Biostatistics (2004), **5**, 3, *pp.* 427–443
doi: 10.1093/biostatistics/kxg046

Classification of gene microarrays by penalized logistic regression

JI ZHU[†]

Department of Statistics, University of Michigan, Ann Arbor, MI 48109, USA
jizhu@umich.edu

TREVOR HASTIE

Department of Statistics, Stanford University, Stanford, CA 94305, USA

SUMMARY

Classification of patient samples is an important aspect of cancer diagnosis and treatment. The support vector machine (SVM) has been successfully applied to microarray cancer diagnosis problems. However, one weakness of the SVM is that given a tumor sample, it only predicts a cancer class label but does not provide any estimate of the underlying probability. We propose penalized logistic regression (PLR) as an alternative to the SVM for the microarray cancer diagnosis problem. We show that when using the same set of genes, PLR and the SVM perform similarly in cancer classification, but PLR has the advantage of additionally providing an estimate of the underlying probability. Often a primary goal in microarray cancer diagnosis is to identify the genes responsible for the classification, rather than class prediction. We consider two gene selection methods in this paper, univariate ranking (UR) and recursive feature elimination (RFE). Empirical results indicate that PLR combined with RFE tends to select fewer genes than other methods and also performs well in both cross-validation and test samples. A fast algorithm for solving PLR is also described.

Keywords: Cancer diagnosis; Feature selection; Logistic regression; Microarray; Support vector machines.

1. INTRODUCTION

Classification of patient samples is an important aspect of cancer diagnosis and treatment. Conventional diagnosis of cancer has been based on examination of the morphological appearance of stained tissue specimens under light microscopy. However, this method is subjective and depends on highly trained pathologists. Microarrays offer hope that cancer classification can be objective and highly accurate, providing clinicians with the information to choose the most appropriate forms of treatment. See Golub *et al.* (1999), Khan *et al.* (2001), Lee and Lee (2002), Ramaswamy *et al.* (2001), Tibshirani *et al.* (2002) and references therein for recent work.

The support vector machine (SVM) is one of the methods that has been successfully applied to the cancer diagnosis problem in previous studies (Lee and Lee, 2002; Mukherjee *et al.*, 1999; Ramaswamy *et*

[†]To whom correspondence should be addressed.

al., 2001). In two-class classification, the linear SVM fits a model $f(\vec{x}) = b_0 + \sum_{j=1}^{p} b_j x_j$ that minimizes

$$\sum_{i=1}^{n} (1 - y_i f(\vec{x}_i))_+ + \frac{\lambda}{2}(\|b_1\|^2 + \cdots + \|b_p\|^2). \tag{1.1}$$

The classification decision is then made according to sign$[f(\vec{x})]$. However, one weakness of the SVM is that it only estimates sign$[p(\vec{x}) - 1/2]$, while the probability $p(\vec{x})$ is often of interest itself, where $p(\vec{x}) = P(C = 1|X = \vec{x})$ is the conditional probability of a point being in class 1 given $X = \vec{x}$. In going from two-class to multi-class classification, the one-vs-rest scheme is often used: given K classes, the problem is divided into a series of K one-vs-rest problems, and each one-vs-rest problem is addressed by a different class-specific SVM classifier (e.g. 'class 1' vs. 'not class 1'); then a new sample takes the class of the classifier with the largest real valued output $c = \text{argmax}_{k=1,\ldots K} f_k$, where f_k is the real valued output of the kth SVM classifier. Recently, Lee and Lee (2002) extends the two-class SVM to the multi-class case in a more direct way and estimates $\text{argmax}_k P(C = k|X = \vec{x})$ directly, but it still lacks the estimates of $P(C = k|X = \vec{x})$ themselves.

In this paper, we use penalized logistic regression (PLR) classifier to address the cancer diagnosis problem. The motivation for the use of PLR is pointed out in Zhu and Hastie (2002). PLR not only performs as well as the SVM in two-class classification, but can also naturally be generalized to the multi-class case. Furthermore, PLR provides an estimate of the underlying class probabilities $p(\vec{x})$.

