PLS AND SVD BASED PENALIZED LOGISTIC REGRESSION FOR CANCER CLASSIFICATION USING MICROARRAY DATA

LI SHEN AND ENG CHONG TAN

School of Computer Engineering, Nanyang Technological University, Singapore 639798

Accurate cancer prediction is important for treatment of cancers. The combination of two dimension reduction methods, partial least squares (PLS) and singular value decomposition (SVD), with the penalized logistic regression (PLR) has created powerful classifiers for cancer prediction using microarray data. Comparing with support vector machine (SVM) on seven publicly available cancer datasets, the new algorithms can achieve very good performance and run much faster. They also have the advantage that the probabilities of predictions can be directly given. PLS based PLR is also combined with recursive feature elimination (RFE) to select a 16-gene subset for acute leukemia cancer classification. The testing error on this subset of genes is empirically zero.

1 INTRODUCTION

The advent of DNA microarray and protein mass spectra has enabled us to measure thousands of expression levels of genes simultaneously. These gene expression profiles can be used to classify different types of tumors and there have been a lot of activities in this area of cancer classification.

One problem often encountered is that there are a huge number n (thousands) of features but relatively small number m (tens or hundreds) of samples or arrays due to the high cost of microarray experiment. Since the data dimension is very large, SVM has been found to be very useful for this classification problem [1]. Apart from the classification task, it is also important to eliminate the irrelevant genes from the dataset and select a small subset of marker genes which discriminate between different types of tissue samples.

The penalized logistic regression (PLR) has been proposed by other authors for cancer classification [2-4]. It has been shown to be a powerful classifier in this field. We, however, combined this method with the dimension reduction method known as partial least squares (PLS) and singular value decomposition (SVD). We will illustrate that the new algorithm is computationally efficient and comparing with SVM on seven publicly available cancer datasets, the performance of PLS and SVD based PLR is very good and competitive. But we also indicate that PLS based PLR generally performs better than SVD based PLR and uses significantly fewer components.

Feature selection is another very important part in the field of cancer classification. Instead of ranking the relevance of genes with the cancers individually, recursive feature elimination (RFE) which was first proposed by Guyon et al. [5] is used in this paper. PLS based PLR was combined with RFE to select a small 16-gene subset for classification. The testing error on this subset using random partition method turns out to be zero. Some of the genes selected in this subset overlap with the genes selected by other authors.

2 METHODS

2.1. Penalized Logistic Regression

Assume we have a number of cancer classification samples from microarray experiments. Each sample can be in one of two classes, e.g. class 0 and class 1. A rule based on logistic regression is to be determined, which uses the gene expression profiles on an array to determine the probability that a sample belongs to one of the two classes. A training dataset of samples with known class labels is present to derive the rule and the rule derived should be able to classify any new sample that comes along.

Let a variable y indicate the class of a microarray sample: $y = 0$ means the sample belongs to class 0; $y = 1$ means the sample belongs to class 1. Let x_j indicate the j th gene expression level of the sample. We are trying to find a formula that gives us the probability p that the sample with its all measured expression $\mathbf{x}^T = \left[x_1, x_2, \ldots, x_n\right]$ represents a class 1 case. Since only two classes are considered, the probability of the sample representing class 0 is consequently $1 - p$. The normal logistic regression model would be

$$\eta = \log \frac{p}{1-p} = \alpha + \sum_{j=1}^{n} \beta_j x_j$$

where α and $\beta_1, \beta_2, \ldots \beta_n$ are parameters and they could be estimated by maximum likelihood (ML) criterion. Then the curve that computes p from η:

$$p = \frac{1}{1 + e^{-\eta}}$$

is called the logistic curve, hence the name logistic regression.

In the setting of microarray experiments, the number of samples, m, is usually on the order of tens or hundreds but the number of variables, n, is usually on the order of thousands or even tens of thousands. So the number of samples is much less than the number of variables. There are three problems in this situation when we are trying to build a logistic regression rule:

- If $m < n$, there will be more unknowns than equations, possible solutions are infinite.
- Data overfitting may occur. That means, we may have zero errors on training data but very poor performance on new samples.
- Multicollinearity largely exists: many genes will show nearly identical patterns across the samples, so they supply no new information to the data; some gene profiles can be linear combinations of the other gene profiles.

