Computers in Biology and Medicine 41 (2011) 435–441

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/cbm

Methods of forward feature selection based on the aggregation of classifiers generated by single attribute

Linkai Luo*, Lingjun Ye, Meixiang Luo, Dengfeng Huang, Hong Peng, Fan Yang

Department of Automation, Xiamen University, Xiamen 361005, PR China

ARTICLE INFO

Article history:
Received 24 March 2010
Accepted 15 April 2011

Keywords:
Forward feature selection
Gene expression data analysis
Aggregation of classifiers
SNR
SVM-RFE

ABSTRACT

Compared to backward feature selection (BFS) method in gene expression data analysis, forward feature selection (FFS) method can obtain an expected feature subset with less iteration. However, the number of FFS method is considerably less than that of BFS method. More efficient FFS methods need to be developed. In this paper, two FFS methods based on the pruning of the classifier ensembles generated by single attribute are proposed for gene selection. The main contributions are as follows: (1) a new loss function, *p*-insensitive loss function, is proposed to overcome the disadvantage of the margin Euclidean distance loss function in the pruning of classifier ensembles; (2) two FFS methods based on the margin Euclidean distance loss function and the *p*-insensitive loss function, named as FFS-ACSA1 and FFS-ACSA2 respectively, are proposed; (3) the comparison experiments on four gene expression datasets show that FFS-ACSA2 obtains the best results among three FFS methods (i.e. signal-to-noise ratio (SNR), FFS-ACSA1 and FFS-ACSA2), and is competitive to the famous support vector machine-based recursive feature elimination (SVM-RFE), while FFS-ACSA1 is unstable.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

The microarray data pose a great challenge on conventional data analysis with a large number of genes and a relatively small number of samples. A lot of studies in microarray data analysis reveal that gene selection or feature selection is more significant than the classification algorithm [1–7]. The simple algorithm with a proper feature selection may achieve the same even better performance than the complex algorithm. In addition, feature selection can also improve the computational efficiency of the classification algorithm. So feature selection becomes the key issue of microarray data analysis.

According to the direction of feature selection, the existing methods can be mainly divided into two types: backward and forward. In each iteration step, the backward feature selection (BFS) method eliminates the least important features while the forward feature selection (FFS) method selects the most important ones. The BFS method is widely used in feature selection of gene expression data. The famous support vector machine-based recursive feature elimination (SVM-RFE) proposed by Guyon et al. [2] is a representative of BFS methods. Many scholars [5–8] carried out a number of improvements and extensions for SVM-RFE. In contrast, there are only a few FFS methods, such as signal-to-noise ratio (SNR) method proposed by Golub et al. [1], SVM-IRFS proposed by Zhou et al. [9] and the incremental forward feature selection proposed by Lee et al. [10].

An advantage of FFS method is that it may obtain a desired feature subset via less iteration while BFS method usually needs to iterate quite a few times. This advantage will be more outstanding if the computational efficiency and/or the small size of desired feature subset are more concerned. However, FFS method tends to miss complementary features or select redundant features so that it does not often separate the data well. For example, the famous SNR method can quickly sort the feature importance while the selected features are always redundant and miss the complementary genes so that it is not as good as SVM-RFE [2]. Therefore, a key issue in FFS method is how to select complementary features and remove redundant features.

Martínez-Muñoz and Suáárez [11] proposed a pruning method (MSPM) in bagging classifier ensembles. MSPM belongs to forward method. At each iteration step, it selects a classifier to the desired aggregation, where the classifier added is the best one so that some optimal objective function is obtained by the desired aggregation. Selecting complementary classifiers and removing redundant ones is the main consideration in the design of objective function. MSPM obtained a lower generalization error with fewer classifiers than original bagging classifier ensembles. Since MSPM is a forward method, it can be applied to the forward feature selection problem. Based on this idea, we propose two FFS methods named as FFS-ACSA1 and FFS-ACSA2. In the two FFS

* Corresponding author.
E-mail address: luolk@xmu.edu.cn (L. Luo).

0010-4825/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiomed.2011.04.005

methods, we first construct the classifiers based on the ratio of signal to noise on single attribute (here a feature corresponds to a classifier), then apply MSPM to select a good combination of classifiers (i.e. a good feature subset). The difference between the two methods is that FFS-ACSA1 uses Euclidean distance loss function while FFS-ACSA2 uses p-insensitive loss function. To investigate the performance of the two FFS methods, we conduct some comparison experiments of FFS-ACSA1, FFS-ACSA2, SNR and SVM-RFE on four gene expression datasets. The experiment results reveal that FFS-ACSA2 improves the classification performance compared with traditional SNR method and obtain a competitive performance with the famous SVM-RFE.

