still contain some useful information. We iteratively removed the previous selected features from the data sets and repeated the feature selection and classification steps for four rounds. The results show that there are many distinct feature subsets that can provide enough information for classification tasks in these two microarray gene expression data sets.

Keywords: 1-norm support vector machine, filter model, incremental forward feature selection, weight score, wrapper model.

1 Introduction

In many data mining applications such as genome projects, text categorization, and image retrieval (Yu & Liu, 2003), the data sets consist of large number of features as well as a relative small number of samples. There are many irrelevant and redundant features in observations of these data sets. For dealing with this kind of data sets, the learning algorithms may generate a more complex model and cause the overfitting problem (Kohavi & John, 1997). The performance of learning algorithms is degraded due to the number of features grows (Kohavi & John, 1997; Yu & Liu, 2003). We call the phenomenon as the curse of dimensionality (Cristianini & Shawe-Taylor, 2000).

The feature selection problem focuses on selecting a subset of original features, such that a learning algorithm can generate a classifier via these selected features without sacrificing the prediction accuracy. Through feature selection, the noise and redundant features can be discarded according to some evaluation criteria. Feature selection approaches can be grouped into two categories, filter model and wrapper model (Kohavi & John, 1997). The filter model evaluates the significance of features and then filters out uninformative features. Feature selection is a stand-alone step before the learning algorithm is performed. It totally ignores the effects of the selected features on the performance of the learning algorithm used. Hence it can be treated as a preprocessing step. The identical feature subset will be selected no matter what learning algorithms are used in the learning tasks. By contrast, the wrapper model tries to choose the best feature subset to use by taking the learning algorithm as part of the evaluation function (Kohavi & John, 1997). It considers the correlations between features used and the underlying learning algorithm while performing feature selection tasks. When applying the wrapper model into the learning tasks, it not only takes the "goodness of fit" into account but also penalizes on the number of selected features. Thus, the wrapper model will select different (suitable) feature subsets for different algorithms.

In this paper, we presented a new feature selection method, incremental forward feature selection (IFFS) which is inspired by incremental reduced SVM (IRSVM) (Lee, Lo, & Huang, 2003). We compared the IFFS with the weight score approach which is proposed in (Golub et al., 1999) as well as 1-norm support vector machine (SVM) (Fung & Mangasarian, 2003; Zhu, Rosset, Hastie, & Tibshirani, 2003; Lee, Mangasarian, & Wolberg, 2003). The weight score approach, a filter model, evaluates the variation of feature values in different classes and can produce a simple ranking criterion. The 1-norm SVM and IFFS, belong to the wrapper model, select features under the classification mechanism by taking the linear classification model into account and select more suitable feature subsets for the learning algorithm. They can select smaller feature subsets than the weight score approach does and still have equal or even better classification accuracy. We also test these three feature selection methods on two microarray gene expression data sets, the acute leukemia data set (Golub et al., 1999) as well as colon cancer data set (Alon et al., 1999). For comparing different feature selection methods, we use the same classification method, linear smooth support vector machine (SSVM) (Lee & Mangasarian, 2001), in all our experiments.

A word about our notations is given below. All vectors will be column vectors unless otherwise specified or transposed to a row vector by a prime superscript $'$. The inner product of two vectors $x, z \in R^n$ will be denoted by $x'z$ and the p-norm of x will be denoted by $\|x\|_p$. A column vector of ones of arbitrary dimension will be denoted by $\mathbf{1}$. The base of the natural logarithm will be denoted by e.

This paper is organized as follows. Section 2 gives an overview of filter model for feature selection. In section 3, we describe the wrapper model for feature selection. The experiments and numerical results for different feature selection approaches are stated in section 4. Section 5 concludes the paper.

