A Framework of Gene Subset Selection Using Multiobjective Evolutionary Algorithm

Yifeng Li, Alioune Ngom, and Luis Rueda

School of Computer Sciences, 5115 Lambton Tower, University of Windsor,
401 Sunset Avenue, Windsor, Ontario, N9B 3P4, Canada
{li11112c,angom,lrueda}@uwindsor.ca
http://cs.uwindsor.ca/uwinbio

Abstract. Microarray gene expression technique can provide snap shots of gene expression levels of samples. This technique is promising to be used in clinical diagnosis and genomic pathology. However, the curse of dimensionality and other problems have been challenging researchers for a decade. Selecting a few discriminative genes is an important choice. But gene subset selection is a NP hard problem. This paper proposes an effective gene selection framework. This framework integrates gene filtering, sample selection, and multiobjective evolutionary algorithm (MOEA). We use MOEA to optimize four objective functions taking into account of class relevance, feature redundancy, classification performance, and the number of selected genes. Experimental comparison shows that the proposed approach is better than a well-known recursive feature elimination method in terms of classification performance and time complexity.

Keywords: gene selection, sample selection, non-negative matrix factorization, multiobjective evolutionary algorithm.

1 Introduction

Microarray gene expression data are obtained through monitoring the intensities of mRNAs corresponding to tens of thousands of genes [1]. There are two types of microarray data: gene-sample data, which compile the expression levels of various genes over a set of biological samples; and gene-time data, which record the expression levels of various genes over a series of time-points. Both types of data can be represented by a two-dimensional (2D) gene expression matrix. This technique provides a huge amount of data to develop decision systems for cancer diagnosis and prognosis, and to find co-regulated genes, functions of genes, and genetic networks. In this study, we focus on this first application through devising efficient and effective gene selection and classification approaches for gene-sample data. The gene-sample data includes data from two classes, for example, healthy samples and tumorous samples. However, noise, curse of dimensionality, and other problems substantially affect the performance of analysis algorithms devised for microarray data. There are two computational solutions for this problem: feature extraction or feature selection. Feature extraction methods are devised to generate new features, for example the research in [2] extracted non-negative new features/metagenes [3] using *non-negative matrix factorization* (NMF) [4]. And feature

T. Shibuya et al. (Eds.): PRIB 2012, LNBI 7632, pp. 38–48, 2012.

selection aims to select a few number of features/genes, which is termed *gene selection*. Gene selection is based on the assumption that only few number of genes contribute to a specific biological phenotype, while most of genes are irrelevant with this. The advantage of gene selection is that it provides directly a small gene subset for biological interpretation and exploration. Any gene selection method needs a gene (or a subset of genes) evaluation criterion to score a gene (or a subset of genes). A search strategy is required for any gene subset selection method, while few gene ranking methods need this strategy.

In the past decade, most feature selection methods were employed for selecting genes, and some feature selection methods are invented specifically for microarray data [5]. *minimum redundancy - maximum relevance* (mRMR) [6] [7] is reported as an efficient and effective method and enjoying much attention. In this method, the mutual information based criteria are proposed to measure the class relevance and feature redundancy. The size of gene subset is fixed by mRMR, and a linear/gready search strategy is proposed. However, it is difficult to decide the weights when combine the two measures into one criterion. *Support vector machine recursive feature elimination* (SVM-RFE) is another successful method for gene selection [8] [9] [10]. SVM-RFE only uses support vectors to rank genes, which is an idea of combining sample selection in gene selection because SVM-RFE selects the boundary samples. There are also some other ideas that prototypic samples are selected to avoid using outliers. Interested reader are referred to [9] for a concise review of sample selection. SVM-RFE can be viewed as both gene ranking method and gene subset selection method. Take Algorithm 2 for example, if the first step (backward search) is only used to sort genes, it is a ranking method; whereas if it involves forward search after backward search to include the sorted genes one by one until the classification performance degenerates, then it is a gene subset selection method. SVM-RFE does not fix the size of gene subset. Mundra and Rajapakse combined the mRMR measure with SVM-RFE (SVM-RFE-mRMR) [11] and reported better accuracy than the original mRMR and SVM-RFE methods. Even a linear search is used to decide the weight of the combination, this is not practically efficient, and the weighting issue between the relevance and redundancy measures is not solved either. Another issue is that SVM-RFE-mRMR may includes unnecessary genes in the gene subset in two cases. Firstly, if the current best validation accuracy in the validation step meets 1, SVM-RFE-mRMR may continue adding genes in the subset until the current validation accuracy is less than 1. For instance, the sequence of the best validation accuracy is $[0.6, 0.8, 1, 1, 1, 0.9]$ and the sorted genes in ascent order is $[\cdots, g_8, g_3, g_{10}, g_2, g_9, g_6]$, SVM-RFE-mRMR may return $[g_6, g_9, g_2, g_{10}, g_3]$, but the algorithm should terminate at the third iterations and return $[g_6, g_9, g_2]$. Secondly, if the current best validation accuracy is less than 1, and this is unchanged until the current validation accuracy is less than it. SVM-RFE-mRMR may keep adding all genes before this. Let us use the above example. If we change 1 to 0.95, similarly SVM-RFE-mRMR may return $[g_6, g_9, g_2, g_{10}, g_3]$. Moreover, since SVM-RFE-mRMR uses a variant of backward search and the number of genes is usually very large, it is too computationally expensive to apply in practice. Computational intelligence approaches, for example evolutionary algorithm, have been used for searching gene subsets. The most crucial part of these approaches is the fitness functions. Good performance has been reported in [12] [13]. This encourages us to design new fitness functions for better result.

