Neurocomputing 72 (2009) 991–999

Contents lists available at ScienceDirect

Neurocomputing

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

An efficient gene selection algorithm based on mutual information

Ruichu Cai [a,*], Zhifeng Hao [a], Xiaowei Yang [b], Wen Wen [a]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China
[b] College of Mathematics Science, South China University of Technology, Guangzhou 510640, China

ARTICLE INFO

Article history:
Received 17 July 2007
Received in revised form
11 April 2008
Accepted 18 April 2008
Communicated by Dr. L. Kurgan
Available online 9 May 2008

Keywords:
Microarray data
Gene expression
Gene selection
Mutual information
Parzen window density estimation

ABSTRACT

Gene selection, a significant preprocessing of the discriminant analysis of microarray data, is to select the most informative genes from the whole gene set. In this paper, an efficient mutual information-based gene selection algorithm (MIGS) is proposed, in which genes are sequentially forward selected according to an approximate measure of the mutual information between the class and the selected genes. In order to improve the efficiency of the MIGS, an effective pruning strategy is introduced in the selection procedure as well as the employment of Parzen window density estimation technique. Extensive experiments are conducted on three public gene expression datasets and the experimental results confirm the efficiency and effectiveness of the algorithm. Though the computational cost of MIGS-Pruning increases with the number of selected genes, it still has good performance applied in the microarray problems.

1. Introduction

In recent years, research on molecular biology and genetics has shifted from the study of individual genes to the exploration of the entire genome. DNA microarray is one of such techniques to measure the expression levels of thousands of genes in a single experiment, which is quite suitable for comparing the gene expression levels in tissues under different conditions, such as healthy versus diseased [12].

Discriminant analysis of microarray data has been widely studied to assist diagnosis [23]. Because lots of genes in the original gene set are irrelevant or even redundant for a specific discriminant problem, gene selection is usually introduced to preprocess the original gene set for further analysis. From the viewpoint of discriminant analysis, gene selection can increase the generalization ability of the classifier and reduce the computational complexity of the learning procedure. As far as the biologists concerned, gene selection provides more compact gene sets to reduce diagnosis costs and facilitate the understanding of functions of related genes.

In the context of pattern recognition, genes are usually treated as features and the gene selection problem can be solved as a feature selection problem. Generally, the feature selection methods can be classified into three categories: the filter, the wrapper and the embedded methods [17]. The filter method employs intrinsic properties of a feature without considering its interaction with other features, and the selection procedure is independent of the classifier. While in the wrapper method, a classifier is usually built and employed as the evaluation criterion. If the criterion is derived from the intrinsic properties of the classifier, the corresponding feature selection method is named as the embedded method.

Fisher's ratio [13], mutual information filter [13], Mahalanobis [6] and *t*-statistics [3] are classical filter methods. Recently, false discovery rate is proposed to pick out the differently expressed genes in the gene selection problem [26,32]. Although most of filter algorithms generate less compact feature sets than wrapper methods and embedded methods, they are more efficient. Thus, the filter algorithms are wildly used in the preprocessing of feature selection problems. In order to reduce the size of the feature set, the fisher ratio is exploited in this study.

The wrapper method is also widely used for gene selection. A typical wrapper method consists of two components: the search procedure and the evaluation criterion. Sequential forward selection (SFS) [33], sequential floating forward selection (SFFS) [33] and genetic algorithms [22] are typical search methods used in the gene selection problem. LS-Bound [33] and LOOC [29] are two criteria based on support vector machine (SVM), which is a commonly used classifier for the gene selection problem. Integrated the above two criteria into SFS scheme, LS-Bound SFS [33] and LOOC-SFS [29] methods are devised to obtain competitive results.

SVM recursive feature elimination (SVM-RFE) algorithm is a typical embedded method [14]. In SVM-RFE, the features are

* Corresponding author. Tel.: +86 13570393001; fax: +86 2039380032.
E-mail address: cairuichu@163.com (R. Cai).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.04.005

eliminated recursively according to the criterion derived from the intrinsic properties of the SVM classifier. SVM-RFE is often considered as one of the best gene selection algorithms in the literature [33] though it is time consuming. Lots of methods have been proposed to alleviate this problem by eliminating chunks of features at a time, such as Furlanello's entropy-based SVM-RFE [11], Ding's simulated annealing-based SVM-RFE [8]. A hybrid algorithm [20,27] of random subspace method (RSM) and SVM-RFE is also devised to improve the robustness and accuracy.

In this study, we pay our attention to a specific kind of feature selection method: the mutual information-based method. Mutual information is a good representation of relevance between two random variables. However, the computation of the mutual information in the high dimensional data is very difficult. Most of the existing work only considered the problems in the setting of low dimension [18,19]. Fleuret et al. [10] extended the mutual information-based feature selection algorithms in high dimensional cases, but this method only supports the binary variant. Tishby et al. [30] proposed an information bottleneck method, which employs a limited set of codewords to squeeze the mutual information between the features and the labels. In this paper, we introduce a new feature selection criterion based on the conditional mutual information. Meanwhile, a novel algorithm based on this criterion is also devised to perform gene selection efficiently.

The rest of this paper is organized as follows. Section 2 introduces the basic information theory. In Section 3, two mutual information-based gene selection methods are proposed: the basic mutual information gene selection method (MIGS) and an improved variant of MISG with pruning strategy (MIGS-Pruning). MIGS-Pruning is evaluated on three gene expression datasets in Section 4. Conclusions and discussions are presented in Section 5.

2. Entropy and mutual information

Information theory [5] provides an intuitive tool to measure the uncertainty of random variables and the information shared by them, in which the entropy and the mutual information are two critical concepts.