2 Filter Model for Feature Selection

The filter model selects the informative features by ranking them according to a criterion function. There are many criterion functions for scoring the importance of features, such as mutual information (MI) (Joachims, 2002), chi-squared test (χ^2-test) (Joachims, 2002), Between-Within ratio (BW) (Dudoit, Fridlyand, & Speed, 2002), Fisher criterion score (Bishop, 1995; Duda & Hart, 1973), and weight score (Furey et al., 2000; Golub et al., 1999; Mukherjee et al., 1998; Slonim, Tamayo, Mesirov, Golub, & Lander, 2000) approaches. These criterion functions are designed to have large values if either the correlation between the feature and class label is stronger or the differences between different classes in the feature is bigger. These values are computed independently of the learning algorithms. Once we have the values, we can filter out uninformative features and then using the selected features in the learning algorithms. Thus the filter model for feature selection can be treated as data preprocessing. We will only consider the weight score method which is used in (Furey et al., 2000; Golub et al., 1999; Mukherjee et al., 1998; Slonim et al., 2000) for microarray gene expression data analysis as follows.

In choosing the informative features (genes), this approach looks for two properties (Slonim et al., 2000):

- The expression values of an informative gene in one class should be quite different from its expression values in the other.
- In addition to the differences in expression values that are demonstrated by the class distinction, the expression values of an informative gene should be as little variation as possible in the same class.

In order to realize these two properties the weight score of j^{th} feature is defined as the ratio between the difference of the means of expression levels and the sum of standard deviations in two classes:

$$w_j = \frac{|\mu_j^+ - \mu_j^-|}{\sigma_j^+ + \sigma_j^-}, \tag{1}$$

where μ_j and σ_j are the mean and standard deviation of j^{th} feature for training samples of class A_+ (superscripted by $+$) or class A_- (superscripted by $-$).

Based on the definition of weight score, the j^{th} gene conforms to be an informative gene, the value of w_j should be greater than genes that do not conform. An informative gene will have a bigger $|\mu_j^+ - \mu_j^-|$ which measures the difference between different classes and a smaller $(\sigma_j^+ + \sigma_j^-)$. This criterion function is similar to the Between-Within ratio (BW) and Fisher criterion score approaches which select the informative genes with the largest ratios of between-groups to within-groups sum of squares. We illustrate the weight score idea by a simple example which consists of three genes expression values from the colon cancer data set. Twenty samples are randomly selected, samples 1-10 belong to Tumor class (class 1) and samples 11-20 belong to Normal (class 2). We summarize this example in Figure 1 and Figure 2. It is obvious that Gene 3 has the highest weight score value. The significance ranking of these three genes in decreasing order will be Gene 3, Gene 2 and Gene 1.

After ranking features, users can set a threshold to select the informative features whose weight scores exceed the threshold. We know that the weight scores were computed independently without taking the interdependence of features into account. Two highly linear correlated features might be selected together. For example, if a data set contains two features, the diameter and radius of cell respectively. These two features will have the same weight score because the

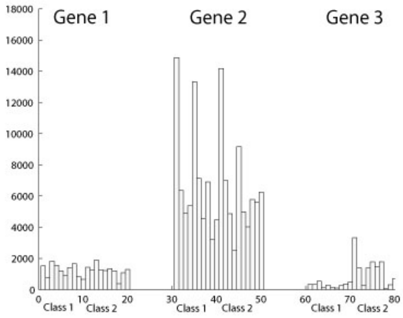

Figure 1: **There are three genes expression data, each gene contains 20 samples which are randomly selected from the colon cancer data set samples 1-10 belong to Tumor (class 1) and 11-20 belong to Normal (class 2).**

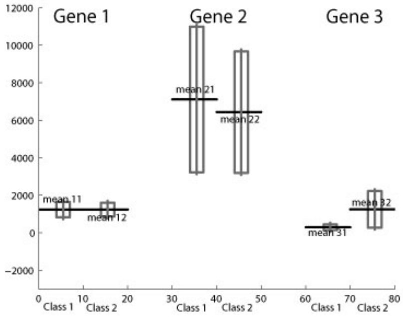

Figure 2: **For each gene and class, the dark horizontal lines illustrate within-class means of gene expression values (i.e., μ_j^+ and μ_j^-, $j \in \{1,2,3\}$)and each box shows the region of one standard deviation (i.e., $\mu_j^+ \pm \sigma_j^+$ and $\mu_j^- \pm \sigma_j^-, j \in \{1,2,3\}$).**

diameter is double of the radius of cell. Hence the two features will be chosen or not at the same time. In some cases, the highly linear correlated features will degrade the performance of classification algorithms. For instance, the classification accuracy of a Naïve Bayesian classifier is hurt more by conditional dependence between features than the presence of irrelevant features (Kohavi & John, 1998; Langley & Sage, 1994). Besides, if a linear classifier such as SVM classifier is used for classification problems, highly linear correlated features will provide redundant information and may degrade the efficiency of learning algorithms.