These problems can be solved by introducing a penalty into the logistic regression formulation. The regularization on the sum of the squares of the regression coefficients is known as ridge regression [6]. It has been applied to logistic regression by [7]. The penalized logistic regression is now given in the following.

Let y_i indicate the class of the i th sample and p_i the probability that $y_i = 1$. Let x_{ij} indicate the j th gene expression level of the i th sample. The model is

$$\eta_i = \log \frac{p_i}{1-p_i} = \alpha + \sum_{j=1}^{n} \beta_j x_{ij} \tag{1}$$

where η_i is called the linear predictor in the jargon of generalized linear models, as it is a linear combination of the explanatory variables. It is connected to p_i by a non-linear (logarithm) so-called link function. The log-likelihood is

$$L = \sum_{i=1}^{m} y_i \log p_i + \sum_{i=1}^{m} (1-y_i) \log(1-p_i) \tag{2}$$

The penalized log-likelihood is

$$L^* = L - \frac{\lambda}{2} \sum_{j=1}^{n} \beta_j^2 \tag{3}$$

where λ is called the penalty parameter. The larger λ, the stronger its influence and the smaller the β_j^2's are forced to be. The value of λ can be determined by cross-validation. The ML method estimates the parameters by maximizing Eq. (3). Let $\mathbf{u}$ be a m-vector of ones; $\mathbf{y} = [y_1, y_2, \ldots, y_m]^T$; $\mathbf{p} = [p_1, p_2, \ldots, p_m]^T$; $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots \beta_n]^T$; $\mathbf{X}$ be a $m \times n$ matrix so that $\mathbf{X}(i, j) = x_{ij}$. Now we take the derivatives of L^* against α and β_j so that:

$$\partial L^* / \partial \alpha = 0 \Rightarrow \mathbf{u}^T (\mathbf{y} - \mathbf{p}) = 0 \tag{4}$$

$$\partial L^* / \partial \boldsymbol{\beta} = 0 \Rightarrow \mathbf{X}^T (\mathbf{y} - \mathbf{p}) = \lambda \boldsymbol{\beta} \tag{5}$$

Eqs. (4) and (5) are non-linear because of the non-linear relationship between $\mathbf{p}$ and α and $\boldsymbol{\beta}$. To get a set of linear equations, we take the first order Taylor expansion of p_i ,

$$p_i = \tilde{p}_i + \frac{\partial p_i}{\partial \alpha}(\alpha - \tilde{\alpha}) + \sum_{j=1}^{n} \frac{\partial p_i}{\partial \beta_j}(\beta_j - \tilde{\beta}_j) \tag{6}$$

where a tilde indicates an approximate solution. Now

$$\frac{\partial p_i}{\partial \alpha} = \tilde{p}_i (1 - \tilde{p}_i) \tag{7}$$

$$\frac{\partial p_i}{\partial \beta_j} = \tilde{p}_i (1 - \tilde{p}_i) x_{ij} \tag{8}$$

Using this and introducing $\tilde{w}_i = \tilde{p}_i (1 - \tilde{p}_i)$, $\tilde{\mathbf{W}} = \operatorname{diag}(\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_m)$, we have

$$\mathbf{u}^T \tilde{\mathbf{W}} \mathbf{u} \alpha + \mathbf{u}^T \tilde{\mathbf{W}} \mathbf{X} \boldsymbol{\beta} = \mathbf{u}^T (\mathbf{y} - \tilde{\mathbf{p}} + \tilde{\mathbf{W}} \tilde{\boldsymbol{\eta}}) \tag{9}$$

$$\mathbf{X}^T \tilde{\mathbf{W}} \mathbf{u} \alpha + (\mathbf{X}^T \tilde{\mathbf{W}} \mathbf{X} + \lambda \mathbf{I}) \boldsymbol{\beta} = \mathbf{X}^T (\mathbf{y} - \tilde{\mathbf{p}} + \tilde{\mathbf{W}} \tilde{\boldsymbol{\eta}}) \tag{10}$$

where $\tilde{\boldsymbol{\eta}} = [\tilde{\eta}_1, \tilde{\eta}_2, \ldots \tilde{\eta}_m]^T$ and