Maximum likelihood and the Newton–Raphson algorithm is the traditional way to solve PLR numerically. However, the computation is prohibitive when the number of variables is large. We use a transformation trick to make the computation feasible, but the computational cost can still be high. To further reduce the computation, we use a sequential minimal optimization (SMO) algorithm (Platt, 1998) to solve PLR in this paper. This method was first proposed in Keerthi *et al.* (2002) for two-class classification; we generalize it to the multi-class case.

Besides predicting the correct cancer class for a given tumor sample, another challenge in microarray cancer diagnosis is to identify the relevant genes which contribute most to the classification. We consider two feature (gene) selection methods in this paper, univariate ranking (UR) (Dudoit *et al.*, 2002; Golub *et al.*, 1999), and recursive feature elimination (RFE) (Guyon *et al.*, 2002), and compare their performance when used with both PLR and the SVM on three cancer diagnosis data sets.

Our analysis of these data sets indicates that PLR and the SVM perform similarly when combined with either univariate ranking or recursive feature elimination; the recursive feature elimination method seems to perform better than the univariate ranking method; PLR tends to select fewer genes than the SVM. Overall, PLR with recursive feature elimination seems to perform the best in both predicting the cancer class and reducing the redundant genes.

The formulation of PLR is given in Section 2. In Section 3, we describe the two gene selection methods, UR and RFE. In Section 4, we apply both PLR and the SVM to three microarray cancer data sets. How to solve PLR using the SMO algorithm is described in the Appendix.

2. PENALIZED LOGISTIC REGRESSION

In standard K-class classification problems, we are given a set of training data $(\vec{x}_1, c_1)$, $(\vec{x}_2, c_2)$, $\ldots (\vec{x}_n, c_n)$, where the input $\vec{x}$ is a p-vector $\vec{x}_i = (x_{i1}, x_{i2}, \ldots x_{ip})^T$, the output c_i is qualitative and assumes values in a finite set $\{1, 2, \ldots, K\}$. We wish to find a classification rule from the training data, so that when given a new input $\vec{x}$, we can assign a class label k from $\{1, 2, \ldots, K\}$ to it. Usually it is assumed that the training data are an independently and identically distributed sample from an unknown probability distribution $P(X, C)$. The conditional probability of a point being in class k given $X = \vec{x}$ is

denoted as $p_k(\vec{x}) = P(C = k|X = \vec{x})$. The Bayes classification rule is given by

$$\text{argmax}_{k\in\{1,2,\dots,K\}}\, p_k(\vec{x}).$$

2.1 *PLR formulation*

The logistic regression model arises from the desire to model the conditional probabilities of the K classes via linear functions in $\vec{x}$, while at the same time ensuring that they sum to one and remain in $[0, 1]$. The model has the form

$$\begin{aligned} p_1(\vec{x}) &= \frac{\mathrm{e}^{f_1(\vec{x})}}{\sum_{k=1}^{K} \mathrm{e}^{f_k(\vec{x})}}, \\ p_2(\vec{x}) &= \frac{\mathrm{e}^{f_2(\vec{x})}}{\sum_{k=1}^{K} \mathrm{e}^{f_k(\vec{x})}}, \\ &\vdots \\ p_K(\vec{x}) &= \frac{\mathrm{e}^{f_K(\vec{x})}}{\sum_{k=1}^{K} \mathrm{e}^{f_k(\vec{x})}}, \end{aligned}$$

where

$$f_k(\vec{x}) = b_{k0} + \sum_{j=1}^{p} b_{kj} x_j = b_{k0} + \vec{b}_k^T \vec{x}, \quad k = 1, 2, \dots, K. \tag{2.1}$$