3 Wrapper Model for Feature Selection

The wrapper model selects feature subsets using the learning algorithm itself as part of the evaluation function (Kohavi & John, 1997). In this section, we introduce two feature selection methods, the 1-norm SVM and IFFS. These two methods take the linear classification model into account while performing feature selection. Hence they may exclude highly linear related features.

3.1 1-norm SVM for Feature Selection

Consider the problem of classifying points into two classes, A_- and A_+. We are given a training data set $\{(x^i, y_i)\}_{i=1}^m$, where $x^i \in \mathcal{X} \subset R^n$ is an input vector and $y_i \in \{-1, 1\}$ is a class label, indicating one of the two classes, A_- and A_+, to which the input point belongs. We represent these data points by an $m \times n$ matrix A, where the i^{th} row A_i corresponds to the i^{th} input data point and the j^{th} column A^j represents the j^{th} feature expression vector. We use an $m \times m$ diagonal matrix $D, D_{ii} = y_i$ to specify the membership of each input point. The main goal of the classification problem is to find a classifier that can predict correctly the unseen class labels for new data inputs. It can be achieved by constructing a linear or nonlinear separating surface, $f(x) = 0$. We classify a test point x to A_+ if $f(x) \geq 0$, otherwise, to A_-. For finding the optimal linear separating hyperplane in the form $f(x) = w'x + b$, the standard SVM solves the following problem for some $C > 0$ (Vapnik, 1995; Cortes & Vapnik, 1995):

$$\begin{array}{lrl} \displaystyle\min_{(w,b,\xi)\in R^{n+1+m}} & C\mathbf{1}'\xi + \frac{1}{2}\|w\|_2^2 & \\ s.t. & D(Aw + \mathbf{1}b) + \xi & \geq \mathbf{1}, \\ & \xi & \geq 0, \end{array} \tag{2}$$

where the components of vector $\xi \in R^m$ in problem (2) are slack variables.

If we replace the regularization term $\frac{1}{2}\|w\|_2^2$ in the objective function (2) by $\|w\|_1$, we can get the 1-norm SVM formulation as follows:

$$\begin{array}{lrl} \displaystyle\min_{(w,b,\xi)\in R^{n+1+m}} & C\mathbf{1}'\xi + \|w\|_1 & \\ s.t. & D(Aw + \mathbf{1}b) + \xi & \geq \mathbf{1}, \\ & \xi & \geq 0. \end{array} \tag{3}$$

The objective function in (3) is a piecewise linear function and can be reformulated as a linear programming problem:

$$\begin{array}{lrl} \displaystyle\min_{(w,s,b,\xi)\in R^{2n+1+m}} & C\mathbf{1}'\xi + \mathbf{1}'s & \\ s.t. & D(Aw + \mathbf{1}b) + \xi & \geq \mathbf{1}, \\ & -s \leq w & \leq s, \\ & \xi & \geq 0, \end{array} \tag{4}$$

where s is the upper bound of the absolute value of w componentwise and the sum of elements of s, $\mathbf{1}'s$, is equal to $\|w\|_1$ at the optimal solution of (4). We note that the formulation of 1-norm SVM is equivalent to least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996; Osborne, Presnell, & Turlach, 2000). The 1-norm SVM can generate a very sparse solution w, thus it is a very useful tool for feature selection. The solution sparsity implies that the separating hyperplane $f(x) = w'x + b$ depends on very few input features when a linear classifier is used. Hence the 1-norm SVM can significantly suppress the features, especially for redundant noise features or highly linear correlated features. (Lee et al., 2003; Fung & Mangasarian, 2003; Zhu et al., 2003). We describe how to use the 1-norm SVM for feature selection below.