$$\tilde{\eta}_i = \tilde{\alpha} + \sum_{j=1}^{n} \tilde{\beta}_j x_{ij} \tag{11}$$

for $i = 1, 2, \ldots, m$. Now Eqs. (9) and (10) constitute a linearized system and iterating with it generally leads to a solution quickly. In most cases ten iterations are enough. Suitable starting values are $\tilde{\alpha} = \log[\bar{y}/(1 - \bar{y})]$ with $\bar{y} = \sum_{i=1}^{m} y_i \big/ m$ and $\tilde{\boldsymbol{\beta}} = 0$. If we introduce $\boldsymbol{\gamma}^T = [\alpha \,|\, \boldsymbol{\beta}^T]$ and $\mathbf{Z} = [\mathbf{u} \,|\, \mathbf{X}]$, Eqs. (9) and (10) can be written as

$$(\mathbf{Z}^T\tilde{\mathbf{W}}\mathbf{Z}+\lambda\mathbf{R})\boldsymbol{\gamma}=\mathbf{Z}^T(\mathbf{y}-\tilde{\mathbf{p}}+\tilde{\mathbf{W}}\mathbf{Z}\tilde{\gamma}) \quad (12)$$

where $\mathbf{R}$ is a $(n+1)\times(n+1)$ identity matrix with $\mathbf{R}(1,1)=0$ to reflect that there is no penalty on α.

2.2. Partial Least Squares and Singular Value Decomposition

The linear system of Eqs. (9) and (10) is huge: thousands of equations with an equal number of unknowns. Solving this could be computationally problematic and storing all the equations takes a substantial amount of memory space. PLS and SVD are both very popular dimension reduction methods and they have been successfully applied to the field of gene expression based cancer classification. In this paper, both of these methods are proposed to undertake the task of solving Eqs. (9) and (10). For an updated survey of PLS, readers can refer to [8]. For definition and computation of SVD, readers can refer to [9]. We would not go into details of these two techniques here.

First, assume the $m\times n$ matrix $\mathbf{X}$ stores all of the gene expression data with its rows being the microarray samples and its columns being the gene profiles, the formulations of PLS and SVD give the decomposition of $\mathbf{X}$:

$$\mathbf{X}=\mathbf{P}\mathbf{V}^T+\mathbf{R}$$

where $\mathbf{P}$ is $m\times p$ matrix, $\mathbf{V}$ is $n\times p$ matrix and $\mathbf{R}$ is $m\times n$ matrix. p is the number of PLS components or singular values and $p\le m$. $\mathbf{R}$ is the residual matrix and can be considered as containing no useful information. Therefore, $\mathbf{X}$ can be approximated as

$$\mathbf{X}\approx\mathbf{P}\mathbf{V}^T \quad (13)$$

In PLS, the columns of $\mathbf{P}$ are called score vectors and the columns of $\mathbf{V}$ are called loading vectors. In SVD, the usual formulation of decomposition of $\mathbf{X}$ is

$$\mathbf{X}\approx\mathbf{U}\mathbf{S}\mathbf{V}^T \quad (14)$$

where $\mathbf{U}$ is $m\times p$ matrix, $\mathbf{S}$ is $p\times p$ diagonal matrix and $\mathbf{V}$ is also $n\times p$ matrix. For convenience, let $\mathbf{P}=\mathbf{US}$ for SVD and use score vectors and loading vectors to name the columns of $\mathbf{P}$ and $\mathbf{V}$. Hence we can use Eq. (13) to represent the decomposition of both PLS and SVD. The loading vectors produced by PLS and SVD are always mutually orthogonal and they are assumed to be normalized in PLS so that $\mathbf{V}^T\mathbf{V}=\mathbf{I}$, which is a $p\times p$ identity matrix. Assume $\boldsymbol{\beta}=\mathbf{V}\boldsymbol{\theta}$ and substitute Eq. (13) into Eqs. (9) and (10) and we have