Notice that $f_1(\vec{x}),\ f_2(\vec{x}), \dots,\ f_K(\vec{x})$ are unidentifiable in this model, for if we add a common $b_0 + \sum_{j=1}^{p} b_j x_j$ to each $f_k(\vec{x}),\ p_1(\vec{x}), p_2(\vec{x}), \dots,\ p_K(\vec{x})$ will not change. To make $f_k(\vec{x})$ identifiable, we consider the symmetric constraint $\sum_{k=1}^{K} f_k(\vec{x}) = 0$, or more explicitly

$$\sum_{k=1}^{K} b_{k0} = 0 \tag{2.2}$$

$$\sum_{k=1}^{K} \vec{b}_k = 0. \tag{2.3}$$

Logistic regression models are usually fit by maximum likelihood. Since $p_k(\vec{x})$ completely specifies the conditional probabilities, the multinomial distribution is appropriate. Given the training set $(\vec{x}_1, c_1)$, $(\vec{x}_2, c_2), \dots, (\vec{x}_n, c_n)$, the negative multinomial log-likelihood is

$$\begin{aligned} &-\sum_{i=1}^{n} \log p_{c_i}(\vec{x}_i) \\ &= \sum_{k=1}^{K} \sum_{c_i=k} [-f_k(\vec{x}_i) + \ln(\mathrm{e}^{f_1(\vec{x}_i)} + \cdots + \mathrm{e}^{f_K(\vec{x}_i)})]. \end{aligned}$$

With expression arrays, it is typical that $p \gg n$ (e.g. $n = 35$, $p = 7000$). The consequent implications on the logistic regression model are:

- The classes are usually linearly separable.

- The log-likelihood achieves a maximum of 0.
- The parameters b_{kj} are not uniquely defined, and indeed many will be infinite.

To avoid this problem, we consider the quadratically-regularized negative log-likelihood (see Hastie *et al.* (2001) for motivation):

$$H = \sum_{i=1}^{n} [-\vec{y}_i^T \vec{f}(\vec{x}_i) + \ln(e^{f_1(\vec{x}_i)} + \cdots + e^{f_K(\vec{x}_i)})] + \frac{\lambda}{2} \sum_{k=1}^{K} \|\vec{b}_k\|^2, \tag{2.4}$$

where $\vec{y}_i$ is a binary K-vector that re-codes the response c_i, with values all zero except a 1 in position k if the class is k, and $\vec{f}(\vec{x}_i) = (f_1(\vec{x}_i), f_2(\vec{x}_i), \ldots, f_K(\vec{x}_i))^T$.

It can be shown that in this quadratically-regularized model, the constraint (2.3) is not necessary any more, for at the minimum of (2.4), $\sum_{k=1}^{K} \vec{b}_k = 0$ is automatically satisfied.

In the microarray cancer diagnosis problem, x_{ij} denotes the expression for gene $j = 1, 2, \ldots, p$ and sample $i = 1, 2, \ldots, n$. Since p is often very large (in the thousands), minimizing (2.4) directly is computationally prohibitive. To reduce the computation, we notice that the score equation of (2.4) is given by

$$\frac{\partial H}{\partial \vec{b}_k} = \lambda \vec{b}_k - X_{n\times p}^T (\vec{y}_k - \vec{p}_k) \tag{2.5}$$

$$= 0, \quad k = 1, \ldots, K \tag{2.6}$$

where $X_{n\times p}$ is a matrix with x_{ij} as the (i, j)th element, $\vec{y}_k$ is a $n \times 1$ vector containing y_{ik}, $i = 1, \ldots, n$, and $\vec{p}_k$ is also a $n \times 1$ vector containing $p_k(\vec{x}_k)$, $i = 1, \ldots, n$. The score equation (2.5)–(2.6) indicates that $\vec{b}_k$ is in the row space of $X_{n\times p}$, or equivalently $\vec{b}_k$ can be written as

$$\vec{b}_k = \sum_{i=1}^{n} a_{ik} \vec{x}_k, \quad k = 1, \ldots, K. \tag{2.7}$$

Let the Euclidean inner product

$$\langle \vec{x}, \vec{x}' \rangle = \sum_{j=1}^{p} x_j x'_j.$$

Then $f_k(\vec{x})$ which minimizes H also has the form

$$f_k(\vec{x}) = b_{k0} + \sum_{i=1}^{n} a_{ik} \langle \vec{x}, \vec{x}_i \rangle, \quad a_{ik} \in \mathcal{R} \tag{2.8}$$

and

$$\|\vec{b}_k\|^2 = \sum_{i,i'} a_{ik} a_{i'k} \langle \vec{x}_i, \vec{x}_{i'} \rangle. \tag{2.9}$$