The entropy H is a measure of the uncertainty of random variables. Let X be a discrete random variable with alphabet χ and $p(x)=\Pr\{X=x\}\, x\in X$ be the probability mass function, the entropy of X is defined as

$$H(X) = -\sum_{x\in\chi} p(x)\log(p(x)) \tag{1}$$

While the mutual information is a measure of information shared by two random variables, defined as

$$\begin{aligned} I(Y;X) &= \sum_{x\in\chi}\sum_{y\in Y} p(y,x)\log\frac{p(x,y)}{p(x)p(y)} \\ &= H(Y) - H(Y|X) \end{aligned} \tag{2}$$

where $H(Y|X)$ is the conditional entropy of Y in the case of X is known, and can be represented as

$$H(Y|X) = -\sum_{x\in\chi}\sum_{y\in Y} p(y,x)\log(p(y|x)) \tag{3}$$

For the continuous random variables, the entropy and the mutual information are defined as

$$\begin{aligned} H(X) &= -\int_x p(x)\log(p(x))\,\mathrm{d}x \\ H(Y|X) &= -\int_{x,y} p(y,x)\log(p(y|x))\,\mathrm{d}x\,\mathrm{d}y \\ I(Y;X) &= \int_{x,y} p(y,x)\log\frac{p(x,y)}{p(x)p(y)}\,\mathrm{d}x\,\mathrm{d}y \end{aligned} \tag{4}$$

3. Fast gene selection algorithm based on mutual information

A typical gene expression dataset can be represented by $\{\langle x_i, y_i\rangle\}(i=1,\ldots,n)$, where n is the number of samples, x_i is an m dimensional vector representing the ith sample's expression profiles over the gene set $\mathbf{G}=\{g_1, g_2,\ldots,g_m\}$, and $y_i=\in\{-1, 1\}$ is the label of the sample. In the rest of this paper, we use $x_{i,j}$ to denote the expression level of gene g_j in the ith sample.

It is very common to have $n=100$ and $m=10{,}000$ in practice, so gene selection is an important preprocessing step in the gene expression data analysis context. In this paper, we study the gene selection problem under the classification framework which aims at selecting the most informative subset $\mathbf{S}(\mathbf{S}\subset\mathbf{G})$ to the classifier.

3.1. Mutual information gene selection

The goal of the feature selection is to select the smallest subset of features but carrying as much information about the class as possible. In this study, mutual information is used as the feature selection criteria because of its good representation of relevance between two random variables and its robustness in the environment of noises and data transformations [24]. Moreover, mutual information is robust for the classifier: theoretically, it can provide the optimal feature set regardless the classifiers [4].

An straightforward attempt is enumerating all subset $\mathbf{S}$ and choosing the subset which maximizes the mutual information criterion $I(Y;\mathbf{S})$ as the feature set [18,19]. This rudimentary method needs to estimate the high dimensional probability density function 2^m times, which causes intractable computation complexity. Furthermore, the error introduced in the estimation of high dimensional probability density function can be extremely large on the condition of no enough samples.

Another attempt is to consider the features independently. At first, the mutual information $I(Y;g_i)$, $i=1,\ldots,m$ between the class and each gene is calculated. And then, top-K features carrying larger mutual information of the class are selected. The main advantage of this method is its lower computational cost. However, it does not consider the relationships among different features, thus lots of redundancies might exist.

In this paper, we propose a novel selection criterion which employs a trade-off between $I(Y;\mathbf{S})$ and $I(Y;g_i)$, $i=1,\ldots,n$. In this new criterion, the interestingness of the gene $g_i\notin\mathbf{S}$ is evaluated by

$$\mathbf{MI}[i] = \min_{g_j\in S} I(Y;g_i|g_j) \tag{5}$$

where $\mathbf{MI}[i]$ is the ith element of the criteria vector $\mathbf{MI}$, and $\mathbf{S}$ is the set of genes that have already been selected. During the selection, the gene $g_i\notin\mathbf{S}$ with the highest $\mathbf{MI}[i]$ is selected. The employment of this criterion is based on the following observations: (1) From the information aspect, new genes holding rich information to Y can be selected through these criteria. The value of $I(Y;g_i|g_j)$ will be lower if gene g_i brings less information of Y or this information has already been contained in a selected gene g_j. So only gene g_i brings a lot of information to Y and the information has not been carried by any selected genes, $\min_{g_j\in S} I(Y;g_i|g_j)$ will be higher. (2) From the computational cost aspect, it is very efficient to estimate a three-dimensional probability density function. (3) Moreover, the criterion has the nice monotonic property for further improvement of this algorithm, which is presented in Section 3.2.

The criteria can be integrated with any existing search algorithms to construct new gene selection algorithms. For example, SFS algorithm is such a greedy heuristic search algorithm. To improve the performance, other complex search algorithms, such as SFFS, can be also used. In this paper, using the SFS algorithm, a novel SFS mutual information gene selection

(MIGS) algorithm is proposed which can be easily extended to other search algorithms.

MIGS is an iterative algorithm, and in each iteration one gene is selected from the unselected gene set **NS** and added to the selected gene set **S**. During the selection procedure, the criteria of each gene is explored and updated.

In the initialization, **S** is empty and **NS** contains all genes. The criteria of all genes are stored in the vector **MI**, where **MI**[i] represents the ith gene g_i's criterion with initialization **MI**[i] = $I(Y; i)$.

In the kth iteration, the following two steps are performed:

Firstly, sequentially scan the unselected gene set **NS**, select the gene with the highest criterion and insert it into the selected vector **S**. Because the criterion are contained in the vector **MI**, this selection procedure can be presented as follows:

$$\mathbf{S}[k] = \arg\max_i \{\mathbf{MI}[i]\} \quad \forall i \in \mathbf{NS} \tag{6}$$

After then, gene $g_{\mathbf{S}[k]}$ is removed from **NS**, and the criteria of genes are updated according to

$$\begin{cases} \mathbf{MI}[i] = \min\{\mathbf{MI}[i], I(Y; g_i|\mathbf{S}[k])\} & \forall i \in \mathbf{NS} \\ \mathbf{MI}[\mathbf{S}[k]] = 0 \end{cases} \tag{7}$$

The pseudo code of the gene selection algorithm is outlined in the following:

Algorithm 1: MIGS

Step 1: Initialization
  for $I = 1$ to m
    MI[i] = $I(Y, g_i)$;
  end
Step 2: Iterative selection procedure
  for $k = 1$ to K
    S[k] = arg max{**MI**[i]};
    MI[$S[k]$] $=^{i} 0$;
    for $i = 1$ to $m-k$
      MI[**NS**[i]] = min{**MI**[**NS**[i]], $I(Y;$ **NS**[i]|**S**[k])};
    end
  end.