In order to apply all the advantages and overcome the disadvantages discussed above, we propose a comprehensive framework to select gene subsets. This framework includes a NMF based gene filtering method, a SVM based sample selection method, and a search strategy use *multiobjective evolutionary algorithm* (MOEA). This MOEA optimizes four fitness functions. Let us call this framework: the MOEA based method for notational simplicity. We also revise the SVM-RFE-mRMR algorithm to solve all its problems, except the weighting issue. In this study, we compared both of the MOEA based method and SVM-RFE-mRMR.

2 Methods

2.1 MOEA Based Gene Subset Selection

In this section, the MOEA based gene subset selection is described in Algorithm 1, and is detailed as below.

Algorithm 1. *MOEA Based Gene Subset Selection*

```
Input: $\boldsymbol{D}$, of size $m$(genes) × $n$(samples), and the class labels $\boldsymbol{c}$
Output: the selected gene subsets: $\boldsymbol{G}$, the best validation accuracy $av$ and its corresponding gene
        subsets $\boldsymbol{G}_{\text{b}}$ ($\boldsymbol{G}_b \subseteq \boldsymbol{G}$), and the list of survived genes $\boldsymbol{f}$
  1. split $\boldsymbol{D}$ into training set $\boldsymbol{D}_{\text{tr}}$ and validation subset $\boldsymbol{D}_{\text{val}}$. Partition $\boldsymbol{D}_{\text{tr}}$ into training subset
     $\boldsymbol{D}_{\text{tr}}^{\text{tr}}$ and test subset $\boldsymbol{D}_{\text{tr}}^{\text{te}}$
  2. NMF based gene filtering (input: $\boldsymbol{D}_{\text{tr}}^{\text{tr}}$ and the number of survived genes $K$; output: $K$
     survived genes $\boldsymbol{f}$)
  3. SVM based sample selection (input: $\boldsymbol{D}_{\text{tr}}^{\text{tr}} = \boldsymbol{D}_{\text{tr}}^{\text{tr}}(f,:)$ and $\boldsymbol{c}_{\text{tr}}^{\text{tr}}$; output: $\boldsymbol{D}_{\text{tr}}^{\text{tr}} = \boldsymbol{D}_{\text{tr}}^{\text{tr}}(:,s)$,
     where $\boldsymbol{s}$ is the selected samples)
  4. search gene subsets by MOEA (input: $\boldsymbol{D}_{\text{tr}}^{\text{tr}}$, $\boldsymbol{c}_{\text{tr}}^{\text{tr}}$, $\boldsymbol{D}_{\text{tr}}^{\text{te}}$, and $\boldsymbol{c}_{\text{tr}}^{\text{te}}$; output: $p$ gene subsets
     $\boldsymbol{G} = \{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_p\}$)
  5. obtain the best validation accuracy and its corresponding gene subsets(input: $\boldsymbol{D}_{\text{tr}}(\boldsymbol{f},:)$,
     $\boldsymbol{c}_{\text{tr}}$, $\boldsymbol{D}_{\text{val}}(\boldsymbol{f},:)$, $\boldsymbol{c}_{\text{val}}$, and $\boldsymbol{G}$; output: the best validation accuracy $av$ and its corresponding
     gene subsets $\boldsymbol{G}_{\text{b}}$)
```