We start with finding the optimal solution (w^*, b^*) of the 1-norm SVM model (3) by all features. Because the solution w^* is very sparse, the 1-norm SVM can discard the j^{th} feature if the corresponding weight w_j^* is very close to 0. By contrast, the larger weight value w_j^* means that the corresponding j^{th} feature is more important for discriminating these two classes data. There are several strategies to remove the unimportant features. For instance, we may start with tuning a small value of criterion parameter $\delta > 0$ such as 10^{-5}. When the corresponding weight value $\left|w_j^*\right| < \delta$, we will remove the j^{th} feature. Then, the rest of features are used to rebuild a classifier. We repeat tuning the criterion δ increasingly until any further elimination will increase the test error dramatically. We also can rank the weight value $|w_j^*|$ descending and retain the top k features. The value k may also be tuned decreasingly until the error is increased.

3.2 Incremental Forward Feature Selection (IFFS)

In this subsection, we proposed a novel feature selection method, the Incremental Forward Feature Selection (IFFS) which is inspired by the key idea of IRSVM (Lee et al., 2003). The intuition behind this method is that a new feature will be added into the current feature subset if it will bring in the most extra information. We measure the extra information for a feature using the distance between this new feature vector and the column space spanned by current feature subset. This distance can be computed by solving a least squares problem. The feature with the farthest distance will be added into the current feature subset. Once the most informative feature is added, the features in the current feature subset are used to rebuild a discriminant model. We repeat this procedure until any further addition does not decrease the test error by the new generated model.

However, it is time-consuming for finding the most informative feature from a large number of features. Hence we may also use another mechanism to add more informative features into the feature subset. We first set a threshold $\delta > 0$ such as 10. Please note that the value of δ should not be too small. When the value r_j, the distance between j^{th} feature vector and the column space spanned by current selected features, is greater than δ, we add the j^{th} feature into the feature subset. We repeat these steps until there is no feature can be added to the current feature subset. Once the procedure is stopped for a certain δ, the features in the current feature subset are used to reconstruct a classifier. We repeat this procedure with setting the threshold δ decreasingly until any further addition does not decrease the test error by the new generated classifier. We describe our IFFS algorithm in Table 1.

The initial selected feature subset can be generated by expert's input if it is available. Otherwise, we will use the feature with the largest weight score defined in (1).

```
Algorithm: Incremental Forward Feature Selection
Input:
  $S_{initial}$  // The Initial index set of selected features
Output:
  $S_{final}$  // The final index set of selected features
  $Model$  // The final discriminant model

(1) Let $S = S_{initial}$.
(2) Set a proper threshold $\delta > 0$.
(3) Let $A^S \in R^{m\times|S|}$ be a submatrix of $A$ with the corresponding
    columns in the selected feature index set $S$, where $|S|$ is the cardinality of $S$.
(4) Choose $j \in \{1, 2, \ldots, n\} \setminus S$ and compute the distance $r_j$ from the
    feature vector $A^j$ to the column space of $A^S$. This distance can be
    computed as $r_j = \|A^S\beta^* - A^j\|_2$, where $\beta^*$ is the optimal solution of
    the least squares problem $\min_{\beta\in R^{|S|}} \|A^S\beta - A^j\|_2^2$.
(5) If the distance $r_j > \delta$, then $S = S \cup \{j\}$. That is, we add the $j^{th}$
    feature into the selected feature index set $S$.
(6) Update the matrix $A^S$.
(7) Repeat Step (4) $\sim$ (6) until there is no feature can be added to the
    current selected feature index set $S$.
(8) Reconstruct a classifier using the current selected feature index set $S$
    and compute the test error by the classifier.
(9) Repeat Step (2) $\sim$ (8) by setting the threshold $\delta$ decreasingly until
    the test error is not decreasing.
(10) Let $S_{final} = S$ and the $Model$ be the final classifier which is constructed by Step (8).
(11) Return $S_{final}$ and $Model$.
```

Table 1: **The IFFS algorithm**

4 Experimental Setting and Numerical Results

All our experiments were performed on a personal computer, which utilizes a 2.0 GHz Intel Pentium IV processor and 512 megabytes of RAM. This computer runs on Windows 2000 professional operating system, with MATLAB 6.0 installed. We tested our IFFS algorithm and compared with the weight score approach as well as the 1-norm SVM on two well-known microarray gene expression data sets, the acute leukemia and colon cancer data sets, respectively (Golub et al., 1999; Alon et al., 1999).