3.2. Mutual information gene selection with pruning strategy

The criterion **MI**[i] = $\min_{g_j \in S} I(Y; g_i|g_j)$ proposed in the above section has a nice monotonic property: the criterion only decreases on the condition that more genes are added to the selected gene set **S**. This property can be formally represented by the following lemma.

Lemma 1. *For* $\forall S_1, S_2 \subset G$, *if* $S_1 \subset S_2$, $\min_{g_j \in S_2} I(Y; g_i|g_j) \leqslant \min_{g_j \in S_1} I(Y; g_i|g_j)$ *holds*.

Proof. Let $S_\Delta = S_2 - S_1$, according to the definition of the criterion, we know that $\min_{g_j \in S_2} I(Y; g_i|g_j) = \min\{\min_{g_j \in S1} I(Y; g_i|g_j), \min_{g_j \in S_\Delta} I(Y; g_i|g_j)\}$. Therefore,$\min_{g_j \in S2} I(Y; g_i|g_j) \leqslant \min_{g_j \in S1} I(Y; g_i|g_j)$ must hold. □

Let $\mathbf{S}_k$ be the selected gene in the kth iteration of MIGS, then we know that $\mathbf{S}_{k+1} \supset S_k$. According to Lemma 1, the criterion of each gene g_i can only decrease during the selection procedure, which is very similar to the Aprior [15]. This property can be used to prune plenty of updates of the vector **MI**.

For example, in Fig. 1 there are five genes and the initial value of the **MI** is **MI** = [0.3, 0.4, 0.3, 0.5, 0.2]. In the first iteration, gene g_4 is selected. Assume the updated criterion vector is **MI** = [0.25, 0.34, 0.29, 0, 0.18]. After the update, the new criterion of gene g_2 is **MI**[2] = 0.34, we can observe that the updates for gene g_3 and g_5 can be pruned according to Lemma 1. Note that, this pruning technique will not affect the accuracy of the algorithm because of the following two reasons: (1) the old criteria for gene g_3 and gene g_5 are 0.3 and 0.2, respectively, all of which are lower than 0.34. (2) The criteria can only decrease if the update is applied to gene g_3 and gene g_5.

However, some of pruned elements might need to be reconsidered in later iterations. Recall the example in Fig. 1, in the third iteration the updated criterion of gene g_1 is 0.2 which is lower than the original criterion of gene g_3, i.e., **MI**[3] = 0.3. So the criterion of gene g_3 needs to be updated in this iteration. In this case, the update of the criterion started from the last updating iteration of the gene, which is stored in vector *LUI*.

The above strategy can be further improved. Intuitively, genes with higher criteria will probably keep their advantages after update. Consider the second iteration of the above example, if we firstly update the criterion of gene g_2, i.e., **MI**[2] = 0.34, we will find that the criteria of gene g_1, g_3 and g_5 are all not necessary to be updated. Thus, the criterion of the genes can be sequentially updated according to the descending order of the criterion, which will help us to reduce much expensive calculation. This heuristic strategy can be easily implemented by sequentially updating the criteria of genes contained in **NS**, which is sorted in descending order according to **MI** (Fig. 2).

Until now, we obtain a more efficient variant of MIGS, i.e., MIGS with pruning strategy (MIGS-Pruning). The pseudo codes are outlined as follows:

Algorithm 2: MIGS-Pruning

Step 1: Initialization
  for $i = 1$ to m
    MI[i] = $I(Y, g_i)$;
    LUI[i] = 0;
  end
Step 2: Iterative selection procedure
  for $k = 1$ to K
    S[k] = arg max{**MI**[i]};
    MI[**S**[k]] = 0;

Iteration \ Gene	1	2	3	4	5
1	0.3	0.4	0.3	**0.5**	0.2
2	0.25	**0.34**	~~0.27~~	0	~~0.18~~
3	0.20	0	**0.21**	0	~~0.17~~

Fig. 1. The basic pruning strategy: the element in bold is the selected gene in this iteration; the elements with strikethrough are the calculation pruned in this iteration.

Iteration \ Gene	1	2	3	4	5
1	0.3	0.4	0.3	**0.5**	0.2
2	~~0.25~~	**0.34**	~~0.27~~	0	~~0.18~~
3	~~0.20~~	0	**0.21**	0	~~0.17~~

Fig. 2. The improved pruning strategy with heuristic information.

```
    Sort the vector NS according to MI in descending order;
    for i = 1 to m−k
        for j = LUI[NS[i]]+1 to k//Update the criterion
            MI[NS[i]] = min{MI[NS[i]], I(Y;g_NS[i]|g_S[i])};
        end
    end
    if MI[NS[i]]<MI[NS[i+1]]//the Pruning strategy is applied
        break;
    end
end.
```

3.3. Efficient calculation of mutual information with Parzen window

In MIGS-Pruning, the efficient and accurate calculation of $I(Y;g_i)$ and $I(Y;g_i|g_j)$ play a significant role in the performance of the algorithm. However, $I(Y;g_i)$ and $I(Y;g_i|g_j)$ are quite expensive to be calculated directly which require estimating the probability density function and integrating those functions. Some existing methods used the histograms to estimate the probability density function [18,19]. However, these methods produced low accuracy in the case of small sample size, and it is difficult to properly set parameters of the histograms. Accordingly, Parzen window method is devised to solve this problem, which is also used in [18,19].

According to (2):

$$\begin{aligned} I(Y;g_i) &= H(Y) - H(Y|g_i) \\ I(Y;g_i|g_j) &= H(Y|g_j) - H(Y|g_i,g_j) \end{aligned} \tag{8}$$

Thus, we can pay our attention to the efficient calculation of $I(Y;g_i)$ and $I(Y;g_i|g_j)$ due to $H(Y)$, $H(Y|g_i)$ and $H(Y|g_i, g_j)$ are all special cases of $H(Y|X)$ where X is a vector. In the following, we are looking for an efficient method to estimate $H(Y|X)$.