NMF Based Gene Filtering. Gene filtering methods aim to remove some genes which have low ranking scores. This idea is based on the assumption that the the genes with low variations across classes do not contribute to classification. Many gene filtering criteria based on t-test, variance, entropy, range, and absolute values. has been widely used [14]. In this study, we use a novel non-negative matrix factorization (NMF) [4] based criteria, because microarray gene expression intensities are non-negative, and it has been experimentally proved that this criterion works well on microarray data [15] [2]. Suppose $\boldsymbol{D}_{\text{tr}}^{\text{tr}}$ contains m genes and l samples, it can be decomposed as follows

$$\boldsymbol{D}_{\text{tr}}^{\text{tr}} \approx \boldsymbol{A}\boldsymbol{Y}, \quad \boldsymbol{D}_{\text{tr}}^{\text{tr}}, \boldsymbol{A}, \boldsymbol{Y} \geq 0, \tag{1}$$

where $\boldsymbol{D}_{\text{tr}}^{\text{tr}}$, $\boldsymbol{A}$, and $\boldsymbol{Y}$ are of size $m \times l$, $m \times r$, and $r \times l$, respectively. $r < \min(m, l)$. A and Y are the basis matrix and the coefficient matrix, respectively. In the application of clustering and feature extraction, columns of A are called metagenes [3] [2] which spans the *feature space*. Each sample is a non-negative linear combination of metagenes. Metagenes are hidden patterns extracted from the original intensity data. Instead of analyzing the original data, we use a criterion on $\boldsymbol{A}$, as below

$$Gene\_score(i) = 1 + \frac{1}{\log_2(r)} \sum_{j=1}^{r} p(i,j) \log_2 p(i,j), \tag{2}$$

where $p(i,q) = \frac{A[i,q]}{\sum_{j=1}^{r} \boldsymbol{A}[i,j]}$. This criterion is based on entropy in information theory. the assumption that if the ith row, corresponding to the ith gene, exhibits discriminability across the metagenes, we say this gene contribute to classification. We select K genes with the top K scores. The differences between this and the above mentioned feature ranking methods are that this criterion is unsupervised and operates on the extracted features, instead of directly on the original data.

SVM Based Sample Selection. Since we use a MOEA as search strategy, we hope the fitness values are calculated as fast as possible. Meanwhile, we also expect the gene selection can use essential samples. In this study, we therefore use a simple sample selection to select bounder samples. A linear SVM [16] [17] is trained over $\boldsymbol{D}_{\text{tr}}^{\text{tr}}$, and the support vectors are used as input of the MOEA gene selection module to calculate the fitness values.

Multiobjective Evolutionary Algorithm. The following four points should be considered when a high-quality gene subset method is being designed. 1) All the genes in a subset should be relevant to classify the samples as correct as possible. 2) The genes in a subset should be as diverse as possible rather than most of selected genes have the similar profiles. 3) The prediction accuracy and generalization of the selected subsets should be as good as possible. 4) At the same time, the gene subsets should be as small as possible. However, 1) and 2) conflict to some extent. 3) is also conflict with 4). MOEA can optimize more than one (conflicting) objectives and return the Pareto front which are a collection of the non-inferior solutions [18]. Since we have the above four criteria, MOEA should naturally be used to solve the weighting problem instead of using the classical methods to combine them into a single objective using weights as [6] and [11] did. NSGA-II [19], a well-known MOEA algorithm, is used in this study. We customize this algorithm for our application as below.

An individual in the population should be a gene subset. Suppose the length of the survived gene list $\boldsymbol{f}$ in Algorithm 1 is h, we encode a gene subset into a 0-1 binary array of length h. For an individual $\boldsymbol{b}$, $\boldsymbol{b}[i] = 1$ indicates the ith gene is selected in the subset.

