Computers in Biology and Medicine 62 (2015) 14–24

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

Improving PLS–RFE based gene selection for microarray data classification

Aiguo Wang [a], Ning An [a,*], Guilin Chen [b], Lian Li [a], Gil Alterovitz [c,d,e]

[a] School of Computer and Information, Hefei University of Technology, Hefei, China
[b] School of Computer and Information Engineering, Chuzhou University, Chuzhou, China
[c] Center for Biomedical Informatics, Harvard Medical School, Boston, USA
[d] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, USA
[e] Children's Hospital Informatics Program at the Harvard/MIT Division of Health Sciences and Technology, Boston, USA

ARTICLE INFO

Article history:
Received 29 November 2014
Accepted 8 April 2015

Keywords:
Gene selection
Classification
Partial least squares
Recursive feature elimination
Sequential backward selection
Annealing schedule

ABSTRACT

Gene selection plays a crucial role in constructing efficient classifiers for microarray data classification, since microarray data is characterized by high dimensionality and small sample sizes and contains irrelevant and redundant genes. In practical use, partial least squares-based gene selection approaches can obtain gene subsets of good qualities, but are considerably time-consuming. In this paper, we propose to integrate partial least squares based recursive feature elimination (PLS–RFE) with two feature elimination schemes: *simulated annealing* and *square root*, respectively, to speed up the feature selection process. Inspired from the strategy of annealing schedule, the two proposed approaches eliminate a number of features rather than one least informative feature during each iteration and the number of removed features decreases as the iteration proceeds. To verify the effectiveness and efficiency of the proposed approaches, we perform extensive experiments on six publicly available microarray data with three typical classifiers, including Naïve Bayes, K-Nearest-Neighbor and Support Vector Machine, and compare our approaches with ReliefF, PLS and PLS–RFE feature selectors in terms of classification accuracy and running time. Experimental results demonstrate that the two proposed approaches accelerate the feature selection process impressively without degrading the classification accuracy and obtain more compact feature subsets for both two-category and multi-category problems. Further experimental comparisons in feature subset consistency show that the proposed approach with *simulated annealing* scheme not only has better time performance, but also obtains slightly better feature subset consistency than the one with *square root* scheme.

1. Introduction

The rapid development and wide use of microarray technology in biomedical research facilitates the high throughput monitoring and measurement of thousands of gene expression profiles simultaneously, and enables their meaningful applications in the diagnosis of cancers, the classification of tumor subtypes and the discovery of drug targets at the molecular level [1–3]. Although various classifiers are constructed and used for the classification of gene expression profiles, however, it has been shown that conventional machine learning and statistical techniques have drawbacks in achieving satisfactory classification performance due to the intrinsic nature of microarray data characterized by high dimensionality (as many as thousands of genes) and small sample sizes (as few as tens of samples) [4]. First, in the classification of microarray data, the number of genes far exceeds the number of samples, then "curse of dimensionality" occurs and may lead to a classification model with poor generalization capability and weak robustness. Second, a minimum of $10*p*C$ training samples are expected to produce a pre-determined level of performance for a p-dimensional classification problem of C classes [5], whereas it is often not practical to obtain corresponding number of microarray samples in actual use due to the high cost of experiments [6]. Third, the gene space that is involved in microarray data can have noisy and irrelevant genes and often generates a classifier that has poor classification performance [7]. One method to cope with these problems is to conduct dimensionality reduction on the gene space by removing noisy, irrelevant and redundant genes from the original gene space using effective gene selection methods [8,9].

[*] Correspondence to: Hefei Tunxi Road 193, Hefei University of Technology, Hefei 230009, China. Tel.: +86 180 1995 6086; fax: +86 551 6290 4642.
E-mail addresses: wangaiguo2546@163.com (A. Wang), ning.g.an@acm.org (N. An), glchen@chzu.edu.cn (G. Chen), llian@hfut.edu.cn (L. Li), gil_alterovitz@hms.harvard.edu (G. Alterovitz).

http://dx.doi.org/10.1016/j.compbiomed.2015.04.011
0010-4825/© 2015 Elsevier Ltd. All rights reserved.

Feature selection, which is also called gene selection in the context of microarray data, is defined as the process of identifying a small subset of features that contains the most discriminative information from the original feature space to improve the classification performance [10,11]. According to whether a classifier is used to evaluate the goodness of candidate features in the feature selection process, feature selection techniques are typically divided into three categories: filter methods, wrapper methods and embedded methods [12]. Filter methods evaluate the quality of a feature using the intrinsic properties of training samples, so they have a lower computational complexity and better generalization ability and are flexible in combination with various classifiers [13,14]. In contrast to filter methods, wrapper methods are tightly coupled with a classifier to evaluate the quality of a candidate feature and often use the classification error rate as the evaluation criterion [12]. Benefited from the fine-tuned interaction between the classifier and training samples, wrapper methods tend to obtain better classification performance than the filter methods [15,16]. There is a deeper interaction in embedded methods between feature selection and the construction of a classifier, because the feature selection process is integrated into the construction of the classifier, which makes embedded methods more computationally tractable than the wrapper methods. Typical embedded methods include decision tree algorithms C4.5 and random forest [17]. Obviously, enumerating all of the possible combinations of feature subsets and evaluating them one by one guarantees obtaining the globally optimal feature subset, while the computational complexity increases exponentially $O(2^N)$ with the number of features N [18]. This approach is often unacceptable because of having such a high time complexity in an actual application, especially in the case of gene expression profiles that have thousands of genes. To achieve the trade-off between computational efficiency and classification accuracy, researchers have proposed a variety of feature subset generation methods, in which feature ranking is an effective and efficient method to select an optimal or near-optimal feature subset and can obtain good results comparable to the globally optimal feature selection methods [19,20].

Partial least squares (PLS) is a non-parametric, multivariate statistical analysis technique. Since PLS can capture the interaction of feature–feature, it has been extensively used to do dimensionality reduction and high-dimensional data analysis [21]. In feature selection, PLS can rank all the features through calculating the projection importance of each feature, and select these features that are ranked in the top [22,23]. A good feature ranking criterion is not necessarily a good feature subset ranking criterion, because the selected top-ranked features may be redundant to each other. To obtain a high-quality feature subset, researchers propose to combine the recursive feature elimination (RFE) search strategy with PLS, called PLS–RFE, to evaluate the goodness of a feature and then generate a ranking of all the features. RFE is a specific sequential backward selection (SBS) scheme that starts from the whole feature set and eliminates one least informative feature progressively until a pre-defined number of features are obtained [24]. In handling high-dimensional data, PLS–RFE not only achieves comparable or better classification performance and more compact gene subsets, but also has better generalization ability and computational efficiency in comparison with the state-of-the-art methods [21,22]. However, when the number of features is large, it would require a large amount of CPU time to rank all the features in the case of microarray data. Therefore, accelerating this process without degrading the high classification performance would be of great value for gene expression analysis.

In this paper, we propose to integrate partial least squares based recursive feature elimination with two elimination schemes: *simulated annealing* (PLS–RFE-SA) and *square root* (PLS–RFE-SQRT), respectively, to speed up the feature selection process. Inspired from the strategy of annealing schedule [25,26], the two proposed approaches eliminate a dynamic number of features rather than a constant number of features during each iteration. In addition, a larger number of features are eliminated at the beginning and a smaller number of features are eliminated as the iteration proceeds. Since most of the removed features are less relevant to the target class in the initial iterations and the remaining features are more relevant to the target class in later stages, this dynamic process is feasible and reasonable and expected to obtain desirable performance.

The rest of this paper is organized in the following way. Basic theories of partial least squares and partial least squares-based recursive feature elimination are illustrated in Section 2. Section 3 details the two improved PLS–RFE approaches, followed by the evaluation measures illustrated in Section 4. In Section 5, we experimentally evaluate our proposed approaches on six publicly available microarray data with three commonly used classifiers in terms of classification accuracy, time cost to rank the features as well as the feature subset consistency, and further compare the proposed approaches with three other well-performing feature selectors including PLS–RFE, PLS and ReliefF. The last section concludes this paper with a brief summary and discussion.