4.1 Acute Leukemia Data Set

In the acute leukemia data set (Golub et al., 1999), there are 25 acute myeloid leukemia (AML) and 47 acute lymphoblastic leukemia (ALL) samples which are taken from 72 patients. Each sample has 7,129 genes obtained from microarray experiments. In ALL class, the 47 samples are further grouped into 38 B-lineage cell ALL (B-Cell ALL) and 9 T-lineage cell ALL (T-Cell ALL) samples. The acute leukemia data set contains a training set and an independent test set. The training set has 38 samples which includes 11 AML and 27 ALL samples (19 B-Cell, 8 T-Cell). There are 34 samples in the test set which consists of 14 AML and 20 ALL samples (19 B-Cell, 1 T-Cell). The summary of this data set is shown in Table 2.

Since the acute leukemia data set contains three categories, we convert this trinary classification problem into two binary classification problems in our experiments. Figure 3 shows the classification flowchart. In Figure 3, the two diamonds represent two classifiers. One of them is used to distinguish AML from ALL and another is used to classify B-Cell ALL and T-Cell ALL. The two ellipses stand for two gene selection steps. They can be performed by any feature selection method. Because the first gene subset is just chosen for discriminating AML from ALL and both T-cell and B-cell are the subclasses of ALL, intuitively T-cell and B-cell should have very closed gene expression levels in this selected gene subset. We cannot use this selected gene subset to distinguish T-cell from B-cell well. Thus we need the second gene selection step to choose a new gene

Acute Leukemia Data Set (72 × 7129)			
	Training	Test	Total
AML	11	14	25
B-Cell ALL	19	19	38
T-Cell ALL	8	1	9
Total	38	34	72
# of genes	7129		

Table 2: **Summary of the acute leukemia microarray gene expression data set**

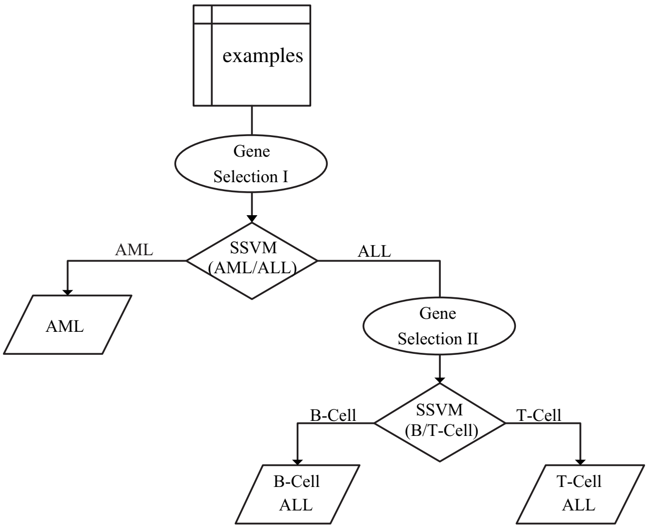

Figure 3: **Classification flowchart for the acute leukemia data set**

subset to discriminate them.

4.2 Colon Cancer Data Set

Microarray gene expression values for 22 normal and 40 colon cancer tissues are collected. Each sample has 6,500 genes which are obtained from microarray experiments. The colon cancer data set (Alon et al., 1999) collected 2,000 genes with the highest minimal intensity across the 62 tissues.

In (Weston, Elisseeff, Schölkopf, & Tipping, 2003; Weston et al., 2001), the

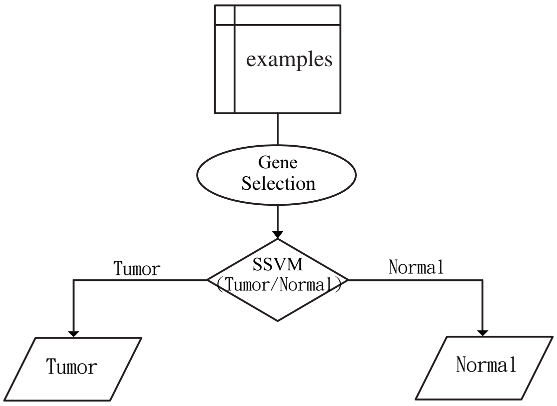

Figure 4: **Classification flowchart for the colon cancer data set**

authors divided the data set into a training set of 50 samples and a test set of 12 samples in 500 bootstrap trials. We followed this scale of test set and performed stratified five-fold cross-validation in our experiments. For each fold, we randomly selected 8 tumor as well as 4 normal samples for a test set and the rest part for a training set. The colon cancer data set contains only tumor and normal samples. Hence it is a binary classification problem. We do just need one feature selection step and use the selected genes to construct suitable classifiers. The classification flowchart for the colon cancer data set is illustrated in Figure 4.