Four fitness functions, considering to class relevance, gene redundancy, prediction accuracy, and gene size, are used. They are formulated as below:

$$f_1(\boldsymbol{b}) = \frac{1}{\frac{1}{\text{sum}(\boldsymbol{b})} \sum_{\boldsymbol{b}[i]=1} I(i, \boldsymbol{c}_{tr}^{tr})}, \tag{3}$$

where $I(i, \boldsymbol{c}_{tr}^{tr})$ is the mutual information of the ith discretized gene profile and the class labels on data $\boldsymbol{D}_{\mathrm{tr}}^{\mathrm{tr}}$;

$$f_2(\boldsymbol{b}) = \frac{1}{\mathrm{sum}(\boldsymbol{b})} \sum_{\boldsymbol{b}[i]=1, \boldsymbol{b}[i']=1} I(i, i'), \tag{4}$$

where $I(i, i')$ is the mutual information of the ith and i'th discretized gene profiles;

$$f_3(\boldsymbol{b}) = linearSVM(\boldsymbol{D}_{\mathrm{tr}}^{\mathrm{tr}}, \boldsymbol{D}_{\mathrm{tr}}^{\mathrm{te}}), \tag{5}$$

where $linearSVM$ is a linear SVM classifier trained on $\boldsymbol{D}_{\mathrm{tr}}^{\mathrm{tr}}$, and returns the prediction accuracy of $\boldsymbol{D}_{\mathrm{tr}}^{\mathrm{te}}$; and

$$f_4(\boldsymbol{b}) = \frac{\mathrm{sum}(\boldsymbol{b})}{\mathrm{length}(\boldsymbol{b})}. \tag{6}$$

Scattered crossover operation is used in our implementation. For two parents from the mating pool, each parent has equal chance to pass its gene to its child at each position. In the mutation step, for a parent selected for mutation, each position has the probability of p_{m} to be chosen to have 0-1 flip. Suppose the portion of 0s and 1s in this parent are p_0 and p_1, respectively. And suppose a position is chosen to mutate. If the value at this position is 1(0), it has the probability of $p_0(p_1)$ to be 0 (1). In this way, we can keep the child has the similar 0-1 portions as its parent.

Classification. If the gene subsets $\boldsymbol{G}$, found by MOEA, are used to predict the class labels of new samples, different prediction accuracies may be obtained. We need to select some gene subsets with the best generalization from $\boldsymbol{G}$. In order to do this, we use $\{\boldsymbol{D}_{tr}^{tr}, \boldsymbol{D}_{tr}^{te}\}$ to train a linear SVM classifier for any gene subset from $\boldsymbol{G}$, respectively, and use $\boldsymbol{D}_{tr}^{val}$ to test the classifier. The validation accuracy is used to decide the generalization of a gene subset. The best gene subsets with respect to generalization form a *gene subset committee*. If we use the gene subset committee to train respective linear SVM classifiers over $\boldsymbol{D}$, we can obtain a *classifier committee*, the class label of a new sample (independent with $\boldsymbol{D}$) is voted by the committee.

2.2 Revised SVM-RFE-mRMR

For the purpose of application and comparison, we revised the SVM-RFE-mRMR method to solve the weaknesses (except the weighting problem) as discussed in Section 1. [8] and [11] only described the gene ranking step, which is actually incomplete, we therefore append the validation step to find the best gene subset. See Algorithm 2 for details.

3 Experiments

We use three well-cited gene-sample datasets in our experiment. See Table 1 for details. We did two experiments.

First, we used 10-fold *cross-validation* (CV) to evaluate the performance of the MOEA based framework, and compared it with the revised SVM-RFE-mRMR. The experiment procedures are shown in Fig. 1 and 2. During each fold of CV of the MOEA