2. Partial least squares-based feature selection

2.1. Partial least squares

Partial least squares (PLS) is a non-parametric, multivariate statistical analysis technique. Since PLS can capture the relationship among features, it can eliminate the effects of multicollinearity between features through finding a linear regression model that projects the explanatory variables (features) and predicted variable (target class) to a new space [21]. In addition, PLS also considers the correlation between explanatory variables and predicted variable when conducting dimensionality reduction. In practical use, PLS has proven to be effective in handling situations where the number of features is significantly greater than the number of samples, i.e. the situation of high dimensionality and small sample sizes [21,22,27].

Let $X = \left[F_1, F_2, \ldots, F_p\right] = [X_1, X_2, \ldots, X_n]^T$ be the $n*p$ matrix with n samples and p features/genes, and $Y = [L_1, L_2, \ldots, L_q] = [Y_1, Y_2, \ldots, Y_n]^T$ be the $n*q$ matrix with n samples and q classes. Explanatory variable X and predicted variable Y are both normalized to be zero mean and one standard deviation for each column before PLS is used. PLS establishes its solutions by finding a pair of projection directions w and c so that $t=X*w$ and $s=Y*c$ meet the following two criteria: (1) t and s contain as much variation information of X and Y as possible and (2) maximizing the correlation coefficient between t and s [21]. Combining the criteria (1) and (2), PLS is required to maximize the covariance of t and s

$$\max \{Cov(t,s)\} = \max \{\sqrt{var(t)var(s)} * r(t,s)\} \tag{1}$$

According to the statistical theories, we can derive the following equation:

$$Cov(t,s) = E(t,s) = w^T E\left(XY^T\right)c = w^T S_{XY} c \tag{2}$$

where S_{XY} is the covariance matrix of X and Y. The PLS projection directions w and c, therefore, can be obtained by maximizing the following formula under the conditions of $w^T w = 1$ and $c^T c = 1$:

$$\underset{w,c}{\arg\max} \left\{Cov(t,s)\right\} = \underset{w,c}{\arg\max} \frac{w^T S_{XY} c}{\sqrt{(w^T w)(c^T c)}} \tag{3}$$

By projecting X on the direction of w and projecting Y on the direction of c, we can obtain the first pair of PLS components

$t_1 = X*w$ and $s_1 = Y*c$, and then establish three regression equations between Y and t_1, Y and s_1, X and t_1. This process terminates if the predetermined precision is achieved by the regression equations; otherwise, the second pair of PLS components t_2 and s_2 are extracted from the residuals. Repeat the above process until the halt condition is satisfied. Statistically Inspired Modification of PLS (SIMPLS) provides us an efficient solution $\{w_1, w_2, \ldots, w_{nfac}\} \in R^p$ and $\{c_1, c_2, \ldots, c_{nfac}\} \in R^q$ to PLS, in which $nfac$ is the number of factors of SIMPLS. The computational complexity of SIMPLS is $O(np)$ on a dataset with n samples and p features [21,28].

2.2. Partial least squares with recursive feature elimination

PLS extracts $t = [t_1, t_2, \ldots, t_h]$ components, which contain as much as variation information of X and Y. In feature selection, we need to analyze the impact of each explanatory variable X_i to Y and rank these explanatory variables according to a given criterion. Variable importance in projection (*VIP*) provides us a metric to quantitatively measure the impact [29]. Given the component t_h, its explanation to the predicted variable Y is

$$Rd(Y; t_h) = \frac{\sum_{i=1}^{q} r^2(Y_i, t_h)}{q} \tag{4}$$

where $r(Y_i, t_h)$ is the correlation coefficient between two random variables Y_i and t_h. We can calculate *VIP* of each X_i with all the components $t = [t_1, t_2, \ldots, t_h]$ using the following formula:

$$VIP(X_i) = \sqrt{p*\frac{\sum_{k=1}^{h} Rd(Y; t_k) w_{ik}^2}{\sum_{k=1}^{h} Rd(Y; t_k)}} \tag{5}$$

where w_{ik} is the ith weight of axis w_k weighting the marginal contribution of X_i to the component t_k. The interpretation of X_i to Y is through t_k and X_i plays an important role in determining t_k, so a strong explanatory power of t_k to Y indicates that the explanatory power of X_i to Y should be regarded to be significant and that larger $VIP(X_i)$ indicates more importance of X_i in interpreting Y. Therefore, *VIP* provides us a basis to rank the explanatory variables for feature selection.

A good feature ranking criterion is not necessarily a good feature subset ranking criterion. To select a feature subset of high quality, researchers propose to combine recursive feature elimination (RFE) with PLS (PLS–RFE) to evaluate the goodness of a feature and then rank all the features progressively [22]. RFE is a specific sequential backward selection scheme, and its main idea is to start with all features: (a) select the least discriminative feature based on a given criterion such as *VIP* in PLS; (b) eliminate it from the original feature space; and (c) repeat the above procedure until all the features are ranked or a predefined number of features are obtained [30]. Algorithm 1 presents the pseudo-code of PLS–RFE. PLS–RFE has experimentally proven to obtain better classification accuracy, stability and more compact gene subsets in comparison with other state-of-the-art feature selection methods in handling microarray data [22]. In addition, PLS–RFE exhibits better robustness to noises and generalization ability in predicting unseen samples.

Algorithm 1. PLS–RFE algorithm

```
Input: dataset $X_{n*p}$ with attributes $S$, encoded target class $Y_{n*q}$
   with $q$ classes
Output: the ranked gene set $R$
initialize $R=[\ ]$, and $nfac=q$;
while $|S| \geq nfac$ do
    $D=X(, S)$; //training set projected over $S$
    calculate $Rd$ and $W$ using SIMPLS($D$, $Y$, $nfac$);
    use Eq. (5) to compute $VIP$ of each gene in $S$;
    $VIP(X_f)=\min(VIP)$; //find the gene with minimal $VIP$
    $R=[S(f), R]$; //add it to $R$
    $S=S(1{:}f-1, f+1{:}|S|)$; //delete it from $S$
end
return $R=[S, R]$.
```

3. Improved PLS–RFE methods

Feature selection has proven to be a NP-hard problem and it is prohibitive to find the optimal feature subset by enumerating all the possible feature subset combinations or by adopting the branch and bound strategy [12,31]. In PLS–RFE, the sequential backward selection strategy is used to rank the features and select feature subsets of good qualities. Although PLS–RFE is efficient and has the time complexity $O(n*p)$ to generate a rank of features on a dataset with n samples and p features in each iteration [32], it would be considerably time-consuming if only one feature is eliminated during each iteration and the overall time complexity would be approximate $O(n*p^2)$. Obviously, PLS–RFE would take a large amount of CPU time to handle the situations where there are thousands of or tens of thousands of features. Therefore, speeding up the feature ranking process of PLS–RFE without degrading the high classification performance would be of great value for gene expression analysis. In contrast to the feature elimination strategy used in PLS–RFE, eliminating a number of features rather than only one least informative feature during each iteration seems to be a feasible approach. Then, how many features are to be eliminated during each iteration determines the quality of the final selected feature subset and is our main focus in this study.

In heuristic search methods, simulated annealing (SA) performs well in finding a good approximation to the global optimum in a large search space for a variety of combinatorial optimization problems [25]. Inspired from the process of annealing metallurgy, SA starts from a random state and conducts the search with a designed annealing schedule. Generally, in the early search stage, there is a high probability for SA to accept a move in the search space to a worse solution. As the search proceeds, the probability to accept a worse solution decreases and SA gradually converges to the approximately optimal solution.

Referring to the ideology of simulated annealing, we propose to integrate PLS–RFE with *simulated annealing* (PLS–RFE-SA) scheme to accelerate the feature selection process with the following strategies: (a) PLS–RFE-SA eliminates a large number of features from the candidate features in the initial iterations and (b) the number of eliminated features decreases as the iteration proceeds. In PLS–RFE-SA, we use such an annealing schedule that $(1/j+1)$ of the remaining features are eliminated from the candidate features in the jth iteration, which means that after the first iteration, half of the original features are eliminated, and the one-third of the remaining features are eliminated after the second iteration. Specially, if the number of remaining features is less than the value of the iteration counter, only one feature is eliminated. This enables PLS–RFE-SA to significantly reduce the computational cost of PLS–RFE in feature selection. Algorithm 2 presents the pseudo-code of the proposed PLS–RFE-SA (lines 6–10 represent the simulated annealing schedule). It should be noteworthy that although we name it as PLS–RFE-SA, it works in a quite different way from the classical simulated annealing algorithm, which accepts a worse solution during annealing. Actually, we utilize SA to control the decay rate and further determine the number of genes to be eliminated during each iteration, thus, we do not consider the case that unfavorable features could be accepted by random probability.