4.3 Numerical Results and Comparisons

We first concentrated on comparing the performance of our IFFS method with other feature selection approaches. For each data set, we applied three different feature selection schemes to select the most informative genes based on the entire data set. Then we use the selected feature subsets to construct the linear SSVM classifiers (Lee & Mangasarian, 2001). We evaluated the performance

<table>
<tr><th colspan="5">Acute Leukemia Data Set (72 × 7129)
(Tested by independent test samples)</th></tr>
<tr><th>Method</th><th colspan="2">ALL/AML</th><th colspan="2">B-Cell/T-Cell</th></tr>
<tr><th></th><th># of Genes</th><th>Errors</th><th># of Genes</th><th>Errors</th></tr>
<tr><td>Golub</td><td>50</td><td>2</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Weston (2001)</td><td>20</td><td>0</td><td>5</td><td>0</td></tr>
<tr><td>Guyon</td><td>8</td><td>0</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Zhu</td><td>17</td><td>2</td><td>N/A</td><td>N/A</td></tr>
<tr><td>Weight Score Approach</td><td>10</td><td>1</td><td>20</td><td>0</td></tr>
<tr><td>1-norm SVM</td><td>6</td><td>0</td><td>8</td><td>0</td></tr>
<tr><td>IFFS</td><td>14</td><td>0</td><td>9</td><td>0</td></tr>
</table>

Golub: (Golub et al., 1999)
Weston (2001): (Weston et al., 2001)
Guyon: (Guyon, Watson, Barnhill, & Vapnik, 2002)
Zhu: (Zhu et al., 2003)
N/A: Denote *not available* results

Table 3: **Numerical results of the acute leukemia data set**

of different feature selection schemes by counting the numbers of misclassified samples of the linear SSVM classifiers which generated by different selected feature subsets.

In the acute leukemia data set, we reported the numbers of the selected genes and the misclassified samples for different feature selection schemes in Table 3 and included previous results done by (Golub et al., 1999; Weston et al., 2001; Guyon et al., 2002; Zhu et al., 2003) as well for comparison purpose. For discriminating AML samples from ALL samples, the 1-norm SVM and IFFS selected 6 genes and 14 genes, respectively, and the SSVM classifiers generated by these two selected gene subsets can separate AML and ALL samples well in the independent test set. However, the weight score approach select 10 genes and has 1 misclassified sample. For distinguishing T-Cell ALL samples from B-Cell ALL samples, the weight score approach, 1-norm SVM, and IFFS selected 20 genes, 6 genes, and 14 genes, respectively. No matter which feature selection scheme is used for constructing the SSVM classifiers, there is no misclassified

Colon Cancer Data Set (62 × 2000)		
(Tested by stratified five-fold cross-validation)		
Method	Tumor/Normal	
	# of Genes	Errors
Weston (2001)	15	1.5
Guyon	8	3
Weston (2003)	20	1.7
Weight Score Approach	20	1.6
1-norm SVM	8	1.4
IFFS	5	1.4

Weston (2001): (Weston et al., 2001)
Guyon: (Guyon et al., 2002)
Weston (2003): (Weston et al., 2003)

Table 4: **Numerical results of the colon cancer data set**

sample occurred in the independent test set.

In the colon cancer data set, the 1-norm SVM and IFFS selected 8 genes and 5 genes, respectively, and the SSVM classifiers generated by these two selected gene subsets have 1.4 misclassified samples in stratified five-fold cross-validation test sets averages. However, the weight score approach select 20 genes and has 1.6 misclassified samples. We summarized these results in Table 4 and included previous results done by (Weston et al., 2001; Guyon et al., 2002; Weston et al., 2003) as well for comparison purpose.