Algorithm 2. *Revised SVM-RFE-mRMR Gene Subset Selection*

```
Input: $\boldsymbol{D}$, of size $m$(genes) $\times$ $n$(samples), and the class labels $\boldsymbol{c}$
Output: selected gene subset $\boldsymbol{g}$, the best validation accuracy $av$,and list of survived genes $\boldsymbol{f}$
  split $\boldsymbol{D}$ into training set $\boldsymbol{D}_{\text{tr}}$ and validation set $\boldsymbol{D}_{\text{val}}$
  filter out the genes over $\boldsymbol{D}_{\text{tr}}$, and get gene list $\boldsymbol{f}$ left
  $\boldsymbol{D}_{\text{tr}} = \boldsymbol{D}_{\text{tr}}(\boldsymbol{f}, :)$
  $\boldsymbol{D}_{\text{val}} = \boldsymbol{D}_{\text{val}}(\boldsymbol{f}, :)$
  ——————gene ranking step——————
  set $\beta$
  given set of genes $\boldsymbol{s}$ initially including by all genes
  ranked set of genes, $\boldsymbol{r} = \{\}$
  repeat
    train linear SVM over $\boldsymbol{D}_{\text{tr}}$ with gene set $\boldsymbol{s}$
    calculate the weight of each gene $\boldsymbol{w}_i$
    for each gene $i \in \boldsymbol{s}$ do
      compute class relevance $R_{\boldsymbol{s},i}$ and feature redundancy $Q_{\boldsymbol{s},i}$ over $\boldsymbol{D}_{\text{tr}}$
      compute $r_i = \beta|\boldsymbol{w}_i| + (1-\beta)\frac{R_{\boldsymbol{s},i}}{Q_{\boldsymbol{s},i}}$
    end for
    select the gene with smallest ranking score, $i^* = \arg\min\{r_i\}$
    update $\boldsymbol{r} = \boldsymbol{r} \cup \{i^*\}$; $\boldsymbol{s} = \boldsymbol{s} \setminus \{i^*\}$
  until all gene are ranked
  ——————validation step——————
  $\boldsymbol{g} = \{\}$
  set the best validation accuracy $av = 0$
  for i=length($\boldsymbol{r}$) to 1 do
    $\boldsymbol{s} = \boldsymbol{s} \cup \{r_i\}$
    train linear SVM classifier over $\boldsymbol{D}_{\text{tr}}$
    obtain the validation accuracy $a$ through validating the classifier over $\boldsymbol{D}_{\text{val}}$
    if $av \leq a$ then
      if $av < a$ then
        $\boldsymbol{g} = \boldsymbol{s}$
      end if
      if $av == 1$ then
        break
      end if
    else
      break
    end if
  end for
```

Table 1. Gene-Sample Datasets

Dataset	#Classes	#Genes	#Samples
Leukemia [3, 20]	2	5000	27+11=38
CNC [3, 21]	2	5893	25+9=34
Colon [22]	2	2000	40+22=62

based method, the whole data $\boldsymbol{O}$ is partitioned into training set $\boldsymbol{O}_{\text{tr}}$ and test set $\boldsymbol{O}_{\text{te}}$. Algorithm 1 is employed to find the selected gene subsets $\boldsymbol{G}$, the best validation accuracy av and its corresponding gene subsets $\boldsymbol{G}_{\text{b}}$, and the list of survived genes $\boldsymbol{f}$. After that the best prediction accuracy and that using voting strategy, depicted in Section 2.1, are obtained. The linear SVM classifier is used in the classification step. After 10-fold CV, these two measures are averaged. We designed the same experiment procedure for SVM-RFE-mRMR method. Since this method only returns a gene subset in each fold, the prediction accuracies of the 10 gene subsets are averaged at the end of CV.

The experiment results are shown in Table 2. The "Pred. Acc." column shows the prediction accuracies. For the MOEA based approach, the values outside the parenthesis are the average of the best prediction accuracies. We trained linear SVM classifiers over $\boldsymbol{O}_{\text{tr}}$ with different gene subsets, and used $\boldsymbol{O}_{\text{te}}$ to test these classifiers. The best prediction accuracy among them are reported at each fold. The values in the parenthesis are the prediction accuracies obtained by the voting method. From this column, we can see that our proposed MOEA based method works well and outperforms SVM-RFE-mRMR in terms of the prediction accuracy. The next column tells us that both our MOEA based method and SVM-RFE-mRMR can obtain a small number of genes. The last column shows the execution time of the whole procedure. We can find that, though using four fitness functions, the MOEA based method is much faster than the revised SVM-RFE-mRMR. Our experimental procedure can avoid *false high prediction accuracy problem* (FHPAP) which is the case that the reported accuracy is higher than the actual one. FHPAP occurs when the whole dataset is used to select features, after that the performance of the selection method is evaluated through dividing the whole dataset into training set and test set, for example FHPAP occurs in [6].