Thus (2.4) becomes

$$H = \sum_{i=1}^{n} [-\vec{y}_i^T \vec{f}(\vec{x}_i) + \ln(e^{f_1(\vec{x}_i)} + \cdots + e^{f_K(\vec{x}_i)})] + \frac{\lambda}{2} \sum_{k=1}^{K} \sum_{i,i'} a_{ik} a_{i'k} \langle \vec{x}_i, \vec{x}_{i'} \rangle. \tag{2.10}$$

The PLR model is fit by finding the a_{ik} that minimize (2.10), then the b_{kj} can be obtained using (2.7). The number of parameters is reduced from the $(p+1)K$ of (2.4) (pK b_{kj} and K b_{k0}) to the $(n+1)K$ of (2.10) (nK a_{ik} and K b_{k0}), making the computations feasible for reasonably small n.

In two-class classification (i.e. $K = 2$), we showed in Rosset *et al.* (2002) that if the training data are separable, then as $\lambda \to 0$, the probability estimates $p_k(\vec{x}) \to \{0, 1\}$ and the classification boundary given by PLR converges to that of the SVM. Therefore, like the SVM, PLR can also be viewed as a technique for fitting a maximum margin classifier.

2.2 *Computational issues*

Since (2.10) is convex, it is natural to use the Newton–Raphson method to minimize H. In order to guarantee convergence, suitable bisection steps can be combined with the Newton–Raphson iterations. Operational details can be found in Zhu and Hastie (2002). The drawback of the Newton–Raphson method is that in each iteration, an $nK \times nK$ matrix needs to be inverted. Although this can be approximated by a one-step Gauss–Seidel or Jacobian approach (reducing the computation to inverting K $n \times n$ matrices), the computational cost is still high when n or K is large.

Recently Keerthi *et al.* (2002) proposed a dual algorithm for two-class PLR which avoids inverting huge matrices. It follows the spirit of the popular sequential minimal optimization (SMO) algorithm (Platt, 1998). Preliminary computational experiments show that the algorithm is robust and fast. Keerthi *et al.* (2002) describes the algorithm for two-class classification; we have generalized it to the multi-class case. A detailed description of the multi-class case algorithm is given in the Appendix.

3. Feature selection

Besides predicting the correct cancer class for a given tumor sample, another challenge in microarray cancer diagnosis is to identify the relevant genes which contribute most to the classification. In this section, we describe two feature (gene) selection methods.

3.1 *Univariate ranking*

Several proposals have been made for ranking genes in terms of their classification performance. Golub *et al.* (1999) first introduced a ranking criterion for each gene in two-class classification. The criterion is defined as

$$\rho_j = \left| \frac{\bar{x}_j^{(1)} - \bar{x}_j^{(2)}}{\sigma_j^{(1)} + \sigma_j^{(2)}} \right|,$$

where $\bar{x}_j^{(k)}$ and $\sigma_j^{(k)}$ indicate the average and standard deviation of the gene expression levels of gene j for all samples of class k. Later on Dudoit *et al.* (2002) used the ratio of between- to within-sum-of-squares of each gene for the multi-class case:

$$\rho_j = \left| \frac{\sum_{k=1}^{K} n_k (\bar{x}_j^{(k)} - \bar{x}_j)^2}{(n-K)\sigma_j^2} \right|, \tag{3.1}$$

where $\bar{x}_j$ is the overall mean expression levels of gene j, n_k is the number of training samples belonging to class k, and σ_j is the pooled within-class standard deviation for gene j:

$$\sigma_j^2 = \frac{1}{n-K} \sum_{k=1}^{K} (n_k - 1)\sigma_j^{(k)2}.$$

The idea is to rank the genes according to the value of ρ_j, and then use the largest m as features in the PLR model. m would then be regarded as a nuisance parameter that has to be determined.

Recently, Tibshirani *et al.* (2002) proposed the shrunken centroids method for classification. Their soft thresholding approach uses shrinkage and gene selection, integrated into a naive Bayes classifier. It also effectively uses univariate gene selection, based on t-statistics for each gene of each class.