The rest of this paper is organized as follows. Section 2 briefly introduces the pruning method for bagging classifier ensembles. In Section 3, we present the framework of FFS-ACSA1 and FFS-ACSA2. In Section 4, we carry out some experiments on four gene expression datasets to compare the classification performance of the classic SNR, FFS-ACSA1, FFS-ACSA2 and the famous SVM-RFE. Finally, some conclusions and discussions are given in Section 5.

2. The aggregation of classifiers

The goal of the classification algorithm is to generate a classification function with a given training dataset so that it can predict the label of unseen sample well. For the sake of simplicity, we only consider the binary classification problem with a given training dataset

$$Tdata = \{(x_i,y_i)|x_i \in R^n, y_i \in \{1,-1\}, i = 1,2,\ldots,l\}, \tag{1}$$

where l is the number of samples, n is the number of the features(attributes).

The classifier ensembles are often helpful to achieve a better classification performance than a single classifier. Bagging is one of the common schemes to generate classifier ensembles. In bagging, a set of classifiers $H = \{h_t(x) \in \{1,-1\} : t = 1,\ldots,T\}$ is generated by training the bootstrap samples and the final decision is given by

$$f(x) = \text{sign}\left(\sum_{t=1}^{T} h_t(x)\right). \tag{2}$$

A drawback of bagging classifier ensembles is the large amount of memory required to store all the classifiers in the ensemble [11,12]. Another drawback is that the correlation among the individual classifiers is not considered. To overcome these drawbacks, some rules are designed to aggregate classifiers so that a subensemble with fewer classifiers and lower generalization error can be obtained. Among these rules, the margin distance minimization [11] has shown a success in aggregation of classifiers.

Let $c_t = (y_1 h_t(x_1), y_2 h_t(x_2), \ldots, y_l h_t(x_l))^T$ is an l-vector. If $(c_t)_i = 1$, then the tth classifier correctly classifies the ith training sample, otherwise misclassifies it. c_t records the classification result of the classifier $h_t(\bullet)$ on training dataset *Tdata*. The classification result of the classifier ensemble on *Tdata* is defined by the average vector

$$c = \frac{1}{T}\sum_{t=1}^{T} c_t. \tag{3}$$

If the ensemble correctly classifies the ith training sample, c_i is positive, otherwise negative. The component c_i represents the classification margin of the ith training sample. Martínez-Muñoz and Suáárez [11] defined a vector

$$o = (p,p,\ldots,p)^T, \tag{4}$$

as the objective position vector of c, where $0 < p \le 1$. The criterion for adding a classifier to the ensemble is minimizing the distance between c and o, i.e.

$$a_u = \underset{k}{\text{argmin}}\, d\left(o, \frac{1}{T}(c_k + \sum_{t=1}^{u-1} c_{s_t})\right), \tag{5}$$

where $\{c_{s_t} \in H, t = 1,\ldots,u-1\}$ is the ensemble, k runs throughout the classifiers outside the ensemble, $d(\cdot,\cdot)$ is the Euclidean distance, and a_u is the uth classifier added to the ensemble. Martínez-Muñoz and Suáárez suggested that the value of p should be chosen between 0.05 and 0.25. For more details about margin distance minimization, please refer to Martínez-Muñoz and Suáárez [11].

3. Forward feature selection based on the aggregation of classifiers generated by single attribute

In this section, we construct two FFS methods based on the pruning of the classifier ensembles where each classifier is generated by single attribute.

Let us consider the feature selection problem for the binary classification with the given training dataset *Tdata* as (1). The task is to select a feature subset I with good classification performance from the feature set $\{1,2,\ldots,n\}$.

In order to apply MSPM, we first construct some classifiers based on single attribute. Let

$$s_i = \frac{\mu_+^{(i)} - \mu_-^{(i)}}{\sigma_+^{(i)} + \sigma_-^{(i)}}, \tag{6}$$

denote the ratio of signal to noise on attribute i, where $\mu_+^{(i)}, \sigma_+^{(i)}$, $\mu_-^{(i)}, \sigma_-^{(i)}$ denote the mean and standard deviation of all samples on attribute i in class $(+)$ or class $(-)$, respectively. The classifier based on single attribute is constructed as

$$f_i(x) = \text{sign}(s_i(x(i) - b_i)), \tag{7}$$

where $b_i = (\mu_i^+ + \mu_i^-)/2$ and $x(i)$ is the value of the sample x on attribute i [1]. A set of classifiers based on single attribute is obtained by

$$G = \{f_i | i = 1,\ldots,n\}. \tag{8}$$

Consequently, we can consider the feature selection problem as a pruning problem of G.