Likewise, we propose another feature selection approach which eliminates *square root* $\sqrt{|S|}$ features from the candidate features during each iteration ($|S|$ is number of remaining features before each iteration). We note this approach as PLS–RFE-SQRT, and the pseudo-code of PLS–RFE-SQRT can be obtained by substituting $|S|*(1/j+1)$ with $\sqrt{|S|}$ (line 6) in Algorithm 2.

Algorithm 2. PLS–RFE with simulated annealing

```
**Input:** data $X_{n*p}$ with attributes $S$, encoded target
  class $Y_{n*q}$with $q$ classes
**Output:** the ranked gene set $R$
((1):      initialize $R=[\ ]$, $nfac=q$, and $j=1$;
((2):      **while** $|S| \geq nfac$ **do**
((3):      $D=X(,S)$; //training set projected over $S$
((4):      calculate $Rd$ and $W$ using SIMPLS($D$, $Y$, $nfac$);
((5):      use Eq. (5) to compute $VIP$ of each gene in $S$;
((6):      **repeat** $|S|*(1/j+1)$ times **do** //eliminate a number
    of genes
((7):      $VIP(X_f)=\min(VIP)$; //find the gene with minimal
    $VIP$
((8):      $R=[S(f), R]$; //add it to $R$
((9):      $S=S(1{:}f-1, f+1{:}|S|)$; //delete it from $S$
((10):     end
((11):     $j=j+1$; //iteration counter
((12):     end
((13):     **return** $R=[S, R]$.
```

4. Evaluation measures

To evaluate the performance of PLS–RFE-SA and PLS–RFE-SQRT, we compare them with PLS–RFE in terms of classification accuracy and actual running time. According to the discussion in the previous section, both PLS–RFE-SA and PLS–RFE-SQRT are expected to speed up the feature selection process impressively. To evaluate the quality of a feature selection method, classification accuracy is a direct and effective criterion and is much more important in practical use, because a feature selector contributing less to the microarray data analysis is of little use [33]. Meanwhile, we compare them in terms of the size of the final selected feature subset that correspondingly achieves the best classification accuracy as well. In microarray data classification, selecting a compact gene subset is preferable for further gene analysis and biological validation and also indicates the power of a feature selection method in selecting informative genes [34].

Besides these, to measure the consistency of two feature subsets obtained by two different feature selection methods, feature subset consistency is used to measure the similarity between two feature subsets. In this study, a similarity-based approach is adopted to quantify the consistency. Given two feature subsets f_i and f_j of equal size generated by two different feature selection methods, Kuncheva put forward Kuncheva Index (KI) to measure the consistency between f_i and f_j using the following formula [35]:

$$KI\left(f_i,f_j\right) = \frac{r*(N-s^2)}{s*(N-s)} = \frac{r-(s^2/N)}{s-(s^2/N)} \tag{6}$$

where $s=|f_i|=|f_j|$ denotes the feature subset size, $|r=f_i \cap f_j|$ is the number of common features of the two subsets, N is the size of the original feature space, and s^2/N term is a correction term. The value of KI satisfies $-1<KI\left(f_i,f_j\right) \leq 1$, and a greater value of KI indicates that the two feature selection methods tend to select a larger number of common features. For simplicity and easy calculation, we use the following formula to calculate the consistency in our study:

$$KI\left(f_i,f_j\right) = \frac{|f_i \cap f_j|}{|fi\ |} \tag{7}$$

5. Experimental results and analysis

5.1. Experimental dataset and experimental setup

Experiments were conducted on six publicly available microarray datasets that cover both two-category and multi-category classification problems. A brief summary to the six datasets is presented in Table 1. The last column SFR denotes the ratio between the number of samples and the number of genes. From SFR, we can see that there exists a great imbalance between the number of samples and the number of genes in each microarray data. All the microarray datasets used in this study can be downloaded from http://www.gems-system.org/, and their brief descriptions are given in the following.

(1) *Brain* tumor dataset: *Brain* consists of 5 human brain tumor types, including medulloblastoma, malignant glioma, atypical teratoid/rhabdoid turmor (AT/RTs), neuroectodermal tumors (PNETS) and normal cerebellum [36]. *Brain* dataset contains 90 samples in total, consisting of 60 medulloblastoma samples, 10 malignant glioma samples, 10 AT/RT samples, 6 PNETs samples, and 4 normal cerebellum samples. Each sample includes 5920 genes. The task is to construct a classifier on this dataset and distinguish these five tumor types.

(2) *Leukemia*1 dataset: A collection of leukemia patient samples from bone marrow and peripheral blood is used for distinguishing between acute myeloid leukemia (AML) and acute lymphoma leukemia (ALL) tissues. This dataset contains 72 samples with 7129 genes: 25 AML samples and 47 ALL samples [2]. The classification task is to distinguish these two types of leukemia.

(3) *Leukemia*2 dataset: A collection of leukemia patient samples from bone marrow and peripheral blood is used for distinguishing between acute myeloid leukemia (AML) and acute lymphoma leukemia (ALL). The data for ALL is further divided in terms of B cells and T cells. The task is to construct a classifier to classify these three subtypes of leukemia according to the gene expression profiles. *Leukemia*2 consists of 72 samples with 5327 genes, and of these samples, 38 are of AML, 9 are of ALL-B and 25 are of ALL-T [2].

(4) Diffuse Large-B-Cell Lymphoma (*DLBCL*) dataset: Diffuse large B-cell lymphomas (BCL) and follicular lymphomas (FL) are two B-cell lineage malignancies [37]. There are 7129 genes with 58 BCL samples and 19 FL samples in the *DLBCL* data. The task on this dataset is to build a classification model to discriminate BCL from FL.

(5) *Prostate* cancer dataset: This dataset contains 50 non-tumor prostate samples and 52 prostate tumors, and each sample is described by 12,600 genes [38]. The classification task is to identify the expression patterns that correlate with the distinction of prostate tumor from the normal.

(6) *Ovarian* cancer dataset: There are 15,154 identities in this dataset. It contains 253 samples (162 ovarian samples and 91 controls). The goal of this experiment is to distinguish ovarian

Table 1
Experimental dataset description.

Dataset	#Genes	#Samples	#Classes	#SFR
Brain	5920	90 (60/10/10/4/6)	5	0.015
*Leukemia*1	7129	72 (47/25)	2	0.010
*Leukemia*2	5327	72 (38/9/25)	3	0.014
DLBCL	7129	77 (58/19)	2	0.011
Prostate	12,600	102 (50/52)	2	0.008
Ovarian	15,154	253 (91/162)	2	0.017

cancer from non-ovarian cancer using the proteomic spectra data [39].

(7) Given a microarray dataset $D=\{(X_i, y_i)\,|\,X_i \in X, y_i \in Y_l, 1 \le i \le n\}$ with n samples, $Y_l=\{l_1, l_{2,}\ldots, l_q\}$ is the class label set with q different classes. First, we are required to encode Y_l as $Y=(y_{ij})_{n*q} \in \{0,\ 1\}^{n*q}$ in the following way.

$$y_{ij} = I(y_i = l_j) = \begin{cases} 1 & y_i = l_j \\ & \\ 0 & \text{others} \end{cases},\ i=1,..,\ n;\ j=1,\ldots q. \tag{8}$$

Then, we can use SIMPLS algorithm to calculate the *VIP* of each gene. The only parameter in solving partial least squares is *nfac*, and the recommended empirical value of *nfac* is the number of classes for each microarray data, and it is used in our experiments [22].