Moreover, we want to explore the correlation coefficients between each pair of selected genes. Table 5 showed the statistic results of the correlation coefficients between each pair of selected genes via three different feature selection methods where the Max_Corr, Min_Corr, and Avg_Corr represented the maximum, minimum, and average values of correlation coefficients, respectively. The numerical results demonstrate that the values under the IFFS and 1-norm SVM are obviously smaller than values under the weight score approach in all class pairs. It indicates that genes selected by the IFFS and 1-norm SVM are mutually less

Gene Selection Methods		Leukemia Data ALL/AML	Leukemia Data B/T-Cell ALL	Colon Data Tumor/Normal
Weight Score Approach	Min_Corr	0.512	0.583	0.143
	Max_Corr	0.713	0.799	0.695
	Avg_Corr	0.637	0.688	0.550
1-norm SVM	Min_Corr	0.236	0.264	0.284
	Max_Corr	0.394	0.608	0.458
	Avg_Corr	0.343	0.497	0.371
IFFS	Min_Corr	0.416	0.273	0.168
	Max_Corr	0.537	0.602	0.304
	Avg_Corr	0.487	0.451	0.223

Table 5: **Summary of correlation coefficients between selected genes**

correlated than those selected by the weight score approach. Therefore, IFFS and 1-norm SVM gene selection methods may avoid selecting highly linear correlated genes which provide redundant information and degrade the efficiency of learning algorithms.

Finally, we argue that the rest of genes still contain some useful information for discriminant purpose. In order to support this argument, we remove the selected genes from the data set and perform the previous experiments repeatedly until the classification ability of SSVM classifiers degrade drastically. Table 6 illustrated the results of performing the above procedure for four rounds. The results showed that we can remove about 100 genes from the acute leukemia data set and about 50 genes from the colon cancer data set without sacrificing the classification ability. It means that there are more than one gene subset can be used to construct an accurate SSVM classifier in each data set no matter which feature selection approach is used. These observations evidence that there are highly related genes in a microarray gene expression data set and some of them can be replaced by others. In general, we need select more genes for constructing a classifier with similar accuracy when the previous selected genes are

Genes Are Selected by Weight Score Approach		Leukemia Data		Colon Data
		ALL/AML	B/T-Cell ALL	Tumor/Normal
Round 1	# of Genes	10	20	20
	Errors	1	0	1.6
Round 2	# of Genes	20	20	20
	Errors	1	0	1.4
Round 3	# of Genes	30	20	30
	Errors	1	0	2.4
Round 4	# of Genes	40	30	40
	Errors	1	0	3.2

Genes Are Selected by 1-norm SVM		Leukemia Data		Colon Data
		ALL/AML	B/T-Cell ALL	Tumor/Normal
Round 1	# of Genes	6	8	8
	Errors	0	0	1.4
Round 2	# of Genes	11	9	19
	Errors	0	0	1.6
Round 3	# of Genes	15	8	24
	Errors	0	0	1.6
Round 4	# of Genes	18	8	25
	Errors	1	0	2

Genes Are Selected by IFFS		Leukemia Data		Colon Data
		ALL/AML	B/T-Cell ALL	Tumor/Normal
Round 1	# of Genes	14	9	5
	Errors	0	0	1.4
Round 2	# of Genes	14	12	6
	Errors	1	1	1.4
Round 3	# of Genes	11	6	5
	Errors	0	0	1.6
Round 4	# of Genes	11	8	5
	Errors	0	0	2

Round 1: Select genes from the original data set
Round 2: Select genes from the remaining genes of Round 1
Round 3: Select genes from the remaining genes of Round 2
Round 4: Select genes from the remaining genes of Round 3

Table 6: **Number of errors made by SSVM when selected genes are iteratively removed**

removed continuously. However, for each round, the number of genes selected by IFFS does not increase dramatically. Besides, the IFFS and 1-norm SVM still select fewer genes than weight score approach for generating an accurate SSVM classifier at each round. These numerical outcomes further demonstrated that the IFFS and 1-norm SVM may exclude highly linear correlated genes.