The ordinary form of $H(Y|X)$ is

$$H(Y|X) = -\int_{x,y} p(y,x)\log(p(y|x))\,dx\,dy \tag{9}$$

In the context of microarray, Y is the class of the sample with only two values, i.e., 1 and −1. So $H(Y|X)$ can be re-written as

$$H(Y|X) = -\int_{x} p(x)\sum_{y\in\{1,-1\}} p(y|x)\log(p(y|x))\,dx \tag{10}$$

However, the computation of (10) is still very difficult. Assume each sample x_i is random sampled from the distribution $p(x)$, based on the Monte Carlo integration method [2], (10) can be further simplified as

$$H(Y|X) = -\sum_{i=1}^{n}\frac{V}{n}\sum_{y\in\{1,-1\}} p(y|x_i)\log(p(y|x_i)) \tag{11}$$

where V is the volume of the area defined by x.

Given $y = y_0$, the value of $p(y_0|x)$ can be estimated according to the full probability formula:

$$p(y_0|x) = \frac{p(x,y_0)}{\Sigma_{y\in\{-1,1\}}p(x,y)} = \frac{p(x|y_0)p(y_0)}{\Sigma_{y\in\{-1,1\}}p(x|y)p(y)} \tag{12}$$

in which $p(y)$ can be easily estimated by $p(y) = n_y/n$ and $p(x/y)$can be estimated using the Parzen window density estimation:

$$\begin{aligned} p(x|y) &= \frac{1}{n_y}\Sigma_{i\in I_y}\phi(x-x_i,h) \\ &= \frac{1}{n_y}\Sigma_{i\in I_y}\frac{1}{(2\pi)^{d/2}h^d|\sum_y|^{1/2}}\exp\left(-\frac{(x-x_i)^{\mathrm{T}}\sum_y^{-1}(x-x_i)}{2h^2}\right) \end{aligned} \tag{13}$$

where I_y represents the sample set belonging to the class y, n_y is the number of samples contained in the set, and $\phi(x-x_i, h)$ is the kernel function. In this paper, the Gaussian window function with the same window width h and the same covariance matrix Σ_y is used for all class y. More detail about the Parzen window density estimation please refer to Appendix A.

Combining (12) with (13),

$$p(y_0|x) = \frac{\sum_{i\in I_{y0}} 1/|\sum_{y0}|^{1/2}\exp(-(x-x_i)^T\sum_{y0}^{-1}(x-x_i)/2h^2)}{\sum_{y\in\{-1,1\}}\sum_{i\in I_y} 1/|\sum_y|^{1/2}\exp(-(x-x_i)^T\sum_y^{-1}(x-x_i)/2h^2)} \tag{14}$$

To date, $H(Y|X)$ can be efficiently estimated through the following steps: (1) estimate $p(y|x)$ according to formula (14); (2) $H(Y|X)$ can be easily calculated by replacing $p(y|x)$ with the estimated value in the first step.

4. Experiments and results

In this section, we evaluated MIGS-Pruning on three open microarray datasets: Leukaemia, Breast-LN and Colon Cancer. All of them are preprocessed using the technique described in [9]. After thresholding, filtering and logarithmic-transforming, the microarray data are standardized to zero mean and unit standard deviation across genes (Table 1).

In order to reduce the number of features and the computational time, top 1000 genes are pre-selected from each dataset according to the fisher's ratio which is defined as $f(g_i)\mu_1-\mu_2/\sigma_1^2+\sigma_2^2$, where μ_1, μ_2, σ_1, σ_2 denote the means and standard deviations of two classes, respectively. All the simulations and comparisons (except the experiment to test the computational time in terms of the size of gene sets) are performed on the pre-selected datasets in this study.

To illustrate the efficiency of MIGS-Pruning, it is compared with three prevail gene selection algorithms: mutual information-based filter, LS-Bound SFS and SVM-RFE. Among them, the mutual information filter is efficient but a lot of redundant features existed in the selected set; while LS-Bound SFS and SVM-RFE can provide more compact selected gene sets but their computational costs are expensive.

In the implementation of the mutual information filter, the Parzen window estimation is exploited to estimate the probability density function. The LS-Bound SFS and SVM-RFE are implemented in the standard version according to [33,14], respectively. And all of the following experiments are conducted in Visual C++ 6.0 environment on a PC with 2.8 GHz P4 CPU and 512 MB RAM.

4.1. The performance of MIGS-Pruning

In the context of discriminant analysis of microarray data, whether the selected gene subset can provide good generalization ability to the classifier or not is very important. In our study, we use external B.632+ to assess the performance of different gene selection algorithms. B.632+ is an unbiased estimation of the generalization ability of a classifier, which is very suitable for the

Table 1
Basic information of three microarray datasets

Dataset	Number of samples	Number of features
Leukemia	72	7129
Breast-LN	49	7129
Colon cancer	62	2000

cases of small sample size [1]. In B.632+, the balanced bootstrap samples are generated for t times, and the samples not contained in the training set constructed the corresponding testing set. Each sample in the original sample set is made to appear exactly t times in the balanced bootstrap samples. In order to reduce the variance of the algorithm's performance, the bootstrap is repeated for $t = 200$ times.

The standard SVM with linear kernel was employed as the classifier to estimate the error rates of different gene selection algorithms. The linear kernel is the most preferred kernel in the gene expression analysis context [33,29,14]. The parameters of the SVM were tuned for each gene set.

In the following experiments, the largest number of selected genes is 50. However, these 50 genes are not the final result because few of them can achieve the same or satisfactory performance. In our opinions, the B.632+ reflects the generalization performance of the selected gene subset. The decision of the final gene subset can be made based on whether the B.632+ approaches the minimum, or whether adding more genes results in insignificant changes.