Table 2. Prediction Accuracy

Data	Method	Pred. Acc.	#Genes	Time
Leukemia	MOEA	1(0.9050)	18.1	1.7×10^4
	SVM-RFE-mRMR	0.9083	30.9	6.2×10^4
CNC	MOEA	0.9167(0.7417)	29.9	1.5×10^4
	SVM-RFE-mRMR	0.7417	39.8	6.2×10^4
Colon	The Proposed	0.9357(0.8429)	36.7	1.7505×10^4
	SVM-RFE-mRMR	0.7881	4.8	6.4×10^4

Second, we used the whole datasets as input of Algorithm 1 and 2, respectively to find gene subsets for devising decision system and future biological exploration. The result is shown in Table 3. From this table, we can see that the MOEA based approach can obtain better validation accuracy than the revised SVM-RFE-mRMR approach.

4 Discussion

It is still an open problem of how to choose the most promising one or more gene subsets,$\boldsymbol{G}_b$, from the gene subsets, $\boldsymbol{G}$, returned by MOEA. We propose to find $\boldsymbol{G}_b$ according to the validation accuracy. Since $\boldsymbol{G}_b$ may contains more than one gene subsets,

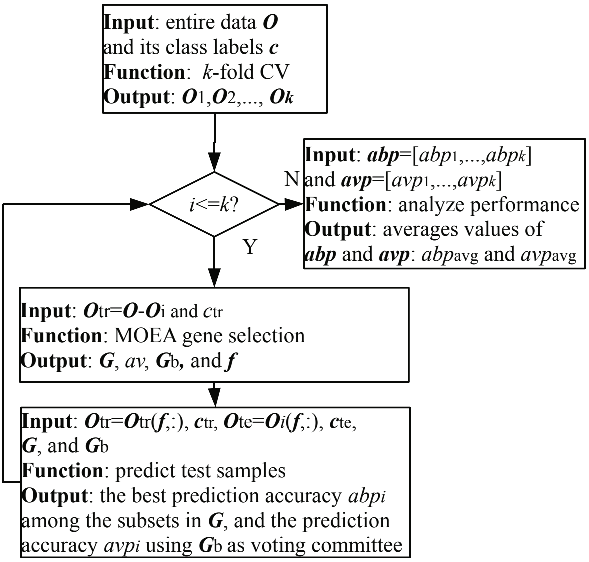
Input: entire data O and its class labels c
Function: k-fold CV
Output: O1,O2,..., Ok
i<=k?
N
Y
Input: abp=[abp1,...,abpk] and avp=[avp1,...,avpk]
Function: analyze performance
Output: averages values of abp and avp: abpavg and avpavg
Input: Otr=O-Oi and ctr
Function: MOEA gene selection
Output: G, av, Gb, and f
Input: Otr=Otr(f,:), ctr, Ote=Oi(f,:), cte, G, and Gb
Function: predict test samples
Output: the best prediction accuracy abpi among the subsets in G, and the prediction accuracy avpi using Gb as voting committee

Fig. 1. Procedure of Evaluating the MOEA Gene Selection

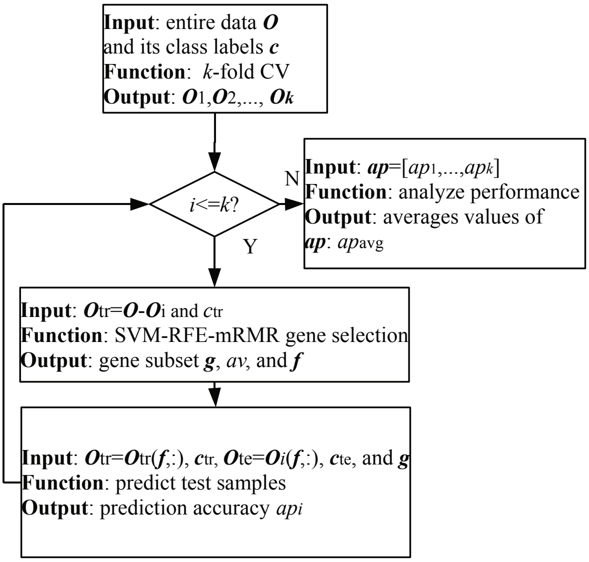
Input: entire data O and its class labels c
Function: k-fold CV
Output: O1,O2,..., Ok
i<=k?
N
Y
Input: ap=[ap1,...,apk]
Function: analyze performance
Output: averages values of ap: apavg
Input: Otr=O-Oi and ctr
Function: SVM-RFE-mRMR gene selection
Output: gene subset g, av, and f
Input: Otr=Otr(f,:), ctr, Ote=Oi(f,:), cte, and g
Function: predict test samples
Output: prediction accuracy api