$$\mathbf{u}^T\tilde{\mathbf{W}}\mathbf{u}\alpha+\mathbf{u}^T\tilde{\mathbf{W}}\mathbf{P}\mathbf{V}^T\mathbf{V}\boldsymbol{\theta}=\mathbf{u}^T(\mathbf{y}-\tilde{\mathbf{p}}+\tilde{\mathbf{W}}\tilde{\boldsymbol{\eta}}) \quad (15)$$

$$\mathbf{V}\mathbf{P}^T\tilde{\mathbf{W}}\mathbf{u}\alpha+(\mathbf{V}\mathbf{P}^T\tilde{\mathbf{W}}\mathbf{P}\mathbf{V}^T+\lambda\mathbf{I})\mathbf{V}\boldsymbol{\theta}=\mathbf{V}\mathbf{P}^T(\mathbf{y}-\tilde{\mathbf{p}}+\tilde{\mathbf{W}}\tilde{\boldsymbol{\eta}}) \quad (16)$$

Multiplying Eq. (16) by $\mathbf{V}^T$ we get

$$\mathbf{u}^T\tilde{\mathbf{W}}\mathbf{u}\alpha+\mathbf{u}^T\tilde{\mathbf{W}}\mathbf{P}\boldsymbol{\theta}=\mathbf{u}^T(\mathbf{y}-\tilde{\mathbf{p}}+\tilde{\mathbf{W}}\tilde{\boldsymbol{\eta}}) \quad (17)$$

$$\mathbf{P}^T\tilde{\mathbf{W}}\mathbf{u}\alpha+(\mathbf{P}^T\tilde{\mathbf{W}}\mathbf{P}+\lambda\mathbf{I})\boldsymbol{\theta}=\mathbf{P}^T(\mathbf{y}-\tilde{\mathbf{p}}+\tilde{\mathbf{W}}\tilde{\boldsymbol{\eta}}) \quad (18)$$

Remember $\tilde{\boldsymbol{\eta}}=\mathbf{Z}\tilde{\gamma}$ where $\mathbf{Z}=[\mathbf{u}\,|\,\mathbf{X}]$ and $\gamma^T=[\alpha\,|\,\boldsymbol{\beta}^T]$, we have

$$\tilde{\boldsymbol{\eta}}=[\mathbf{u}\,|\,\mathbf{X}]\begin{bmatrix}\tilde{\alpha}\\ \tilde{\boldsymbol{\beta}}\end{bmatrix}=[\mathbf{u}\,|\,\mathbf{P}\mathbf{V}^T]\begin{bmatrix}\tilde{\alpha}\\ \mathbf{V}\tilde{\boldsymbol{\theta}}\end{bmatrix}=[\mathbf{u}\,|\,\mathbf{P}]\begin{bmatrix}\tilde{\alpha}\\ \tilde{\boldsymbol{\theta}}\end{bmatrix}$$

Redefine $\mathbf{Z}=[\mathbf{u}\,|\,\mathbf{P}]$ and $\gamma^T=[\alpha\,|\,\boldsymbol{\theta}^T]$ so that we have

$$\tilde{\boldsymbol{\eta}} = \mathbf{Z}\tilde{\boldsymbol{\gamma}} \tag{19}$$

Thus the system of Eqs. (17) and (18) can also be represented by Eq. (12).

The length of $\boldsymbol{\theta}$ is p, the number of score vectors. Therefore, the total equations in Eqs. (17) and (18) is $p+1$. Since $p \leq m << n$, the order of the system of Eqs. (9) and (10) is now effectively reduced from thousands to tens. Only a small amount of memory space is required and the equations can be solved quickly.

2.3. Feature Selection

RFE tries to find a subset of genes which are most relevant with the cancers instead of evaluating the importance of each gene individually. Firstly, we need to define the ranks of the genes by:

$$\tilde{\boldsymbol{\beta}} = \mathbf{V}\tilde{\boldsymbol{\theta}} \tag{20}$$

where $\tilde{\boldsymbol{\beta}}$ gives the estimates of the regression coefficients and its absolute values indicate the relative importance of the genes in the subset. The RFE procedure is designed as:

- For a subset of genes, leave-one-out cross-validation (LOOCV) is performed to find the λ which corresponds to the minimum LOOCV error.
- The averaged regression coefficients $\overline{\boldsymbol{\beta}}$ are calculated using 100 bootstrap samples from the original data with the λ fixed.
- The genes with the smallest $\left|\overline{\beta}_j\right|$ are eliminated to obtain a smaller subset.
- Evaluate the performance of the new subset of genes.