Figs. 3–5 illustrate the external B.632+ errors on Leukaemia, Breast-LN and Colon Cancer datasets respectively. On the Leukaemia dataset, 23 genes are selected finally and MIGS-Pruning obtains the lowest B.632+ error. Similarly, 12 genes are

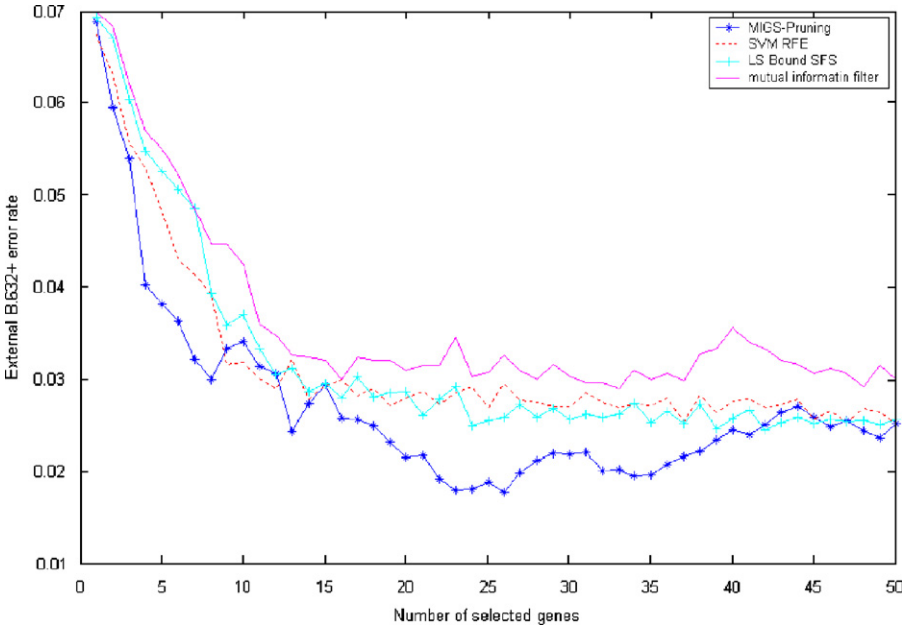

Fig. 3. The external B.632 +error for the Leukemia dataset.

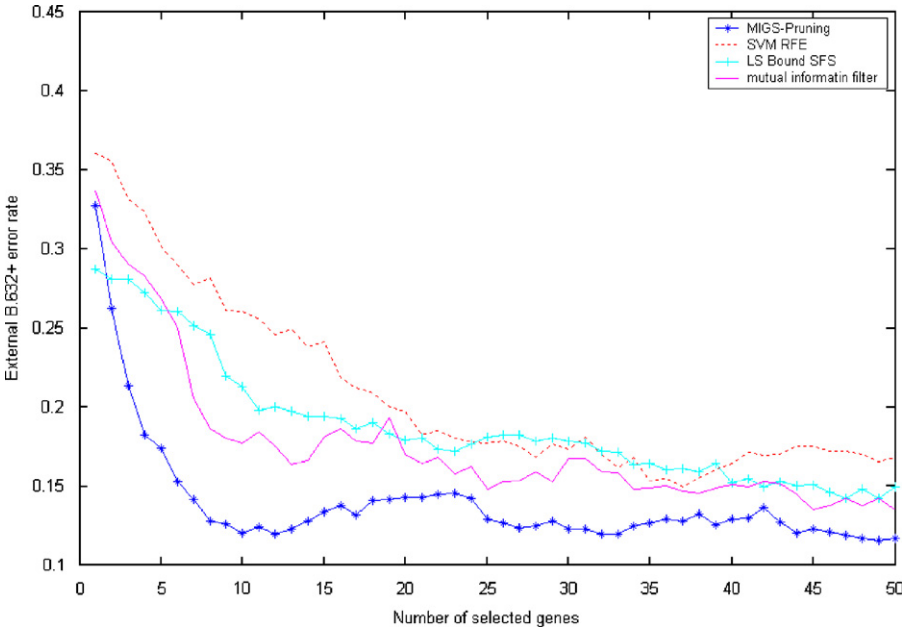

Fig. 4. The external B.632 +error for the Breast-LN dataset.

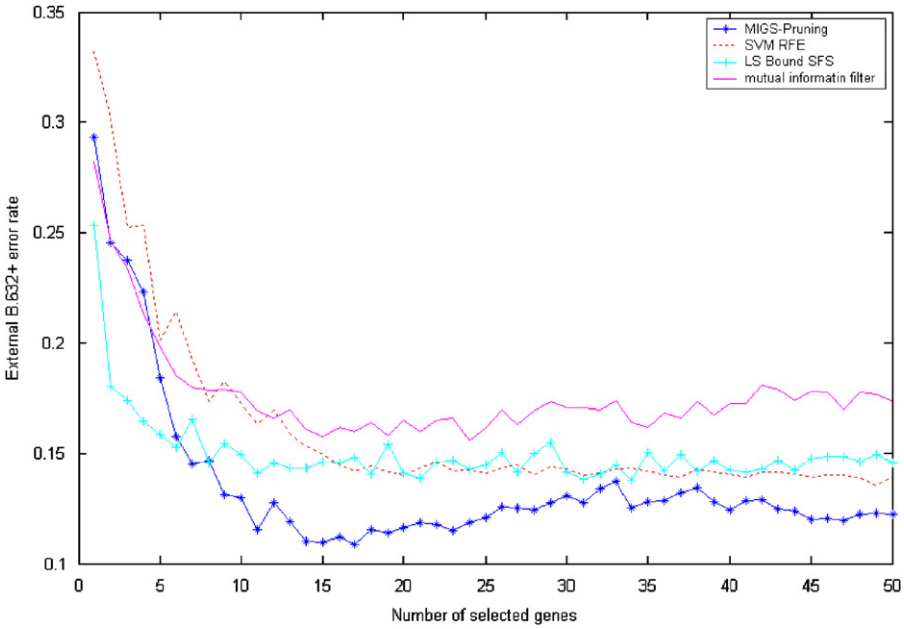

Fig. 5. The external B.632 +error for the colon cancer dataset.

selected from the Breast-LN dataset and 14 genes from the Colon Cancer dataset.

Generally speaking, MIGS-Pruning achieves the lowest B.632+ error among the four gene selection algorithms.