In MSPM, each classifier is obtained by Bagging and has the same weight. In our method, each classifier is obtained by the ratio of signal to noise on single attribute. As the ratio of signal to noise on each attribute is different, each classifier should have different weight. The natural weights are their ratios of signal to noise. Therefore, we change the rule for adding an attribute j to the selected feature subset I to

$$j = \arg\ \min_{k \notin I}\ d(O, G_k), \tag{9}$$

where $O = (o_1,\ldots,o_l)^T (o_i = p_1, 0 < p_1 \le 1, i = 1,\ldots,l)$ is the objective position vector, $d(\cdot,\cdot)$ is the Euclidean distance between two vectors,

$$G_k = \beta_k F_k + \sum_{i \in I} \beta_i F_i \tag{10}$$

is the classifier ensembles, and

$$\beta_t = \frac{|s_t|}{\sum_{t \in I \cup \{k\}} |s_t|}, \quad F_t = (f_t(x_1),\ldots,f_t(x_l))^T, \quad t \in I \cup \{k\}. \tag{11}$$

The $d(\cdot,\cdot)$ can be regarded as the loss function of G_k. This loss function requires that G_k must be close to the objective position vector O. Even when $(G_k)_i > p_1 > 0$, it still has loss in sample x_i. In fact, $(G_k)_i > p_1 > 0$ indicates that G_k classifies sample x_i correctly with a margin more than p_1. The larger classification margin means the classification is more credible. Consequently, the loss

function $d(\cdot,\cdot)$ exists limitation for it requests reducing the classification margin of the samples that are classified very well. In order to overcome this shortcoming, we define a new loss function

$$Loss(O,G_k)=\sum_{i=1}^{l} Loss_i, \tag{12}$$

where

$$Loss_i=\begin{cases}0, & \text{if } (G_k)_i \geq p_2\\ p_2-(G_k)_i, & \text{otherwise}\end{cases}, \tag{13}$$

$o_i=p_2,\ 0<p_2\leq 1,\ i=1,\ldots,l$. The parameter p_2 denotes the requirement for classification margin. This new loss function is similar to the ε-insensitive loss function in SVM, hence we name it as p-insensitive loss function. It requires the improvement for the samples classified difficultly meanwhile it does not require reducing the classification margin for the samples classified very well (i.e., the samples whose classification margin is larger than p_2). Using p-insensitive loss function, we can get another rule for adding attribute j to the selected feature subset I

$$j=\underset{k\notin I}{\operatorname{argmin}}\ Loss(O,G_k). \tag{14}$$

According to feature selection rule (9) or (14), the forward feature selection algorithms based on the aggregation of classifiers generated by single attribute (FFS-ACSA) are described in Algorithm 1. The algorithm with rule (9) is denoted as FFS-ACSA1, while the algorithm with rule (14) is denoted as FFS-ACSA2.

Algorithm 1. Forward feature selection based on the aggregation of classifiers generated by single attribute (FFS-ACSA)

Inputs: Training examples
$Tdata=\{(x_i,y_i)|x_i\in R^n, y_i\in\{1,-1\}, i=1,2,\ldots,l\}$
Output: Feature ranked list r.
Initialize: Feature ranked list $r=[\,]$.
Constructing the set of classifiers generated by
Single attribute $G=\{f_i|i=1,\cdots,n\}$
According to formulas (6)–(8).
Parameters p_1, p_2.
For $i=1:n$
Choosing feature j from G according to formula (9) or (14)
$G=G\backslash\{f_j\}$
$r=[r,j]$
End

Table 1
Basic information of the datasets.

Dataset	Samples	Genes	Source
Leukemia	72	3571	[13]
DLBCL	77	5469	[14]
Colon	62	2000	[15]
Duke	44	7129	[16]

4. Experiments and discussions

4.1. Datasets

Four public datasets ALL-AML Leukemia (Leukemia), DLBCL, Colon Tumor (Colon) and Duke are used to investigate the performance of SNR, FFS-ACSA1, FFS-ACSA2 and SVM-RFE. Table 1 lists the basic information of these datasets. More details can be found from the source websites and the references therein.

4.2. Experimental program

Considering the excellent performance of SVM-RFE on feature selection for gene expression datasets, we use it as an indicator of SNR, FFS-ACSA1 and FFS-ACSA2. In order to accelerate SVM-RFE, we remove $d\%$ of the features in each iteration step according to the proposal of Ding and Wilkins [8].