This procedure can be iterated for many times until there is only one gene left. An optimal subset of genes can be finally chosen.

3 RESULTS

3.1. Evaluation of classifier accuracy

For convenience, name the PLS based logistic regression as PLS-LOG and the SVD based logistic regression as SVD-LOG. The specifications of seven publicly available cancer datasets are listed in Table 1, which were chosen from [10]. A MATLAB version SVM [11] was also used to compare with the two methods. For each of these datasets, 100 random partitions were performed and each dataset was separated into a training dataset and a testing dataset. The means and standard deviations of testing errors and total time cost of the three classifiers were then recorded and listed in Table 2. The programs were all written in MATLAB and running on an ALPHA machine. In construction of the PLS-LOG and SVD-LOG, the number of components are empirically set to fifteen.

The classification accuracy of PLS-LOG and SVM are very similar and they both show minor advantage over SVD-LOG except on the lung cancer dataset. Though PLS-LOG generally runs faster than SVD-LOG, both of them cost much less time than SVM. We did not get results on prostate cancer data for SVM because the training appears to be endless. Solving the quadratic programming problem for SVM depends on the characteristics of different datasets and there seems to encounter some problem in convergence on this dataset.

Table 1 Description of the datasets.

Dataset	Genes	Partition Setting
Breast Cancer	24481	60 training v.s. 37 testing
Central Nervous System	7129	40 training v.s. 20 testing
Colon Tumor	2000	40 training v.s. 22 testing
Acute Leukemia	7129	40 training v.s. 32 testing
Lung Cancer	12533	100 training v.s. 81 testing
Ovarian Cancer	15154	150 training v.s. 103 testing
Prostate Cancer	12600	100 training v.s. 36 testing

Table 2 The means and standard deviations of testing errors of PLS-LOG, SVD-LOG and SVM on the seven datasets. The minimum testing errors and time cost are indicated in bold font.

Dataset	Testing Errors			Time Cost		
	PLS-LOG μ,σ	SVD-LOG μ,σ	SVM μ,σ	PLS-LOG (s)	SVD-LOG (s)	SVM (s)
Breast Cancer	**12.88, 2.89**	13.50, 2.58	13.09, 2.70	**4602**	5656	32615
Central Nervous System	**7.68, 1.76**	8.14, 2.58	7.76, 2.06	930	**924**	2297
Colon Tumor	3.97, 1.62	4.26, 1.55	**3.78, 1.23**	**404**	419	1577
Acute Leukemia	1.94, 1.82	2.74, 1.88	**1.57, 1.33**	**937**	1210	2363
Lung Cancer	0.83, 0.71	**0.52, 0.61**	0.83, 0.82	**5443**	7565	23245
Ovarian Cancer	**0.08, 0.42**	1.29, 1.26	0.22, 0.50	**14962**	17010	37931
Prostate Cancer	**4.68, 1.77**	5.38, 1.75	NA	**5419**	7200	NA

3.2. Choosing λ

In this and the following sections, the acute leukemia dataset is used for all data analysis and comparisons, which consists of two classes: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). This dataset was given by Golub *et al.* [12] and has totally 72 samples (47 ALL and 25 AML).

The estimates of the regression coefficients $\tilde{\boldsymbol{\beta}}$ are affected by λ significantly. A direct way is to select the penalty parameter by cross-validation. To find an optimal value of λ, it was varied in steps over a large range: $2^{-15}-2^{15}$, using 31 linearly spaced values for $\log_2 \lambda$. Fig. 1 shows the LOOCV errors v.s. $\log_2 \lambda$. For both PLS-LOG and SVD-LOG, the minimum LOOCV error is 3, which happens when $\log_2 \lambda$ is relatively small: about -14 to -7. The LOOCV error turns out to be 25, which is the number of AML samples, when $\log_2 \lambda$ is and larger than 0. The values of log-likelihood v.s. $\log_2 \lambda$ are also shown in the right panel of Fig. 1. The minimum log-likelihood is nearly zero. It indicates a successful training has been done.