When very few genes were selected, all of these four gene selection algorithms had similar performance. Its reasonable, because LS-Bound, SVM-RFE, mutual information filter and MIGS-Pruning are all greedy search algorithms, and they can achieve reasonable local optimal results when the number of selected genes is very small. In addition, the redundancy between the selected genes can be ignored, so MIGS-Pruning and the mutual information algorithm obtain similar results.

However, while the number of selected genes increases, MIGS-Pruning is the best choice because of the following reasons: (1) in the criterion of MIGS-Pruning, the redundancy among genes are considered; (2) only three-dimensional probability density function needs to be estimated which is very suitable for the case of lower sample number; (3) the Parzen window estimation is used to estimate the probability density function which improves the accuracy of the estimation.

4.2. The analysis of the selected genes

Another important criterion of evaluating gene selection methods is correlation between the selected genes and a certain disease. We use the selected times of a gene in 200 repeated iterations to evaluate the importance of this gene, and the most 10 important genes are listed for each dataset. The result on the Leukemia dataset is presented in Table 2, and results for the rest datasets are given in Appendix A.

From Table 2, we can conclude that all of the 10 genes are very robust during the selection, even the 10th gene U40343 are selected for 181 times during 200 repeated selection iterations.

Most of the 10 genes have been reported to be related to the leukemia. M84526(Adipsin) is associated with myeloid cell differentiation [31], and the myeloid cell is strongly related to the leukemia. X95735 (Zyxin) encodes an important protein for cell adhesion and is highly correlated with acute myelogenous leukemia [16]. J03779(CD10) is found to be the common acute lymphoblastic leukemia antigen early in 1989 [21]. And L15326(PTGS2,COX2) is identified as one of the up-regulated genes in the leukemia sample [28]. S57212(MEF2C) is activated by multiple mechanisms in a subset of T-acute lymphoblastic leukemia cell lines [7]. U88667(ABCA4) is recognized as associated with drug resistance in [25].

Table 2
The selected gene for Leukemia

Access no.	Selected times	Description
M84526	200	Adipsin (D component of complement)
X95735	200	Zyxin
J03779	200	CD10
L15326	197	PTGS2(COX2)
D26308	196	NADPH-flavin reductase
S57212	192	MEF2C
U75276	192	BRF1
U88667	189	ABCA4
M31994	184	ALDH1
U40343	181	CDKN2D

Though there are some researches on the genes U40343, M31994, U75276 and D26308, the relationships between these genes and the leukemia are still not clear. Some important knowledge might be discovered if biologists pay more attention to these genes.

4.3. The computational complexity of MIGS-Pruning

The computational complexity is an important aspect of gene selection algorithms. In the following, the computational time of MIGS-Pruning in terms of the number of selected genes and the size of the whole gene set will be evaluated. Note that, the time shown in the following is the average time of 200 running.

Fig. 6 illustrates the runtime of four different algorithms with different numbers of selected genes. As shown in this figure,

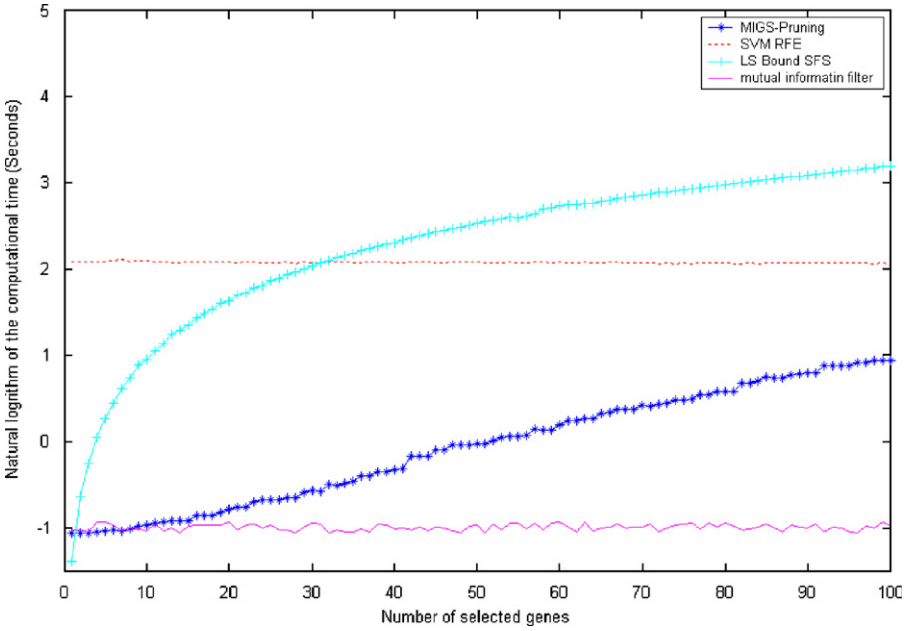

Fig. 6. The relationship between the computational time and the number of selected genes on Leukemia dataset.

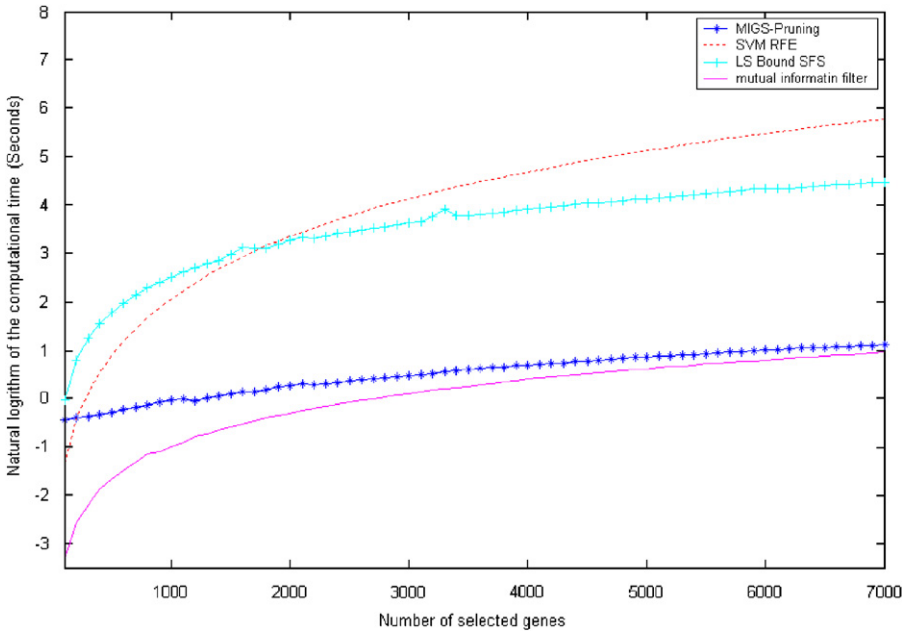

Fig. 7. The relationship between the computational time and the size of the gene set on Leukemia dataset.