For each dataset, 60% of the samples are randomly selected as the training set, and the remaining 40% as the test set. SNR, FFS-ACSA1, FFS-ACSA2 and SVM-RFE are all run on the same training set and test set. The SNR classifier is used to evaluate the performances of three FFS methods while the SVM classifier is used for SVM-RFE. The parameter settings are as follows: for FFS-ACSA1 $p_1=0.2$, for FFS-ACSA2 $p_2=0.2$, for SVM-RFE $C=100$ and $d=1$. All methods are coded in Matlab and Libsvm-mat-2.84-1 [17] is used for SVM. To ensure the credibility of the results, 100 group random experiments are executed.

4.3. Experimental results

Figs. 1–4 show the average test accuracy trends for SNR, FFS-ACSA1, FFS-ACSA2 and SVM-RFE on four datasets, respectively, while Figs. 5–8 are the corresponding standard deviation trends. Table 2 lists the highest average test accuracies. Considering the feature subset with small size is crucial in microarray data analysis, we also report the average test accuracies in Table 3 when the number of genes is 10, 20, 30, 50 and 100 as Niijima and Kuhara [18] had done. A summary of results on each dataset is as follows:

- Leukemia: The performance of FFS-ACSA2, SNR and SVM-RFE are close, while the test accuracy of FFS-ACSA1 is not good and the standard deviation is relatively large.

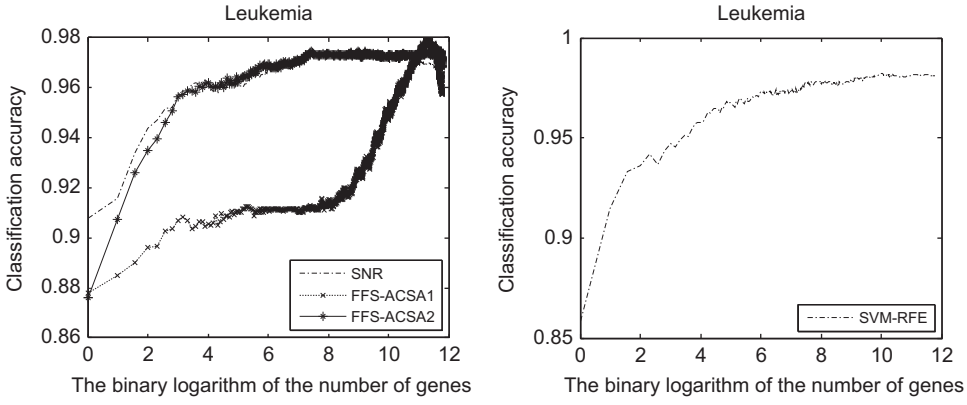

Fig. 1. Average test accuracy trends of 100 group experiments on Leukemia dataset.

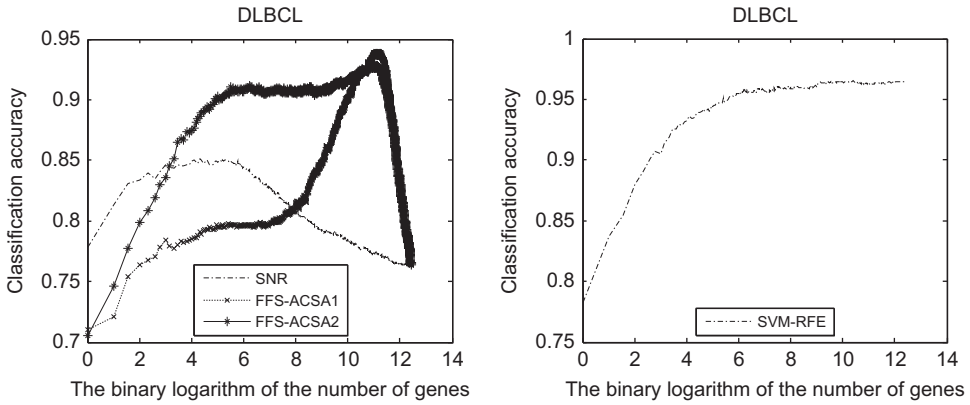

Fig. 2. Average test accuracy trends of 100 group experiments on DLBCL dataset.