Fig. 2 shows the probabilities, p, of prediction for ALL (AML is thus $1-p$). The results by PLS-LOG and SVD-LOG are given in the left and right panels, respectively. When λ becomes large, the probabilities of all samples converge to a fixed value, which equals the percentage of ALL in all samples. Another thing which should be noticed is that the curves given by PLS-LOG in the left panel overlap. That means, the probabilities given by PLS-LOG have much smaller deviance than SVD-LOG.

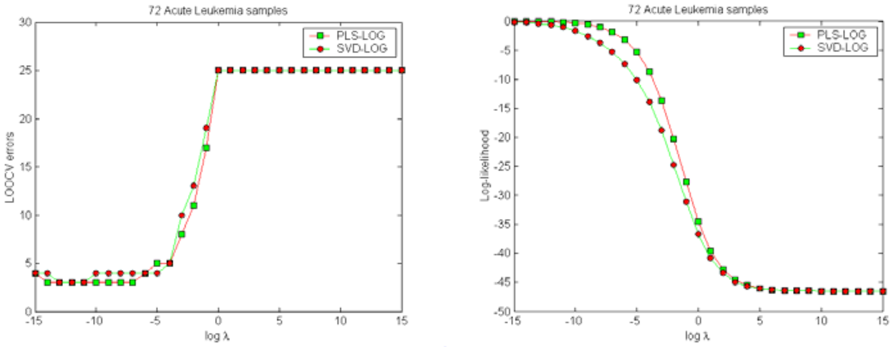

Fig. 1 LOOCV errors (left panel) and log-likelihood (right panel) v.s. $\log_2 \lambda$ for both PLS-LOG and SVD-LOG.

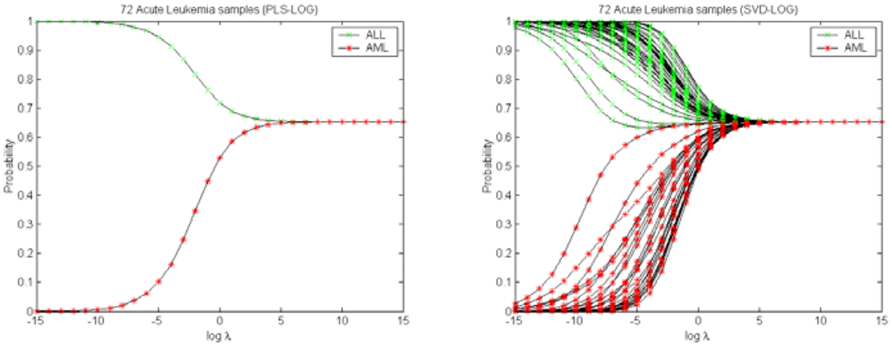

Fig. 2 Probability v.s. $\log_2 \lambda$ for both PLS-LOG (left panel) and SVD-LOG (right panel).

3.3. Components selection

Another important issue for PLS-LOG and SVD-LOG is the number of components used for training and testing. Because there is always noises in microarray cancer data, the maximum number of components that can be extracted from PLS and SVD is equal to the number of samples in the dataset. It is not necessary to use all of these components. Since we always sort these components according to their variances in descending order, only the first a few components are needed and the other components can be considered as noises. To determine the effect of components selection, we set $\lambda = 0$ and then perform LOOCV on the acute leukemia data while the number of components varies from 1 to 20. Fig. 3 illustrates this.

PLS-LOG achieves the minimum LOOCV error using only 5 components. It can even reaches 4 LOOCV errors using only 2 components. More components used than 5 did not help PLS-LOG to make better results. For SVD-LOG, the minimum LOOCV error appears when 10 components are used. The results of LOOCV begins to stabilize when 15 and more components are used for SVD-LOG. It is convincing that PLS produces components that are of more quality than SVD from this comparison. This condition can be further shown in Fig. 4, where we plot all samples using 2 components from PLS and SVD in the left and right panels, respectively. Two PLS components are enough to nicely separate all acute leukemia samples while the two clusters in the plot of SVD components overlap heavily.