Fig. 2. Procedure of Evaluating the Revised SVM-RFE-mRMR Gene Selection

a voting strategy can be used to determine the class labels of the new coming samples. [18] has a general discussion on this issue. Domain knowledge should be consider

Table 3. Validation Accuracy

Data	Method	Valid. Acc.	#Genes
Leukemia	MOEA	1	14.2
	SVM-RFE-mRMR	1	1
CNC	MOEA	1	24.9
	SVM-RFE-mRMR	0.8182	1
Colon	MOEA	1	19.5
	SVM-RFE-mRMR	0.8571	2

to select the best point on the Pareto front for specific application. Therefore, more thought should be inspired to discover the most discriminative gene subsets from $\boldsymbol{G}$.

After designing a feature selection method, two steps have to be followed. The first step aims to computationally evaluate the performance of the designed method, and to compare with other existing methods. This requires two substeps: training substep and test substep. Note that the generalized definition of training substep should include both feature selection and training a classifier. The working data should be split into two exclusive parts: the training set and test set (perhaps by cross-validation). If we need to decide some superparameters of the feature selection model, we need to further split the training set into training subset and validation subset. The superparameters could be, for example, the parameter of a scoring function, the size of the feature subset, or the best feature subset if the feature method returns more than one feature subsets. During training, we need to estimate the superparameters. For example, if we need to decide the best size of gene subsets, we need to train a classifier by the training subset, and validate its accuracy. If the validation accuracy is not satisfactory, we need to adjust the size of gene subsets, and repeat until we find the proper size. Once the proper superparameters are found, the training set, including both of the training subset and validation subset, is used to train a classifier. In the test substep, the prediction accuracy is obtained to measure the classification performance of the designed feature selection method, and to compare with other benchmark methods. It is unnecessary to report any feature subset selected, because the main task of this step should be evaluating the performance of a method.

After the first step, the confidence about the designed feature selection method is obtained. The next step is to use the whole dataset to select a gene subset, train a classifier, and wait for predicting new samples whose class labels are unknown. At this step, only the validation accuracy can be obtained if there are superparameters to optimize. However, there is no prediction accuracy to report, because the class labels are unknown. When optimizing the superparameters, the whole data can be divided into training set and validation set. The feature selection runs over the training set, while the validation set is used to adjust the superparameters of the feature selection method according to its output. After the promising superparameters are obtained, the whole dataset with the selected feature subset is used to learn a classifier. If the feature subset needs to be reported, the feature selection method should take the whole data as input. There is no need to worry about the quality of the reported feature subset, because the confidence of its quality comes from the first step. Furthermore, the prediction accuracy of the reported feature subset is expected higher than the prediction accuracy at the first step. The reason is that the reported feature subset uses larger number of samples at the second step.

Some researchers may mixed up the above two steps. For example, the whole dataset is firstly preprocessed and used to select the feature subset (this is actually the task of the second step), and then k-fold CV is employed to split the whole dataset into training sets and test sets. And the training set with the selected feature subset is used to learn a classifier; after that, predicted accuracy is reported through testing the classifier by the test set. Unfortunately, the prediction accuracy is overestimated because the test set has already been used during feature selection. If a sensitive feature selection method is subject to overfitting easily, then the prediction accuracy would be overestimated significantly. Also, some researchers may be wondering how to report the feature subset because they have k feature subsets from k-fold CV, respectively. The issue here is that they try to report the feature subset right after the first step. If the feature subset is reported at the second step, this issue can be avoided.