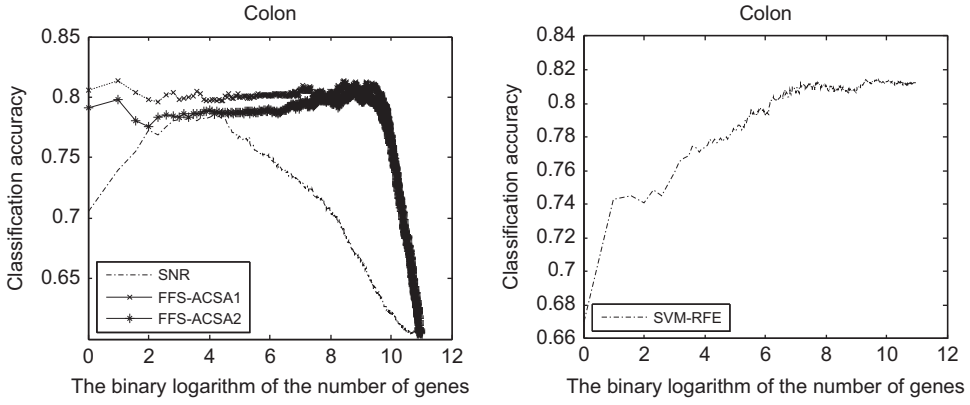

Fig. 3. Average test accuracy trends of 100 group experiments on Colon dataset.

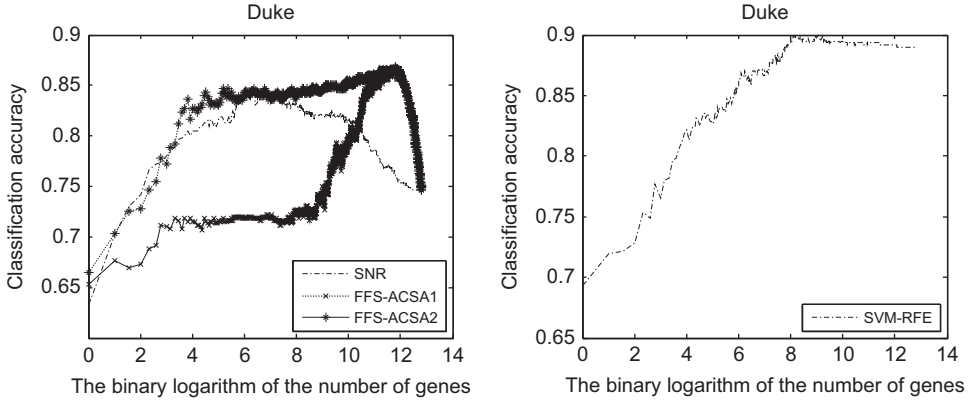

Fig. 4. Average test accuracy trends of 100 group experiments on Duke dataset.

- DLBCL: FFS-ACSA2 achieves the best performance among the three FFS methods, while the performance of FFS-ACSA1 is the worst. FFS-ACSA2 reaches about 90% of the highest test accuracy with about 32 genes while FFS-ACSA1 reaches the close test accuracy with about 1024 genes. SVM-RFE is superior to FFS-ACSA2 on this dataset.
- Colon: FFS-ACSA1 has the best performance among all methods. FFS-ACSA2 is better than SNR. The test accuracies of FFS methods are all superior to SVM-RFE when the number of genes is less than 25. The test accuracy of SNR decreases after the number of genes exceeds 20. The test accuracies of FFS-ACSA1 and FFS-ACSA2 reach 81.40% and 79.80%, respectively, with 2 genes, while they are 74.00% and 74.28% for SNR and SVM-RFE, respectively.
- Duke: FFS-ACSA2 and SVM-RFE obtain close test accuracy when the genes are less than 50. Both of them are a little higher than SNR, while the performance of FFS-ACSA1 is the worst.

Overall speaking, FFS-ACSA2 obtains the best results among three FFS methods and is competitive with SVM-RFE, while FFS-ACSA1 is not stable. FFS-ACSA2 obtains the good performance

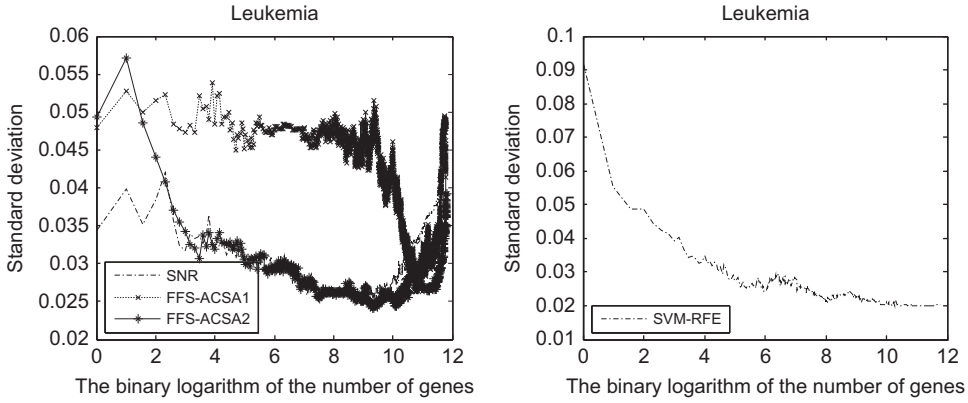

Fig. 5. Standard deviation trends of 100 group experiments on Leukemia dataset.