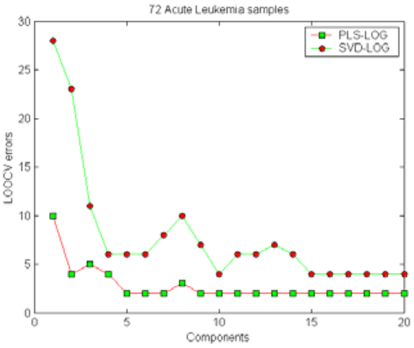

Fig. 3 LOOCV errors of PLS-LOG and SVD-LOG v.s. components.

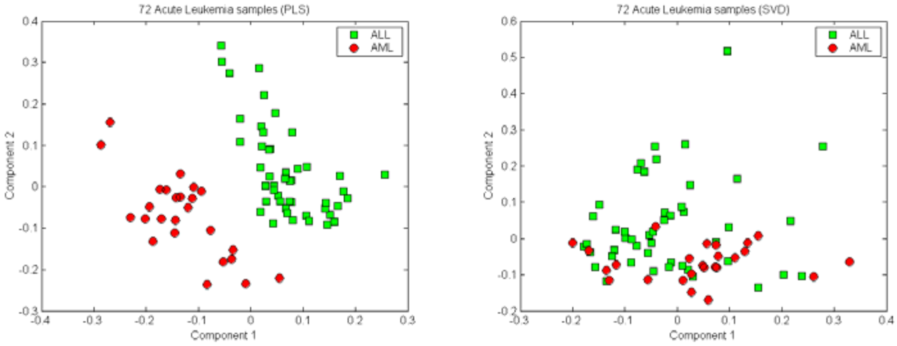

Fig. 4 Scatter plot of all acute leukemia samples using two components from PLS (left panel) and SVD (right panel).

3.4. Feature selection

Feature selection has been done on the acute leukemia data using RFE. Assume the number of genes in the subset is n. The optimal way to do RFE is to eliminate the least important gene one at each time but this can cost enormous time to complete. An eclectic method is to eliminate a large amount of genes at each time when n is large and less amount of genes when n becomes small. We design the RFE procedure in a way so that n is fixed for each subset, thus a series of nested subsets can be obtained and n would be 4096, 2048, 1024, 512, and so on. Each time n is halved until it is less than 10, then one gene is eliminated at each time. PLS-LOG was used to do RFE and the training and testing errors of each subset are listed in Table 3.

There are five subsets whose testing errors are zero. They are denoted as bold face in Table 3. Denote the subsets by their iterations, the smallest subset among them is subset 10 which has 16 genes. Also, very good results can be achieved by the subsets with less than 10 genes. We list the gene accession number, gene description and the averaged regression coefficient in Table 4 for the 16 genes in subset 10. Some of these genes overlap with the genes that were selected by other authors [5,12].

Table 3 Recursive feature elimination for acute leukemia data using PLS-LOG.

Iteration	Genes	Training μ	Training σ	Testing μ	Testing σ
1	7129	2.920000	1.495313	1.190000	1.541972
2	4096	0.790000	0.714850	0.270000	0.468287
3	2048	0.180000	0.386123	0.050000	0.219043
4	**1024**	**0.030000**	**0.171447**	**0.000000**	**0.000000**
5	**512**	**0.090000**	**0.287623**	**0.000000**	**0.000000**
6	256	0.010000	0.100000	0.010000	0.100000
7	**128**	**0.000000**	**0.000000**	**0.000000**	**0.000000**
8	64	0.230000	0.446196	0.150000	0.385992
9	**32**	**0.080000**	**0.307482**	**0.000000**	**0.000000**
10	**16**	**0.080000**	**0.272660**	**0.000000**	**0.000000**
11	10	0.080000	0.272660	0.040000	0.196946
12	9	0.250000	0.500000	0.120000	0.383498
13	8	0.110000	0.345096	0.060000	0.277798
14	7	0.140000	0.376588	0.080000	0.307482
15	6	0.560000	0.715203	0.420000	0.622475
16	5	0.670000	0.697108	0.500000	0.703526
17	4	1.100000	0.703526	0.640000	0.718022
18	3	1.720000	0.877093	0.940000	0.930081
19	2	2.290000	0.924362	1.210000	1.148517
20	1	3.330000	1.198105	1.530000	0.926108

Table 4 Sixteen genes selected by RFE for acute leukemia data.