5 Conclusion and Future Works

This paper proposes a MOEA based framework to select gene subsets. This approach mainly includes a NMF based gene filtering method, a SVM based sample selection method, and a MOEA search strategy. We revise the SVM-RFE-mRMR method for comparison. Our approach overcomes the drawback of the mRMR and the revised SVM-RFE-mRMR methods. Experimental results show that the MOEA based approach outperforms the revised SVM-RFE-mRMR method. We also clarify some experimental issues when estimating designed feature selection methods. Since MOEA outputs more than one gene subsets, our future research will focus on finding better methods to identify the best gene subset after running MOEA. The biological relevance of the genes selected will be investigated as well.

Acknowledgments. This research has been supported by IEEE CIS Walter Karplus Summer Research Grant 2010, Ontario Graduate Scholarship 2011-2012, and Canadian NSERC Grants #RGPIN228117-2011.

References

1. Zhang, A.: Advanced Analysis of Gene Expression Microarray Data. World Scientific, Singapore (2009)
2. Li, Y., Ngom, A.: Non-Negative Matrix and Tensor Factorization Based Classification of Clinical Microarray Gene Expression Data. In: BIBM, pp. 438–443. IEEE Press, New York (2010)
3. Brunet, J.P., Tamayo, P., Golub, T.R., Mesirov, J.P.: Metagenes and Molecular Pattern Discovery Using Matrix Factorization. PNAS 101(12), 4164–4169 (2004)
4. Lee, D.D., Seung, S.: Learning the Parts of Objects by Non-Negative Matrix Factorization. Nature 401, 788–791 (1999)

5. Saeys, Y., Inza, I., Larrañaga, P.: A Review of Feature Selection Techniques in Bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
6. Ding, C., Peng, H.: Munimun Redundancy Feature Selection from Microarray Gene Expression Data. Journal of Bioinformatics and Computational Biology 3(2), 185–205 (2005)
7. Peng, H., Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
8. Guyon, I., Weston, J., Barnhill, S.: Gene Selection for Cancer Classification Using Support Vector Machines. Machine Learning 46, 389–422 (2002)
9. Mundra, P.A., Rajapakse, J.C.: Gene and Sample Selection for Cancer Classification with Support Vectors Based t-statistic. Neurocomputing 73(13-15), 2353–2362 (2010)
10. Mundra, P.A., Rajapakse, J.C.: Support Vectors Based Correlation Coefficient for Gene and Sample Selection in Ccancer Classification. In: CIBCB, pp. 88–94. IEEE Press, New York (2010)
11. Mundra, P.A., Rajapakse, J.C.: SVM-RFE with MRMR Filter for Gene Selection. IEEE Transactions on Nanobioscience 9(1), 31–37 (2010)
12. Liu, J., Iba, H.: Selecting Informative Genes Using A Multiobjective Evolutionary Algorithm. In: CEC, vol. 1, pp. 297–302. IEEE Press, New York (2002)
13. Paul, T.K., Iba, H.: Selection of The Most Useful Subset of Genes for Gene Expression-Based Classification. In: CEC, vol. 2, pp. 2076 - 2083. IEEE Press, New York (2004)
14. Kohane, I.S., Kho, A.T., Butte, A.J.: Microarrays for An Integrative Genomics. MIT Press, Cambridge (2003)
15. Kim, H., Park, H.: Sparse Non-Negatice Matrix Factorization via Alternating Non-Negative-Constrained Least Squares for Microarray Data Analysis. Bioinformatics 23(12), 1495–1502 (2007)
16. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Berlin (1995)
17. Chang, C., Lin, C.: LIBSVM : A Library for Support Vector Machines. ACM Transactions on Intelligent Systems and Technology 2(2), 27:1–27:27 (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
18. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithm. Wiley, West Sussex (2001)
19. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
20. Golub, T.R., Slonim, D.K., Tamayo, P., et al.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science 286(15), 531–537 (1999),
`http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi`
21. Pomeroy, S.L., Tamayo, P., Gaasenbeek, M., et al.: Prediction of Central Nervous System Embryonal Tumour Outcome Based on Gene Expression. Nature 415, 436–442 (2002), Data Available at
`http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi`
22. Alon, U., Barkai, N., Notterman, D.A., et al.: Broad Patterns of Gene Expression Revealed by Clustering of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. PNAS 96(12), 6745–6750 (1999), Data Available at
`http://genomics-pubs.princeton.edu/oncology`