MIGS-Pruning outperformed SVM-RFE and LS Bound SFS. But MIGS-Pruning is slightly inferior to the mutual information filter, which is one of fastest gene selection methods. The main shortcoming of MIGS-Pruning is that the computational cost increases significantly with the increase of the number of selected genes. However, it has little influence in the applications of microarray data, where selected genes are related to a certain disease (such as a cancer) and the number is usually lower than 50.

Fig. 7 illustrates the computational time of the algorithm in terms of the size of gene set. As shown in this figure, MIGS-Pruning performs similarly with the most efficient method, i.e., mutual information filter. Especially, when the size of the gene set is larger than 3000, the difference between the computational cost of MIGS-Pruning and the mutual information filter is very trivial.

Furthermore, in contrast with MIGS-Pruning, the computational costs of SVM-RFE, LS-Bound SFS and mutual information filter increase significantly with the increase of the size of gene set. This advantage of MIGS-Pruning mainly benefits from the pruning strategy of MIGS. When the gene set size is large, a lot of updates of the criteria and computation will be avoided. This also confirms that MIGS-Pruning retains the characteristic of efficiency even in the case of large size of the gene set. This nice characteristic is especially important in the microarray data analysis.

5. Conclusions and future work

In this paper, an efficient gene selection algorithm MIGS-Pruning is proposed based on mutual information. In MIGS-Pruning, genes are sequentially selected according to the newly proposed criterion, which can be easily estimated using the Parzen window estimation. An effective pruning technique is also devised to reduce the computational cost of MIGS-Pruning.

Experimental results show that MIGS-Pruning outperforms SVM-RFE and LS-Bound SFS in both accuracy and efficiency. Furthermore, MIGS-Pruning shows good scalability in terms of the size of input datasets. Although the computational cost of MIGS increases significantly with the number of selected genes, it is still efficient in the applications of the microarray problems. Moreover, MIGS-Pruning can be easily generalized to the multi-classification feature selection problems, which is quite difficult for SVM-RFE and LS-Bound SFS.

Moreover, MIGS-Pruning can be further improved by integrating with other search scheme to enhance the efficiency, extending it to the multi-classification case or conducting more thorough theoretical analysis of MIGS-Pruning.

Appendix A. Parzen window density estimation technology

Parzen window density estimation is one of the most used nonparametric density estimation technology. It essentially superposes window functions placed at each sample. In this way, each sample contributes to the probability density function estimate. Given a set of n d-dimensional samples $\{x_1, x_2, \ldots, x_n\}$, the Parzen window density estimation is as following:

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} \phi(x - x_i, h) \tag{A.1}$$

where $\phi(x-x_i, h)$ is the window function and h is the window width.

Parzen showed that $p(x)$converges to the true density if $\phi(x-x_i, h)$ and h are selected properly. The window function is required to be a finite-valued nonnegative density function where $\int \phi(x, h)\, dx = 1$, and the width parameter is required to be a function of n such that $\lim_{n\to+\infty} h(n) = 0$ and $\lim_{n\to+\infty} nh^{d}(n) = \infty$.

The rectangular and the Gaussian window functions are commonly used window function. The choice of the window function is strongly related to the noise type of the sample. In the gene expression data, the noise satisfies Gaussian statistics, and the following Gaussian window function $K(x-x_i)$ is used:

$$\phi(x - x_i, h) = \frac{1}{(2\pi)^{d/2} h^{d} |\sum|^{1/2}} \exp\left(-\frac{(x - x_i)^{\mathrm{T}} \sum^{-1} (x - x_i)}{2h^2}\right) \tag{A.2}$$

According to (A.1) and (A.2), we have

$$p(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{(2\pi)^{d/2} h^{d} |\Sigma|^{1/2}} \exp\left(-\frac{(x - x_i)^{\mathrm{T}} \Sigma^{-1} (x - x_i)}{2h^2}\right) \tag{A.3}$$

where Σ is the covariance matrix that can be directly estimated from the dataset.

Appendix B. The selected gene for the Breast-LN and colon cancer dataset

The selected gene for Breast-LN is shown in Table B1 and the selected gene for colon cancer in Table B2.

Table B1
The selected gene for breast-LN

Access no.	Selected times	Description
J02906	200	Human cytochrome P450IIF1 protein (CYP2F) mRNA, complete cds
S60415	200	Myasthenic Syndrome Antigen B
J04423	200	*E. coli* bioB gene biotin synthetase
U07807	197	Metallothionein IV (MT4)
X56681	195	Jun D proto-oncogene (JUND)
S54005	190	thymosin beta-10
X97065	183	Sec23 homolog B
X03453	180	Bacteriophage P1 cre gene for recombinase protein
Y10871	177	twist homolog 1
HG2668	179	Bradykinin receptor

Table B2
The selected gene for colon cancer

Access no.	Selected times	Description
X55715	200	Human Hums3 mRNA for 40S ribosomal protein s3.
R87126	200	Myosin heavy chain, nonmuscle (Gallus gallus)
T70062	198	Interleukin enhancer binding factor 2
L37792	195	Human syntaxin 1A mRNA, complete cds
R88740	188	ATP synthase coupling factor 6, mitochondrial precursor
M84349	182	Human transmembrane protein (CD59) gene
T57882	176	Myosin heavy chain, nonmuscle type A
T63508	173	Ferritin heavy chain
T61661	170	Profilin I
H15813	160	CCAAT/ENHANCER binding protein beta