Gene description	Accession number	$\bar{\beta}$
ALDH1 Aldehyde dehydrogenase 1, soluble	M31994	0.352435
CTSD Cathepsin D (lysosomal aspartyl protease)	M63138	0.288498
Zyxin	X95735	0.240914
MPO Myeloperoxidase	M19507	0.251528
CD33 CD33 antigen (differentiation antigen)	M23197	0.200143
Azurocidin gene	M96326	0.235837
GB DEF = CD171 protein	Y10207	0.179925
Tryptase-III mRNA, 3' end	M33493	0.128311
Heat-stable enterotoxin receptor mRNA	M73489	0.190334
Methyl sterol oxidase (ERG25) mRNA	U60205	0.159493
BGLAP Bone gamma-carboxyglutamate (gla) protein (osteocalcin)	X04143	0.176478
Biliverdin-IXalpha reductase mRNA	U34877	0.160293
Liver mRNA for interferon-gamma inducing factor(IGIF)	D49950	0.158959
S100A2 gene, exon 1, 2 and 3	Y07755	0.152358
GLUTATHIONE S-TRANSFERASE, MICROSOMAL	U46499	0.150263
A-Myb (Gb:X13294)	HG2562	0.146328

4 DISCUSSIONS

From the experiments, the penalty parameter λ chosen by PLS-LOG and SVD-LOG tends to be zero. This indicates that the first components contain little redundance. Larger λ were selected by other authors [2] who used all the original data for penalized logistic regression.

PLS generates components in the direction that maximizes the covariance between $\mathbf{X}$ and $\mathbf{y}$ while SVD components are in the direction that maximizes the variance of $\mathbf{X}$. Therefore, the PLS components already contain information about the class labels of the samples. Our results show that PLS-LOG generally performs better than SVD-LOG. The PLS components also show higher quality than SVD components.

There is no standard criterion to evaluate the quality of the subset of genes. One direct way is to look at the testing accuracy on the subset but the genes that distinguish well between two classes do not necessarily be the causes of the cancers. The 16-gene subset selected by PLS-LOG has achieved zero testing accuracy but further study are required to learn the roles that these genes play in causing the cancers.

REFERENCES

1. M. Brown, W. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. Furey, Jr.M. Ares and D. Haussler (2000) Knowledge based analysis of microarray gene expression data using support vector machines. *Proc. Natl Acad. Sci. USA*, 97, 262-267.
2. P.H.C. Eilers, J.M. Boer, G.J.B. Van Ommen, H.C. Van Houwelingen (2001) Classification of microarray data with penalized logistic regression. *Proceedings of SPIE volume 4266: progress in biomedical optics and imaging*, 2, 187–198.
3. M.G. Schimek (2003) Penalized logistic regression in gene expression analysis. Available: http://www.quantlet.org/hizirjsp/schimek/schimek.pdf.
4. J. Zhu and T. Hastie (2004) Classification of gene microarrays by penalized logistic regression. *Biostat*., 5, 427-443.
5. I. Guyon, J. Weston, S. Barnhill, V. Vapnik (2002) Gene selection for cancer classification using support vector machines. *Maching learning*, 46, 389-422.
6. A.E. Hoerl and R.W. Kennard (1970) Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12, 55-67.
7. S. le Cessie and J.C. van Houwelingen (1992) Ridge estimators in logistic regression. *Applied Statistics*, 41, 191-201.
8. J.A. Wegelin (2000) *A survey of Partial Least Squares (PLS) methods, with emphasis on the two-block case* (Technical Report). Department of Statistics, University of Washington, Seattle.
9. G.H. Golub and C.F. Van Loan (1996). *Matrix Computations*. The Johns Hopkins University Press.
10. J. Li and H. Liu (2002) Kent Ridge Biomedical Data Set Repository. Available: http://sdmc-lit.org.sg/GEDatasets.
11. A. Schwaighofer (2001) Available: http://www.cis.tugraz.at/igi/aschwaig/svm_v251.tar.gz.
12. T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfeld, E.S. Lander (1999) Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*. Vol 286, 531-537.