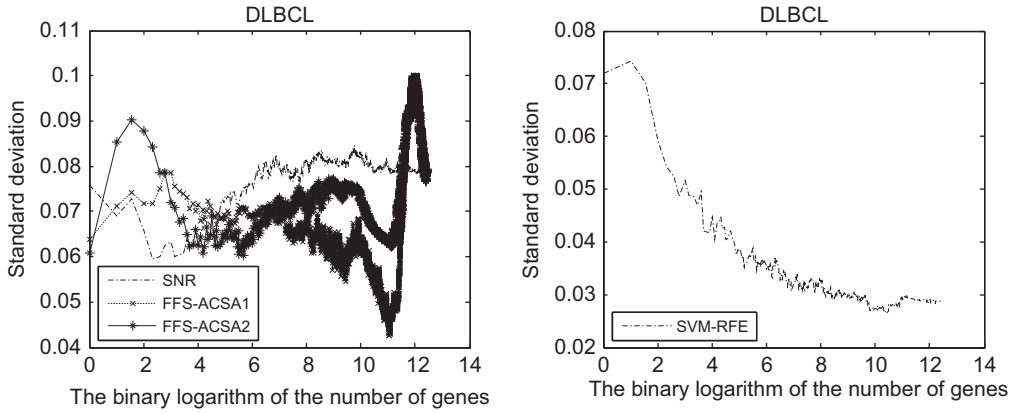

Fig. 6. Standard deviation trends of 100 group experiments on DLBCL dataset.

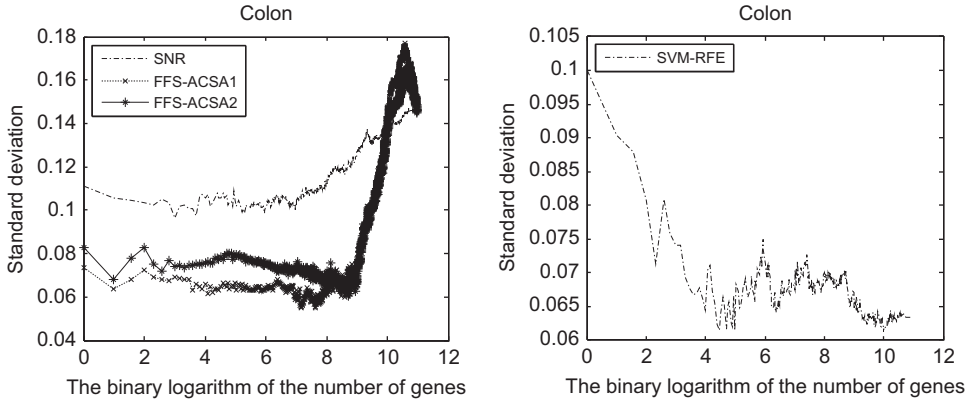

Fig. 7. Standard deviation trends of 100 group experiments on Colon dataset.

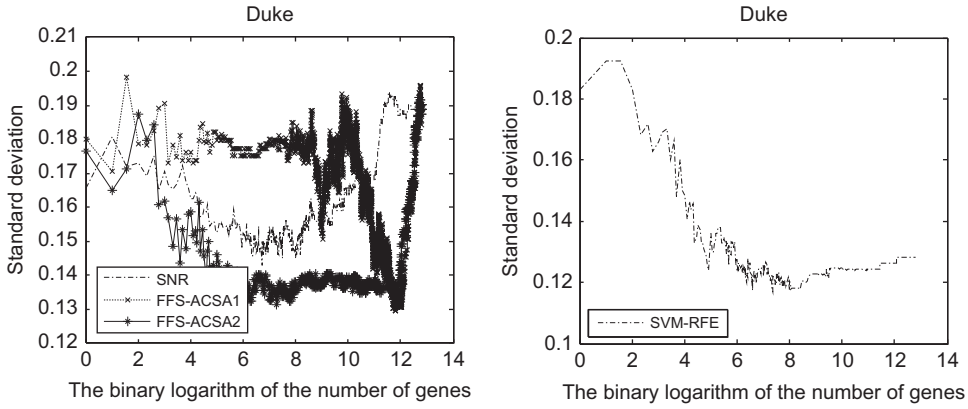

Fig. 8. Standard deviation trends of 100 group experiments on Duke dataset.