References

[1] C. Ambroise, G.J. McLachlan, Selection bias in gene extraction on the basis of microarray gene-expression data, Proc. Natl. Acad. Sci. USA 99 (10) (2002) 6562–6566.
[2] B.A. Berg, Markov Chain Monte Carlo Simulations and Their Statistical Analysis, World Scientific, Singapore, 2004.
[3] C.F. Chang, K.M. Wai, H.G. Patterton, Calculating the statistical significance of physical clusters of co-regulated genes in the genome: the role of chromatin in domain-wide gene regulation, Nucl. Acids Res. 32 (5) (2004) 1798–1807.
[4] T.W.S. Chow, D. Huang, Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information, IEEE Trans. Neural Networks 16 (1) (2005) 213–224.
[5] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, New Jersey, 2005.
[6] R.M.C.R. de Souza, F.A.T. de Carvalho, C.P. Tenorio, Two partitional methods for interval-valued data using mahalanobis distances, Adv. Artif. Intell.—Iberamia 2004 3315 (2004) 454–463.
[7] S. Debernardi, et al., Genome-wide analysis of acute myeloid leukemia with normal karyotype reveals a unique pattern of homeobox gene expression distinct from those with translocation-mediated fusion events, Genes Chromosomes Cancer 37 (2) (2003) 149–158.
[8] Y.Y. Ding, D. Wilkins, Improving the performance of SVM-RFE to select genes in microarray data, BMC Bioinformatics 7 (2006).
[9] S. Dudoit, J. Fridlyand, T.P. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, J. Am. Stat. Assoc. 97 (457) (2002) 77–87.
[10] F. Fleuret, Fast binary feature selection with conditional mutual information, J. Mach. Learning Res. 5 (2004) 1531–1555.
[11] C. Furlanello, et al., Entropy-based gene ranking without selection bias for the predictive classification of microarray data, BMC Bioinformatics 4 (2003).
[12] T.R. Golub, et al., Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (5439) (1999) 531–537.
[13] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learning Res. 3 (2003) 1157–1182.
[14] I. Guyon, et al., Gene selection for cancer classification using support vector machines, Mach. Learning 46 (1–3) (2002) 389–422.

[15] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, CA, 2000.
[16] L. Kelly, J. Clark, D.G. Gilliland, Comprehensive genotypic analysis of leukemia: clinical and therapeutic implications, Curr. Opin. Oncol. 14 (1) (2002) 10–18.
[17] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1–2) (1997) 273–324.
[18] N. Kwak, C.H. Choi, Input feature selection for classification problems, IEEE Trans. Neural Networks 13 (1) (2002) 143–159.
[19] N. Kwak, C.H. Choi, Input feature selection by mutual information based on Parzen window, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1667–1671.
[20] C. Lai, M.J.T. Reinders, L. Wessels, Random subspace method for multivariate feature selection, Pattern Recogn. Lett. 27 (10) (2006) 1067–1076.
[21] T.W. LeBien, R.T. McCormack, The common acute lymphoblastic leukemia antigen (CD10)—emancipation from a functional enigma, 1989, pp. 625–635.
[22] L.B. Li, et al., A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset, Genomics 85 (1) (2005) 16–23.
[23] C.L. Nutt, et al., Gene expression-based classification of malignant gliomas correlates better with survival than histological classification, Cancer Res. 63 (7) (2003) 1602–1607.
[24] I. Priness, O. Maimon, I. Ben-Gal, Evaluation of gene-expression clustering via mutual information distance measure, 2007, p. 111.
[25] M. Raaijmakers, ATP-binding-cassette transporters in hematopoietic stem cells and their utility as therapeutical targets in acute and chronic myeloid leukemia, Leukemia 21 (10) (2007) 2094–2102.
[26] A. Reiner, D. Yekutieli, Y. Benjamini, Identifying differentially expressed genes using false discovery rate controlling procedures, Bioinformatics 19 (3) (2003) 368–375.
[27] Ruichu Cai, Zhifeng Hao, W. Wen, A novel gene ranking algorithm based on random subspace method, in: Neural Networks, 2007, IJCNN 2007, International Joint Conference on, 2007, Orlando, FL.
[28] P. Secchiero, et al., Potential pathogenetic implications of cyclo-oxygenase-2 overexpression in B chronic lymphoid leukemia cells, 2005, pp. 1599–1607.
[29] E.K. Tang, P.N. Suganthan, X. Yao, Gene selection algorithms for microarray data based on least squares support vector machine, BMC Bioinformatics 7 (2006).
[30] N. Tishby, F.C. Pereira, W. Bialek, The information bottleneck method, in: The 37th Annual Allerton Conference on Communication, Control and Computing, 1999.
[31] E.T.L. Wong, et al., Changes in chromatin organization at the neutrophil elastase locus associated with myeloid cell differentiation, Blood 94 (11) (1999) 3730.
[32] J.J. Yang, M.C.K. Yang, An improved procedure for gene selection from microarray experiments using false discovery rate criterion, BMC Bioinformatics 7 (2006).
[33] X. Zhou, K.Z. Mao, LS bound based gene selection for DNA microarray data, Bioinformatics 21 (8) (2005) 1559–1564.

Ruichu Cai, born in 1983, is Ph.D. candidate in the School of Computer Science and Engineering, South China University of Technology. His researches are mainly in the fields of Data mining and Bioinformatics, including statistical learning theory, frequent pattern discovery and its application to the microarray data analysis.

Zhifeng Hao, born in 1968, Ph.D., and Professor in the School of Computer Sciences and Engineering, South China University of Technology. He has over 80 publications in journals and conference proceedings. His research interests are mainly in the fields of algebra, machine learning, bioinformatics and intelligence computation.

Xiaowei Yang, born in 1969, Ph.D., and Associate Professor in the College of Mathematical Sciences, South China University of Technology. He has over 70 publications in journals and conference proceedings. His research interests are mainly in the fields of support vector machine, intelligence computation, topology optimization and computational mechanics.

Wen Wen, born in 1981, Ph.D. candidate in the College of Computer Science and Engineering, South China University of Technology. Her research interests include kernel methods and pattern recognition.