5.2. Performance of classification

For microarray data that has high dimensionality and small sample sizes, to evaluate the quality of the finally selected feature subset, a ten-fold cross validation is commonly used to generate the independent training set and test set in order to obtain objective classification accuracy [40]. In the ten-fold cross validation scheme, each one of the ten folds is used as the test set and the remaining nine folds are used as the training set for the classifier construction, and the final classification accuracy is the average of the ten results [41]. Each of the feature selection algorithms is then applied on the training set to rank all the features and select the m top-ranked features. The classifier is trained on the training set projected over the selected features and the test set projected over the selected features is evaluated by the constructed classifier. On the top m features, classification accuracy and the size of the selected feature subset with maximum classification accuracy are recorded. To demonstrate the effectiveness of partial least squares-based feature selection methods, in our experimental study, besides PLS–RFE, and the proposed PLS–RFE-SA and PLS–RFE-SQRT, we also include two other well-performing feature selectors, ReliefF and PLS, as a comparison. PLS is a feature ranking approach and ranks features in descending order according to the value of *VIP* as we previously discussed in Section 2 [29,42]. The greater *VIP* of a feature it is, the more important it is in contributing to the classification performance. Being one of the classical distance-based filter measures for feature selection, ReliefF has great power in choosing discriminative features and good stability to the perturbation of training set, even if it fails to consider the redundancy among the selected features [43]. In our experiments, we use the default parameter values with 5 neighbors and 30 instances in ReliefF, and set *nfac* in PLS equal to the number of categories in each microarray data. Studies have suggested a few genes are sufficient in constructing an effective classifier for gene expression profile classification [34]. To find the best feature subset that can obtain the best classification accuracy, we consider the 50 top-ranked features and further search the best feature subset in it. Specifically, we first rank all the features in descending order according to a given criterion, and then generate feature subsets by picking the top m features sequentially, where m=1, 2,…, 50. In this way, we can obtain fifty feature subsets and construct classifiers on training set projected over each of the feature subsets. Then, the one that achieves the best classification accuracy corresponds to the best feature subset [8]. If two or more feature subsets produce equal classification accuracy, the one with smallest number of features is selected as the best feature subset.

To evaluate the quality of the finally selected feature subset, three commonly used classifiers with different metrics are used: Naïve Bayes (NB) [44], 3-Nearest-Neighbor (3NN) with Euclidean distance metric [45], and Support Vector Machine (SVM) with linear kernel and default parameter values [46]. Experimental comparisons are first conducted in terms of the best classification accuracy and corresponding number of selected genes. To show the effectiveness of the two proposed approaches (PLS–RFE-SA and PLS–RFE-SQRT), we take PLS-RFE as the baseline approach and use a Wilcoxon signed-rank test with a significance interval of 95% to determine whether there is any difference between the accuracy of PLS–RFE and the accuracies of other four feature selectors. In our experiments, the difference of accuracy is significantly different if its p-value is less than 0.05.

Tables 2–4 present the experimental results for NB, 3NN and SVM, respectively. For comparison, the last column "Unselected" presents the classification accuracy without using feature selection. The last but one row "AVE" shows the average classification accuracy and the average number of selected genes for each method. We also make a comparison between PLS–RFE and other methods at the aspect of Win/Tie/Loss. The last row "W/T/L" presents the number of times that the corresponding method is win/tie/loss in accuracy compared with PLS–RFE. In addition, notation "$*$" represents that the accuracy in the entry is significantly better than the corresponding one of PLS–RFE, and notation "\widehat" indicates that the accuracy in the entry is significantly worse than the corresponding one of PLS–RFE according to the Wilcoxon signed-rank test.

According to the results in Tables 2–4, we can observe that compared to the situation without using feature selection, all the feature selection methods greatly improve the classification accuracy and reduce the feature dimensionality, which demonstrates the effectiveness of gene selection methods in classifying microarray data. Specifically, for the case of NB, the average accuracy of PLS–RFE-SA and PLS–RFE-SQRT on the all the experimental datasets are 96.76% and 96.67%, respectively, which are comparable to 96.55% of PLS–RFE and outperform 92.74% of PLS and 95.52% of ReliefF. From the entry "W/T/L", it is noted that the PLS–RFE-SA and PLS–RFE-SQRT achieve comparable accuracy to PLS–RFE, and that PLS and ReliefF are inferior to PLS–RFE in the majority of datasets. For instance, the entry "2/3/1" in Table 2 denotes that PLS–RFE-SA wins 2 cases, ties 3 cases and loses 1 case in comparison with PLS–RFE, and entry "2/2/2" means that PLS–RFE-SQRT wins 2 cases, ties 2 cases and loses

Table 2
Experimental results with Naïve Bayes.

Dataset	PLS–RFE		PLS–RFE-SQRT		PLS–RFE-SA		PLS		ReliefF		Unselect
Brain	90.44	13	90.00	45	91.31	48	76.97\widehat	50	84.72\widehat	45	87.67
*Leukemia*1	98.75	14	98.75	8	98.75	14	97.32	9	97.32\widehat	9	98.57
*Leukemia*2	100.00	26	100.00	22	100.00	17	96.07\widehat	40	98.57	37	97.33
DLBCL	95.00	21	96.07	30	95.00	8	91.00\widehat	38	97.50	7	80.72
Prostate	95.09	15	96.09	23	95.27	12	96.18	42	96.18	43	63.00
Ovarian	100.00	21	99.62	24	99.62	22	98.82	46	98.83	44	92.49
AVE	96.55	18	96.76	25	96.67	20	92.74	38	95.52	31	86.63
W/T/L	–		2/2/2		2/3/1		1/0/5		2/0/4		0/0/6

Table 3
Experimental results with 3-nearest-neighbor.

Dataset	PLS–RFE		PLS–RFE-SQRT		PLS–RFE-SA		PLS		ReliefF		Unselect
Brain	92.56	33	92.67	30	92.67	29	80.53\widehat	39	86.00\widehat	35	89.53
*Leukemia*1	100.00	12	98.75	7	100.00	12	98.75	50	96.07\widehat	47	90.36
*Leukemia*2	100.00	39	100.00	36	100.00	34	97.32\widehat	38	98.57	23	89.23
DLBCL	100.00	16	100.00	11	100.00	15	94.82\widehat	43	98.75\widehat	40	85.48
Prostate	98.09	42	98.09	17	98.00	12	97.00\widehat	15	94.18\widehat	11	81.46
Ovarian	100.00	6	100.00	14	100.00	6	100.00	38	98.85	30	94.08
AVE	98.44	25	98.25	19	98.45	18	94.74	37	95.40	31	88.36
W/T/L	–		1/4/1		1/4/1		0/1/5		0/0/6		0/0/6

Table 4
Experimental results with support vector machine.

Dataset	PLS–RFE		PLS–RFE-SQRT		PLS–RFE-SA		PLS		ReliefF		Unselect
Brain	97.78	33	96.67	24	96.67	27	82.22\widehat	41	85.56\widehat	37	85.56
*Leukemia*1	100.00	16	98.61	6	100.00	16	97.22\widehat	44	97.22\widehat	30	95.83
*Leukemia*2	98.61	8	98.61	8	98.61	8	97.22	38	98.61	5	93.06
DLBCL	98.70	13	98.70	7	98.70	13	96.10\widehat	42	98.70	39	97.40
Prostate	98.04	11	98.04	15	98.04	11	96.08\widehat	14	97.06\widehat	5	91.18
Ovarian	100.00	6	100.00	10	100.00	6	100.00	36	99.61	22	100.00
AVE	98.86	15	98.44	12	98.67	14	94.81	36	96.13	23	93.84
W/T/L	–		0/4/2		0/5/1		0/1/5		0/2/4		0/1/5

2 cases compared with PLS–RFE. In addition, we can see that both PLS–RFE-SA and PLS–RFE-SQRT achieve high accuracies, which are not significantly different from that of PLS–RFE (Wilcoxon signed-rank test with the 95% confidence interval). Furthermore, the size of the final selected feature subsets of PLS–RFE-SA and PLS–RFE-SQRT are comparable to that of PLS–RFE, and are much smaller than these of PLS and ReliefF, which indicate that the two proposed approaches are able to select a compact subset of discriminative features. For the case of 3NN and SVM, similar conclusions can be drawn from Tables 3 and 4, respectively.