with fewer features on four datasets. Compared with SNR, it obtains an improvement on DLBCL, Colon, and Duke. And they are close on Leukemia. Compared with SVM-RFE, it is inferior on DLBCL, while they are close on Leukemia, Colon and Duke. Moreover, FFS-ACSA2 costs less than SVM-RFE in the process of selecting small feature subset. Fig. 9 describes the relation between the iteration steps and the size of selected feature subset for FFS-ACSA2 and SVM-RFE on Leukemia. The average execution time at each iteration step of FFS-ACSA2 is 0.0081s, which is only 23.68% of SVM-RFE (0.0342 s). The main reason is that FFS-ACSA2 only needs to choose a feature from G according to (14) at each iteration step while SVM-RFE needs to train a SVM model, which is relatively time-consuming. Note that FFS-ACSA2 is run in Matlab code while SVM is in Dynamic Link Library [17] on our experiments. FFS-ACSA2 will cost much less time than SVM-RFE does if they were run in the same environment. From the iteration steps and the average execution time at each iteration step, we can see that the smaller the size of selected feature subset is, the more outstanding the advantage of FFS-ACSA2 is. For example, it only needs 1, 10, 20 iteration steps (0.0081, 0.081, 0.162 s), respectively, to obtain a feature subset with 1, 10, 20 features on Leukemia dataset, while SVM-RFE needs 414, 405,395 steps (14.1588, 13.851, 13.509 s). The comparison results about iteration steps and the average cost time at each step on other three datasets are similar.

FFS-ACSA1 only achieves a good performance on Colon, while it is not satisfactory on Leukemia, DLBCL, and Duke. We think the main reason is its loss function $d(O, G_k)$. This loss function requires that G_k must be close to the objective position vector O. Even when $(G_k)_i > p_1 > 0$, it still has loss in sample x_i. It means that FFS-ACSA1 still tends to adjust the samples classified actually very well, which will result in a frequent adjustment and an unstable performance. This is also the reason that we propose p-insensitive loss function.

Table 2
Highest average test accuracy.

Dataset	SNR	FFS-ACSA1	FFS-ACSA2	SVM-RFE
Leukemia	97.41	97.93	97.55	98.21
DLBCL	85.16	94.00	92.81	96.50
Colon	78.48	81.40	80.88	81.44
Duke	84.33	86.83	86.83	90.00

Table 3
Average test accuracy when the number of genes is 10, 20, 30, 50 and 100.

	Number of genes				
	10	20	30	50	100
Leukemia					
SNR	95.86	95.79	96.24	96.41	96.86
FFS-ACSA1	90.69	90.48	91.00	91.07	91.14
FFS-ACSA2	95.86	96.03	96.21	96.66	96.97
SVM-RFE	94.90	96.21	96.72	96.86	97.21
DLBCL					
SNR	84.34	85.16	84.91	84.75	83.25
FFS-ACSA1	77.72	79.06	79.37	79.69	79.81
FFS-ACSA2	85.09	88.62	90.09	90.66	90.72
SVM-RFE	91.87	93.50	94.16	95.16	95.56
Colon					
SNR	78.20	78.48	76.84	75.40	73.52
FFS-ACSA1	80.00	79.80	80.12	80.12	80.12
FFS-ACSA2	78.36	78.72	78.72	78.88	79.24
SVM-RFE	76.76	77.88	78.28	79.24	80.72
Duke					
SNR	79.00	81.00	81.00	82.50	84.17
FFS-ACSA1	71.83	71.00	71.33	72.00	71.83
FFS-ACSA2	79.17	82.83	83.33	83.67	84.00
SVM-RFE	78.17	82.33	82.83	84.50	86.83

5. Conclusions and discussions

In this paper, we first generate the classifiers by the ratio of signal to noise on single attribute. Then we apply the pruning of the aggregation of classifiers to select feature subset. Based on the traditional Euclidean distance loss function and the p-insensitive loss function presented in this paper, we propose two FFS methods: FFS-ACSA1 and FFS-ACSA2. The comparison experiments on four gene expression datasets reveal that FFS-ACSA2 is not only superior to the SNR method, but also achieves a competitive performance with the famous SVM-RFE, while FFS-ACSA1 is not stable.

In future work, we will study the tuning of parameters p_1 and p_2 for different datasets. In addition, other methods of constructing individual classifier, more applications of p-insensitive loss function, and the forward feature selection methods for multi-classification problem all will be investigated.

Conflict of interest statement

None declared.