5 Conclusions

In this paper, we presented a new feature selection scheme, IFFS, which belongs to the forward feature selection wrapper model. A new feature will be added into the current feature subset if it will bring in the most extra information which is measured by the distance between this new feature vector and the column space spanned by current feature subset. We tested the IFFS and compared with the weight score approach as well as 1-norm SVM on two famous microarray gene expression data sets, the acute leukemia and colon cancer data sets. Our experimental results showed that the IFFS outperforms the weight score approach and has the same performances as the 1-norm SVM in the numbers of the selected genes and the misclassified samples. The IFFS can exclude highly linear correlated features. The correlation coefficients between each pair of selected genes support this characteristic. Finally, we argue that the rest of genes still contain some useful information. The numerical results demonstrated that we can remove the previous selected genes from the data sets and select another set of genes to construct a new SSVM classifier without sacrificing the prediction ability. The IFFS scheme is a very useful tool for feature selection in particular; there are so many redundant and highly linear correlated features.

References

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., & Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Cell Biology*, *96*, 6745-6750.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* New York: Oxford University Press.

Cortes, C., & Vapnik, V. (1995). Support vector networks. *Machine Learning, 20*, 273-279.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines.* Cambridge: Cambridge University Press.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis.* New York: John Wiley & Sons.

Dudoit, S., Fridlyand, J., & Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of The American Statistical Association*, *97*(457), 77-87.

Fung, G., & Mangasarian, O. L. (2003). A feature selection Newton method for support vector machine classification. *Computational optimization and applications*, 1-18.

Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M., & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, *16.*

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., & Lander, E. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, *286*(5439).

Guyon, I., Watson, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1/2/3), 191-202.

Joachims, T. (2002). *Learning to classify text using support vector machines.* Norwell, Massachusetts: Kluwer Academic Publishers.

Kohavi, R., & John, G. (1997). Wrapper for feature subset selection. *Artificial Intelligent Journal*, 273-324.

Kohavi, R., & John, G. (1998). The wrapper approach. In H. Liu & H. Motoda (Eds.), *Feature selection for knowledge discovery and data mining* (p. 33-50). Dordrecht, The Netherlands: Kluwer Academic Publishers.

Langley, P., & Sage, S. (1994). Induction of selective Bayesian classifiers. In *Proceedings of the tenth conference on uncertainty in artificial intelligence.* Seattle.

Lee, Y.-J., Lo, H.-Y., & Huang, S.-Y. (2003). Incremental reduced support vector machine. In *Proceedings of the 2003 international conference on informatics, cybernetics, and systems (icics 2003).* Kaohsiung, Taiwan.

Lee, Y.-J., & Mangasarian, O. L. (2001). SSVM: A smooth support vector machine. *Computational Optimization and Applications*, *20*, 5-22. (Data Mining Institute, University of Wisconsin, Technical Report 99-03. ftp://ftp.cs.wisc.edu/pub/dmi/tech-reports/99-03.ps)

Lee, Y.-J., Mangasarian, O. L., & Wolberg, W. H. (2003). Survival-time classification of breast cancer patients. *Computational Optimization and Applications*, *25*, 151-166.

Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J., & Poggio, T. (1998). *Support vector machine classification of microarray data* (Tech. Rep. No. AI Memo/CBCL Paper #1677/#182). MIT AI Lab and CBCL.

Osborne, M. R., Presnell, B., & Turlach, B. A. (2000). On the lasso and its dual. *Journal of Computational and Graphical Statistics*, *9*(2), 319-337.

Slonim, D., Tamayo, P., Mesirov, J., Golub, T., & Lander, E. (2000). Class prediction and discovery using gene expression data. In *Fourth annual international conference on computational molecular biology* (p. 263-272).

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267-288.

Vapnik, V. N. (1995). *The nature of statistical learning theory.* New York: Springer.

Weston, J., Elisseeff, A., Schölkopf, B., & Tipping, M. (2003). Use of the zero-norm with linear models and kernel methods. *Jorunal of Machine Learning Research*, *3*, 1439-1461.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2001). Feature selection for SVMs. In *Advances in neural information processing systems 13* (p. 668-674).

Yu, L., & Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the twentieth international conference on machine learning (icml).* Washington DC.

Zhu, J., Rosset, S., Hastie, T., & Tibshirani, R. (2003). 1-norm support vector machines. In *Advances in neural information processing systems 07.*