Additionally, it is observed from Tables 2–4 that the selected feature subset exhibits inconsistent performances for different classifiers on the same microarray dataset. To reduce the bias of a feature subset evaluation based on a specific classifier, we calculate the mean classification accuracy of the three classifiers for each of the feature selection methods. Fig. 1 presents the experimental results, where *X*-axis refers to the number of selected features and *Y*-axis represents the mean accuracy of the three classifiers on corresponding feature subsets. From Fig. 1, we can observe that both PLS–RFE-SA and PLS–RFE-SQRT achieve comparable average accuracy to PLS–RFE on all the microarray datasets and consistently outperform PLS and ReliefF, which further demonstrates the effectiveness of the two proposed approaches and their robustness to the choice of classifiers.

5.3. *Time cost comparsion*

In the previous section, it is concluded that both PLS–RFE-SA and PLS–RFE-SQRT not only achieve comparable classification performance to PLS–RFE, but also outperform PLS and ReliefF in classification accuracy and the size of selected feature subsets. In this section, we investigate the computational cost of the two proposed approaches to generate a rank of the original features and compare them with PLS–RFE. All the algorithms are implemented with matlab programing language, and experiments are conducted on a Quad-core Intel Core i5 CPU (3.2 GHz processor and 4 G RAM). Table 5 presents the time cost comparison of the three approaches on the six microarray datasets. In each cell, the time(s) cost is followed by the number of iterations that are required to rank all the features.

From Table 5, one can observe that there are significant differences in the time cost of the three approaches, and that PLS–RFE-SA is much faster than the other two and PLS–RFE-SQRT is much faster than PLS–RFE on the all the experimental datasets. Even on the small dataset *Leukemia*2 with 72 samples and 5327 genes in each sample, it takes PLS–RFE 2.9 h and PLS–RFE-SQRT 187.5 s to rank all the features, while PLS–RFE-SA only consumes 19.5 s, which is about 530 times faster than PLS–RFE and 10 times faster than PLS–RFE-SQRT. For the dataset with a larger number of genes, the difference is much more significant. For instance, the time cost on *Ovarian* is about 10.8 h for PLS–RFE, 429.5 s for PLS–RFE-SQRT and only 31.4 s for PLS–RFE-SA.

Furthermore, we take *Leukemia*2, *DLBCL*, *Prostate* and *Ovarian* as examples to show the relationship between the number of iterations and the number of remaining features. Fig. 2 presents the iteration process curves of PLS–RFE-SA and PLS–RFE-SQRT on the four datasets. Since PLS–RFE eliminates one feature in each iteration, it is a straight line with 135 degree tilt angle and would take large space in the figure. Hence, the curve of PLS–RFE is not drawn in Fig. 2. The *X*-axis refers to the number of iterations and *Y*-axis refers to the corresponding number of remaining features after each iteration. From Fig. 2, one can observe that PLS–RFE-SA eliminates a large number of features in the initial iterations, and then removes a smaller number of features as the iteration proceeds. In contrast to PLS–RFE-SA, PLS–RFE-SQRT eliminates a smaller number of features in each iteration and converges slower.

The time complexity of SIMPLS to calculate *VIP* on a dataset with n samples and p features is O(n*p), so the computational cost of the partial least squares based approaches is mainly determined by: (a) the total number of iterations required to rank all the features and (b)the number of features in running SIMPLS. Because the proposed approach with *simulated annealing* scheme has a faster decay rate and convergence rate than the one with *square root* scheme, therefore, this theoretically explains why PLS–RFE-SA outperforms both PLS–RFE and PLS–RFE-SQRT in terms of running time, which is also experimentally supported by the results in Table 5 and Fig. 2.

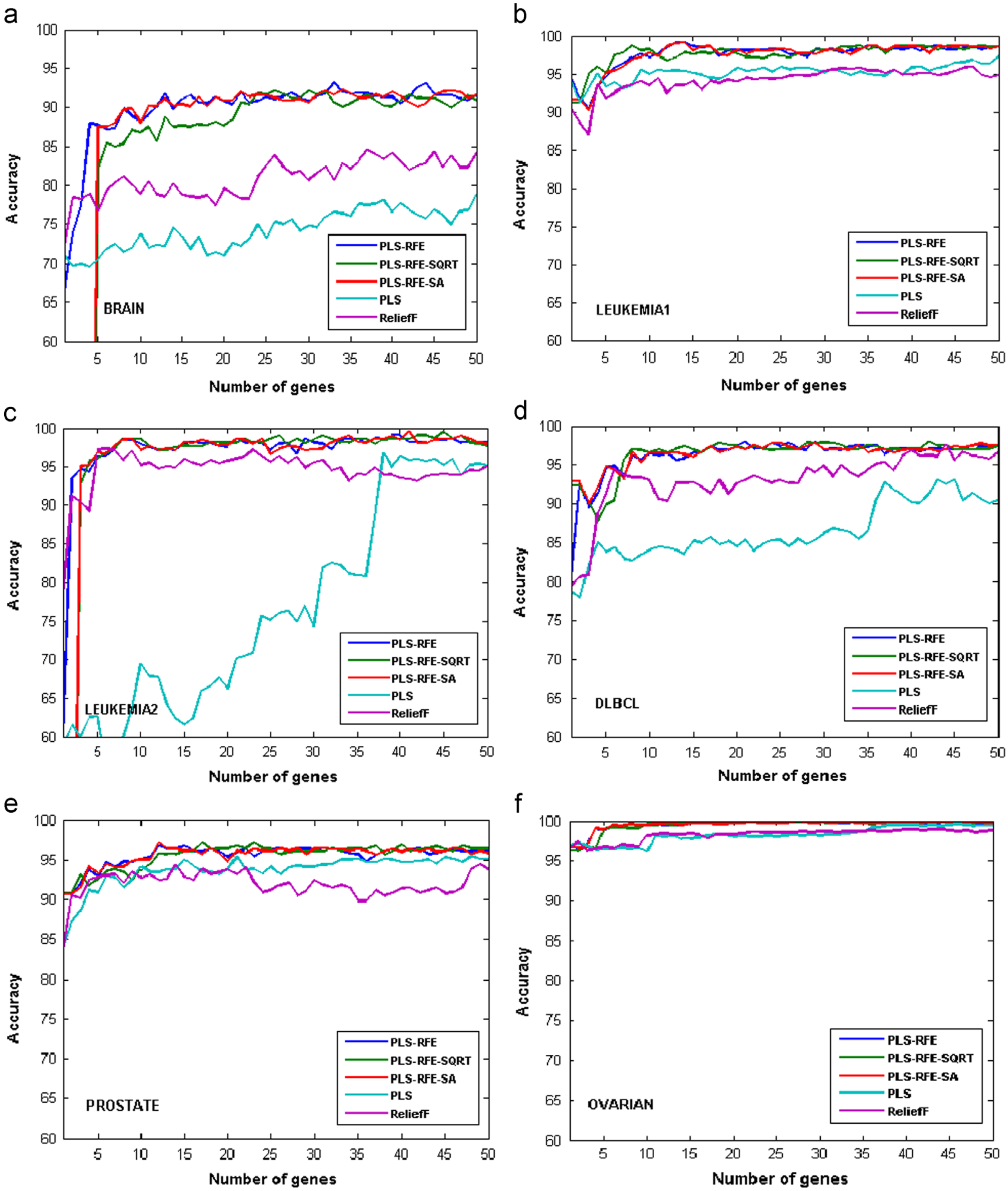

Fig. 1. Mean classification accuracy vs. the number of selected genes on the six microarray datasets. (a) Leukemia2, (b) DLBCL, (c) Prostate, (d) Ovarian.

Table 5
Time(s) cost comparison of the three feature selectors.

Dataset	PLS–RFE	PLS–RFE-SQRT	PLS–RFE-SA
Brain	34941.0 (5920)	605.8 (153)	59.7 (135)
Leukemia1	8375.3 (7129)	136.7 (168)	14.3 (150)
Leukemia2	10327.0 (5327)	187.5 (145)	19.5 (128)
DLBCL	8303.0 (7129)	134.0 (168)	12.8 (150)
Prostate	26044.0 (12,600)	311.7 (224)	24.5 (198)
Ovarian	39026.0 (15,154)	429.5 (246)	31.4 (216)

5.4. Feature subset consistency

In this section, we compare the consistency of PLS–RFE, PLS–RFE-SA and PLS–RFE-SQRT by calculating the feature subset similarity between any two of them using formula (7). Fig. 3 presents the results of the top 50 genes, where RFE&SA means the feature subset consistency between two feature subsets obtained by PLS–RFE and PLS–RFE-SA, respectively, RFE&SQRT is the feature subset consistency of PLS–RFE and PLS–RFE-SQRT, and SA&SQRT is the feature subset consistency of PLS–RFE-SA and PLS–RFE-SQRT.