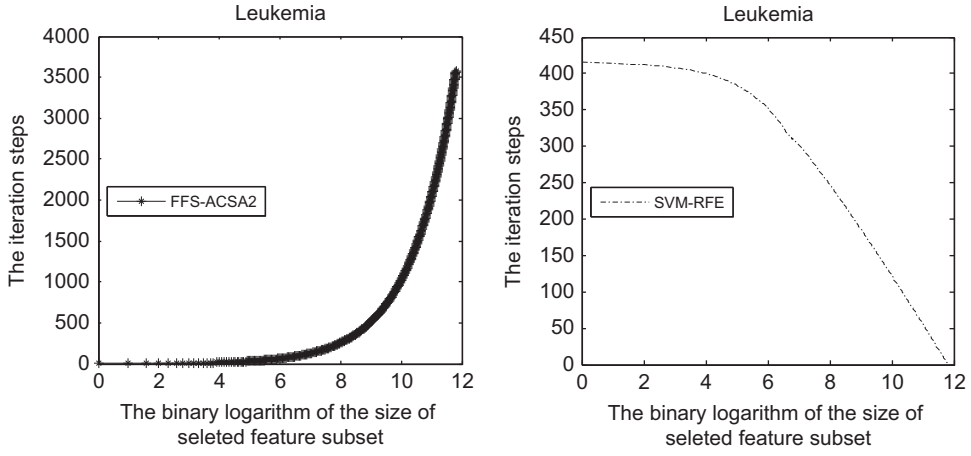

Fig. 9. Relation between the iteration steps and the size of selected feature subset on Leukemia dataset.

Acknowledgments

This work was partially supported by the Fundamental Research Funds of China for the Central Universities under Grant no. 2010121065 and the Natural Science Foundations of Fujian Province of China under Grant no. 2009J05153.

References

[1] T.R. Golub, et al., Molecular classification of cancer:class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.
[2] I. Guyon, et al., Gene selection for cancer classification using support vector machines, Machine Learning 46 (1–3) (2002) 389–422.
[3] M. Yousef, S. Jung, L.C. Showe, M.K. Showe, Recursive cluster elimination (RCE) for classification and feature selection from gene expression data, BMC Bioinformatics 8 (2007) 144.
[4] L.K. Luo, D.F. Huang, L.J. Ye, Q.F. Zhou, G.F. Shao, H. Peng, Improving the computational efficiency of recursive cluster elimination for gene selection, IEEE/ACM Transactions on Computational Biology and Bioinformatics 8 (1) (2011) 122–129.
[5] K.B. Duan, et al., Multiple SVM-RFE for gene selection in cancer classification with expression data, IEEE Transactions on Nanobioscience 4 (3) (2005) 228–234.
[6] X. Zhou, D.P. Tuck., MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data, Bioinformatics 23 (2006) 1106–1114.
[7] S. Deegalla, H. Boström, Classification of microarrays with kNN: comparison of dimensionality reduction methods, in: Proceedings of the Eighth International Conference on Intelligent Data Engineering and Automated Learning, LNCS 4881, Springer-Verlag, 2007, pp. 800–809.
[8] Y. Ding, D. Wilkins, Improving the performance of SVM-RFE to select genes in microarray data, BMC Bioinformatics 7 (Suppl. 2) (2006) S12.
[9] X. Zhou, X.Y. Wu, K.Z. Mao, P. Tuck David, Fast gene selection for microarray data using SVM-based evaluation criterion, in: Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, 2008, pp. 386–389.
[10] Y.J. Lee, C.C. Chang, C.H. Chao, Incremental forward feature selection with application to microarray gene expression data, Journal of biopharmaceutical statistics 18 (5) (2008) 827–840.
[11] G. Martínez-Muñoz, A. Suáárez, Aggregation ordering in bagging, in: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications, Acta Press, 2004, pp. 258–263.
[12] D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting, in: Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 211–218.
[13] M. Dettling, 〈http://stat.ethz.ch/∼dettling/bagboost.html〉, 2005.
[14] A. Statnikov, I. Tsamardinos, Y. Dosbayev, C.F. Aliferis, 〈http://www.gems-system.org/〉, 2009.
[15] U. Alon, et al., Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon cancer tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. 96 (1999) 6745–6750.
[16] C.C. Chang, C.J. Lin, Libsvm data: 〈http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/〉, 2009.
[17] C.C. Chang, C.J. Lin, LIBSVM: A Library for Support Vector Machines, 2001. Software available at: 〈http://www.csie.ntu.edu.tw/∼cjlin/libsvm〉.
[18] S. Niijima, S. Kuhara, Recursive gene selection based on maximum margin criterion: a comparison with SVM-RFE, BMC Bioinformatics 7 (2006) 543.