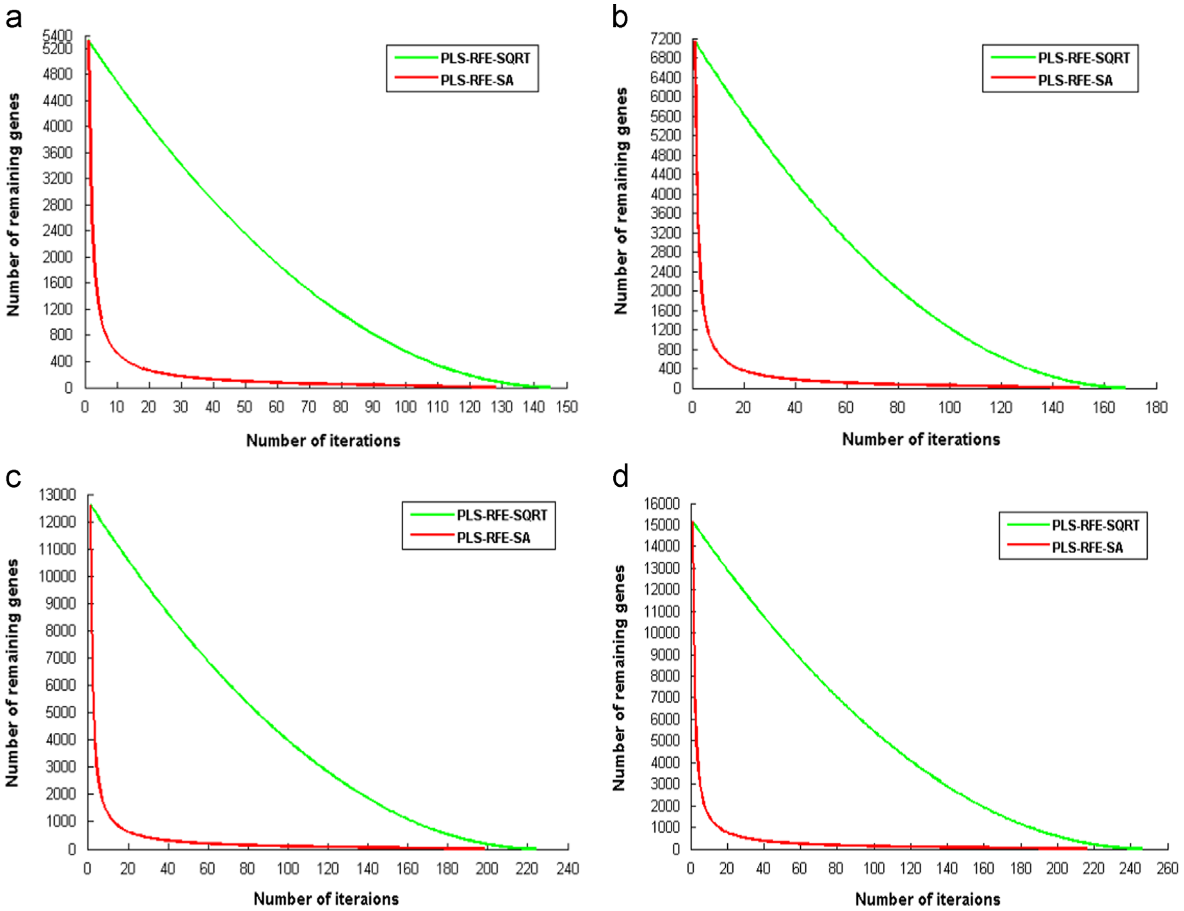

Fig. 2. Illustration to the iteration process on four microdata datasets.

X-axis refers to the top *k* genes obtained by each feature selector, and *Y*-axis refers to the corresponding feature subset consistency between two gene subsets.

One can observe that the consistency between any two of them on all the datasets is greater than 0.8 when the number of selected features is not less than 10, which indicates that the three methods can select very similar genes. This is probably because the three approaches adopt the same criterion in calculating the relevance of a feature to the target class and the partial least squares technique can effectively eliminate the effects of multicollinearity and identify the relevant explanatory variables. Meanwhile, this partially explains why PLS–RFE-SA and PLS–RFE-SQRT can obtain comparable classification accuracy to PLS–RFE. Further, we can observe that the feature subset consistency between PLS–RFE and PLS–RFE-SA is greater than that of PLS–RFE and PLS–RFE-SQRT, which means that PLS–RFE-SA is more likely to select the same genes as PLS–RFE than PLS–RFE-SQRT. In particular, PLS–RFE-SA selects the same subset of genes as PLS–RFE on *DLBCL*, *Leukemia*1 and *Leukemia*2 when the number of selected genes is less than 35, and can select exactly the same gene subset on *Prostate* and *Ovarian* when the number of selected genes is less than 50. Additionally, it should be noted that some parts of the green curve in each figure are missing. In fact, those parts overlap the corresponding blue parts and are covered by the blue parts. This is because PLS–RFE-SA selects the same feature subset as PLS–RFE and the consistency between PLS–RFE-SQRT and PLS–RFE is equal to the consistency between PLS–RFE-SQRT and PLS–RFE-SA.

Overall, according to the experimental results and analysis, we conclude that in comparison with PLS–RFE, both PLS–RFE-SA and PLS–RFE-SQRT achieve comparable classification accuracy while significantly speeding up the feature selection process for both the two-category and multi-category microarray data classification problems. This demonstrates the effectiveness of the two proposed approaches. In a further analysis, we can see that PLS–RFE-SA not only runs faster than PLS–RFE-SQRT, but also has slightly better feature subset consistency.

6. Conclusions

Feature selection, or gene selection in the context of microarray data, plays crucial roles in the analysis of gene expression profiles. Correspondingly, various machine learning and statistical learning techniques are used to identify a small subset of discriminative features from the original feature space. In practical use, partial least squares-based recursive feature elimination (PLS–RFE) approach is experimentally demonstrated to obtain feature subsets of good qualities in comparison with the state-of-the-art feature selectors. However, it is computationally expensive in handling datasets characterized by high dimensionality such as microarray data with thousands of genes. In this paper, we proposed to integrate two dynamic feature elimination schemes, *simulated annealing* scheme and *square root* scheme, respectively, into PLS–RFE to speed up the feature selection process. Inspired from the strategy of the annealing schedule, the two proposed approaches eliminate a number of genes rather than just one least informative gene during each iteration and the number of eliminated genes decreases as the iteration proceeds. To show the effectiveness and efficiency of the two proposed approaches, we included two other feature selectors, PLS and ReliefF as a comparison, and conducted experimental comparisons on six publicly available microarray data in terms of classification accuracy and

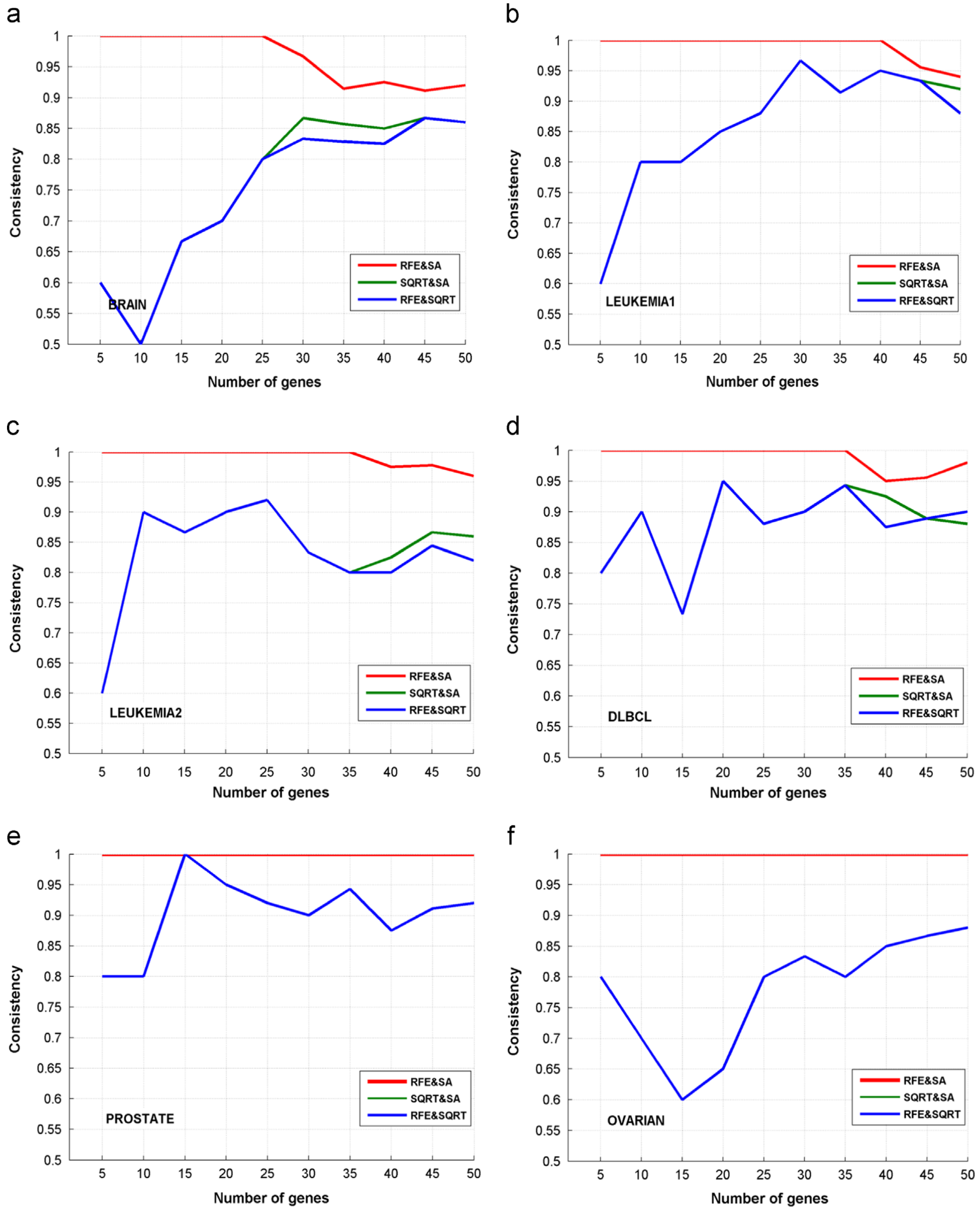

Fig. 3. Feature subset consistency between PLS–RFE, PLS–RFE-SA and PLS–RFE-SQRT. (a) Brain, (b) Leukemia1, (c) Leukemia2, (d) DLBCL, (e) Prostate, (f) Ovarian.

running time. In addition, we used Naïve Bayes, 3-Nearest-Neighbor and Support Vector Machine to evaluate the quality of the final selected features. Experimental results show that the two proposed approaches achieve comparable classification accuracy to PLS–RFE and outperform PLS and ReliefF, and that PLS–RFE is greatly accelerated with the two proposed feature elimination schemes. Furthermore, a further comparison in running time and feature subset consistency was conducted between the two proposed approaches and indicates that the one with *simulated annealing* scheme not only runs much faster, but also achieves better feature subset consistency than the one with *square root* scheme. Notably, in our study, although we name the approach, which combines PLS–RFE and simulated annealing, as PLS–RFE-SA, we do not consider the case that PLS–RFE-SA accepts some unfavorable features by random probability during each iteration. Therefore, investigating whether the classical simulated annealing algorithm in the context of PLS–RFE, which retains a part of unfavorable features during each iteration, could obtain better feature subsets than PLS–RFE-SA is an interesting issue for future research. Furthermore, although we focused only on the gene expression profiles and demonstrated the effectiveness and efficiency of the proposed approaches merely using microarray data, they are general feature selection methods that can be applied to other fields that also suffer from curse of dimensionality such as text categorization [47], proteomics and RNA-Seq datasets [48]. Our future research work will extend the proposed approaches and test their performance in these fields.

7. Summary

Since microarray data is characterized by high dimensionality and small sample sizes and contains irrelevant and redundant genes, gene selection plays a crucial role in constructing effective and efficient classifiers in classifying microarray data. Accordingly, various machine learning and statistical techniques have been proposed and applied. In practical use, partial least squares based feature recursive feature (PLS–RFE) has been experimentally shown to obtain feature subsets of good qualites in comparison with other state-of-the-art feature selectors, therefore, it can help construct powerful classifiers. However, it is considerably time-consuming in handling datasets with high-dimensionality such as the microarray data. How to accelerate this process without degrading the high accuracy is the mian focus of our study.

In this paper, we propose to accelerate PLS–RFE with two improved feature elimination schemes, similated annealing scheme and square root scheme. In contrast with the classical approach which eliminates only one least informative feature, the two proposed approaches eliminate a larger number of features in the initial iterations and eliminate a smaller number of features as the iteration proceeds. Specifically, the approach with simulated annealing scheme eliminates $|S|/(j+1)$ features during each iteration, and the one with square root scheme eliminates $\sqrt{|S|}$ features during each iteration, where $|S|$ is the number of remaining features before each iteration and j is the iteration counter. Obviously, the two approaches would definitely accelerate the feature selection process by reducing the running times of SIMPLS required to rank all the features.

To verify the effectivenss of the two proposed approaches, experiemental comparisons were conducted on six publicly available microarry datasets in terms of classification accuracy and the size of the final selected feature subsets. Besides in comparison with PLS–RFE, we include two other well-performing feature selectors, PLS and ReliefF as well. In addition, to evaluate the quality of the final selected feature subsets, three commonly used classifiers with different metrics, Naïve Bayes, 3-Nearest-Neighbor and Support Vector Machine are used in our study. Experimental results show: (a) that the two proposed approaches obtain comparable classification accuracy to PLS–RFE and outperform both PLS and ReliefF; (b) and that the size of the final selected feature subsets of our approaches is comparable to that of PLS–RFE and is much smaller than that of PLS and ReliefF, which is preferable in the further analysis of microarray data and biological validation;(c) and that the two proposed approaches greatly speed up the feature selection process. Furthermore, experimental comparisons were conducted between the two proposed approaches in feature subset consistency and running time. Experimental results show that the approach with simulated annealing scheme has better feature subset consistency and time performance in comparison to the one with square root scheme.

Conflict of interest statement

None declared.

Acknowledgments

This work was supported partially by the “111 Project” of the Ministry of Education and State Administration of Foreign Experts Affairs (No. B14025), the International S&T Cooperation Program of China (No. 2014DFA11310), the “University Featured Project” of the Ministry of Education (No. TS2013HFGY031), the Chinese National Key Technology R&D Program (No. 2013BAH19F01), the Major Project of Natural Science Foundation for Anhui Province Higher Education (No. KJ2011ZD06), and the National Natural Science Foundation of China (Nos. 61472057 and 61305064). Aiguo Wang was a visiting student in the Center for Biomedical Informatics at Harvard Medical School who was sponsored by the China Scholarship Council. The authors are very grateful to the anonymous reviewers for their constructive comments and suggestions for the improvement of this research.

References

[1] W. Zhou, J. Dickerson, A novel class dependent feature selection method for cancer biomarker discovery, Comput. Biol. Med. 47 (2014) 66–75.

[2] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, E. Lander, Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–537.

[3] T. Abeel, T. Helleputte, Y. de Peer, P. Dupont, Y. Saeys, Robust biomarker identification for cancer diagnosis with ensemble feature selection methods, Bioinformatics 26 (2010) 392–398.

[4] G. Piatetsky-Shapiro, P. Tamayo, Microarray data mining: facing the challenges, ACM SIGKDD Explorations Newsletter, 5, 20031–5.

[5] A. Jain, R. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 4–37.

[6] M. Pepe, G. Longton, G. Anderson, M. Schummer, Selecting differentially expressed genes from microarray experiments, Biometrics 59 (2003) 133–142.

[7] J. Hua, W. Tembe, E. Dougherty, Performance of feature-selection methods in the classification of high-dimension data, Pattern Recognit. 42 (2009) 409–424.

[8] A. Sharma, S. Imoto, S. Miyano, A top-r feature selection algorithm for microarray gene expression data, IEEE/ACM Trans. Comput. Biol. Bioinform. 9 (2012) 754–764.

[9] I.A. Gheyas, L.S. Smith, Feature subset selection in large dimensionality domains, Pattern Recognit. 43 (2010) 5–13.

[10] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (2005) 491–502.

[11] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (2007) 2507–2517.

[12] R. Kohavi, G. John, Wrappers for feature subset selection, Artif. Intell. 97 (1997) 273–324.

[13] I. Inza, P. Larrañaga, R. Blanco, J. Cerrolaza, Filter versus wrapper gene selection approaches in DNA microarray domains, Artif. Intell. Med. 31 (2004) 91–103.

[14] T. Hwang, C.H. Sun, T. Yun, G. Yi, FiGS: a filter-based gene selection workbench for microarray data, BMC Bioinform. 11 (2010) 50.

[15] R. Ruiz, J. Riquelme, J. Aguilar-Ruiz, Incremental wrapper-based gene selection from microarray data for cancer classification, Pattern Recognit. 39 (2006) 2383–2392.

[16] S. Maldonado, R. Weber, A wrapper method for feature selection using support vector machines, Inform. Sci. 179 (2009) 2208–2217.

[17] K. Moorthy, M.S. Mohamad, Random forest for gene selection and microarray data classification, Bioinformation 7 (2011) 142.

[18] S.S. Shreem, S. Abdullah, M.Z. Nazri, Hybridising harmony search with a Markov blanket for gene selection problems, Inform. Sci. 258 (2014) 108–121.

[19] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, J. Mach. Learn. Res. 5 (2004) 1205–1224.

[20] C. Zhang, X. Lu, X. Zhang, Significance of gene ranking for classification of microarray samples, IEEE/ACM Trans. Comput. Biol. Bioinform. 3 (2006) 312–320.

[21] A. Boulesteix, K. Strimmer, Partial least squares: a versatile tool for the analysis of high-dimensional genomic data, Brief. Bioinform. 8 (2007) 32–44.

[22] W. You, Z. Yang, G. Ji, PLS-based recursive feature elimination for high-dimensional small sample, Knowl.-Based Syst. 55 (2014) 15–28.

[23] K. Cao, S. Boitard, P. Besse, Sparse PLS discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems, BMC Bioinform. 12 (2011) 253.

[24] M. Dash, H. Liu, Consistency-based search in feature selection, Artif. Intell. 151 (2003) 155–176.

[25] M.C. Robini, P. Reissman, From simulated annealing to stochastic continuation: a new trend in combinatorial optimization, J. Global Optim. 56 (2013) 185–215.

[26] R. Precup, R.C. David, E.M. Petriu, S. Preitl, M. Radac, Fuzzy control systems with reduced parametric sensitivity based on simulated annealing, IEEE Trans. Ind. Electron. 59 (2012) 3049–3061.

[27] A. Krishnan, L.J. Williams, A.R. McIntosh, H. Abdi, Partial Least Squares (PLS) methods for neuroimaging: a tutorial and review, Neuroimage 56 (2011) 455–475.

[28] S. de Jong, SIMPLS: an alternative approach to partial least squares regression, Chemom. Intell. Lab. Syst. 18 (1993) 251–263.

[29] R. Gosselin, D. Rodrigue, C. Duchesne, A bootstrap-VIP approach for selecting wavelength intervals in spectral imaging applications, Chemom. Intell. Lab. Syst. 100 (2010) 12–21.

[30] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.

[31] I.A. Gheyas, L.S. Smith, Feature subset selection in large dimensionality domains, Pattern Recognit. 43 (2010) 5–13.

[32] W. Ma, T. Zhang, P. Lu, S. Lu, Partial least squares based gene expression analysis in estrogen receptor positive and negative breast tumors, Eur. Rev. Med. Pharmacol. Sci. 18 (2014) 212–216.

[33] K. Liu, B. Li, Q. Wu, J. Zhang, J. Du, G. Liu, Microarray data classification based on ensemble independent component selection, Comput. Biol. Med. 39 (2009) 953–960.

[34] W. Li, Y. Yang, How many genes are needed for a discriminant microarray data analysis, in: S.M. Lin, K.F. Johnson (Eds.), Methods of Microarray Data Analysis, Kluwer Academic Publishers, Springer, US, 2002, pp. 137–149.

[35] L.I. Kuncheva, A stability index for feature selection, Artif. Intell. Appl. (2007) 421–427.

[36] A. Statnikov, I. Tsamardinos, Y. Dosbayev, C. Aliferis, GEMS: a system for automated cancer diagnosis and biomarker discovery from microarray gene expression data, Int. J. Med. Inform. 74 (2005) 491–503.

[37] M.A. Shipp, K.N. Ross, P. Tamayo, A.P. Weng, J.L. Kutok, R. Aguiar, M. Gaasenbeek, et al., Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning, Nat. Med. 8 (2002) 68–74.

[38] D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, et al., Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2002) 203–209.

[39] E.F. Petricoin, A.M. Ardekani, B.A. Hitt, P.J. Levine, V.A. Fusaro, S.M. Steinberg, L. A. Liotta, Use of proteomic patterns in serum to identify ovarian cancer, Lancet 359 (2002) 572–577.

[40] U. Braga-Neto, E. Dougherty, Is cross-validation valid for small-sample microarray classification? Bioinformatics 20 (2004) 374–380.

[41] J. Huang, H. Fang, X. Fan, Decision forest for classification of gene expression data, Comput. Biol. Med. 40 (2010) 698–704.

[42] S. Sun, Q. Peng, A. Shakoor, A kernel-based multivariate feature selection method for microarray data classification, PLoS One 9 (2014) e102541.

[43] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (2003) 23–69.

[44] T. Calders, S. Verwer, Three naive Bayes approaches for discrimination-free classification, Data Min. Knowl. Discov. 21 (2010) 277–292.

[45] A. Wang, N. An, G. Chen, L. Li, G. Alterovitz, Accelerating incremental wrapper based gene selection with K-Nearest-Neighbor, in: Proceedings of the 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Belfast, November 2–5, 2014, pp. 21–23.

[46] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27.

[47] G. Forman, An extensive empirical study of feature selection metrics for text classification, J. Mach. Learn. Res. 3 (2003) 1289–1305.

[48] M.G. Grabherr, B.J. Haas, M. Yassour, J.Z. Levin, D.A. Thompson, I. Amit, A. Regev, Full-length transcriptome assembly from RNA-Seq data without a reference genome, Nat. Biotechnol. 29 (2011) 644–652.

Aiguo Wang was born in Anhui Province, China in 1986. He received B.Sc. in Hefei University of Technology in 2010. He is now a Ph.D. candidate with the School of Computer and Information of Hefei University of Technology. His research interests include data mining, bioinformatics, and healthcare information system.

Ning An was born in Gansu Province, China in 1971. He received B.Sc. and M.Sc. in Lanzhou University in 1993 and 1996, respectively, and Ph.D. in Pennsylvania State University in 2002. He is now a professor with School of Computer and Information, Hefei University of Technology. His research interests include gerontechnology, healthcare informatics, data mining, and spatial information management.

Guilin Chen was born in Anhui Province, China in 1965. His received M.Sc. in Anhui Normal University 1985. He is now a professor with School of Computer and Information Engineering, Chuzhou University. His research interests include Internet of Things, Cloud Computing, and pervasive computing.

Lian Li was born in Shangdong Province, in 1951. He is a professor of Hefei University of Technology. His research interests include computational mathematics, grid computing, and social computing.

Gil Alterovitz is assistant professor of Pediatrics at Harvard Medical School and is on the faculty of Boston Children's Hospital. He received his S.M. and Ph.D. in Electrical and Biomedical Engineering at Massachusetts Institute of Technology (MIT) in 2001 and 2006, respectively. His research interests include bioinformatics and proteomics.