These criteria all implicitly assume orthogonality among the features (genes), because each ρ_j is computed with information about a single gene and does not take into account mutual information between genes.

3.2 *Recursive feature elimination*

Guyon *et al.* (2002) described another gene selection method based on recursive feature elimination (RFE). At each step of the iterative procedure, a classifier is fitted with all the current features, a ranking criterion is computed for each feature, and the feature with the smallest ranking criterion is removed. A commonly used ranking criterion is defined as

$$\Delta P_j = \frac{1}{2}\frac{\partial^2 P}{\partial b_j^2} b_j^2, \tag{3.2}$$

where P is a loss function calculated on the training data, and b_j is the coefficient corresponding to feature j. ΔP_j roughly approximates the sensitivity of P to feature j. For the mean-squared-error classifier ($P = \|y - \vec{b}^T\vec{x}\|^2$) and linear SVMs, $\Delta P_j = b_j^2\|x_j\|^2$ which is similar to a t-statistic, where x_j is the n-vector of sample values for gene j. Assuming that the values of each feature have similar ranges, $\Delta P_j = b_j^2$ is also often used. For computational reasons, it may be more efficient to remove a large number of features at a time, at the expense of possibly degrading the classification performance.

4. RESULTS

In this section, we fit both PLR and the SVM models to three cancer diagnosis microarray data sets. We use the two feature selection methods in Section 3 to reduce the number of genes. For the univariate ranking method (UR), we first use (3.1) to compute a ranking (in the descending order of ρ_j) of all the genes. Then we employ an iterative procedure that goes as the following: we start with fitting both PLR and the SVM models using all the genes, then we remove 10% of the genes in the model that are at the bottom of the ranking, we refit the models with the remaining genes, and iterate. For the recursive feature elimination (RFE) with PLR, at each step of the iteration, we fit the model as in (2.1) and (2.8):

$$\begin{aligned} f_k(\vec{x}) &= b_{k0} + \sum_{i=1}^{n} a_{ik}\langle \vec{x}, \vec{x}_i\rangle \\ &= b_{k0} + \sum_{j=1}^{p} b_{kj} x_j \end{aligned}$$

and eliminate the genes with the overall smallest 10% b_{kj}^2 values. So at each step we may eliminate different number of genes for different k. For the SVM RFE, since we use the one-vs-rest scheme as described in Section 1, the b_{kj}^2 are not comparable for different k, hence at each step, we remove the genes with the smallest 10% b_{kj}^2 values for each class k. The number of genes in the final model is selected by cross-validation and the performance of the final model is evaluated on test samples.

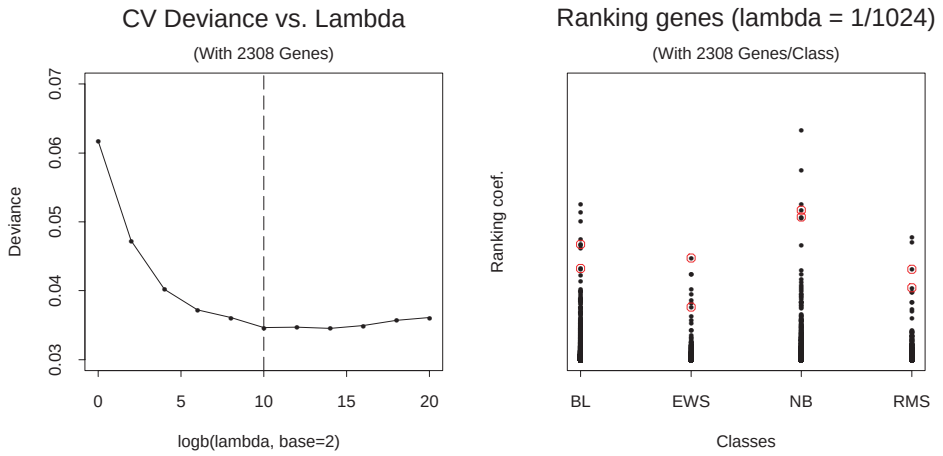

Fig. 1. Choosing λ. SRBCT data. Left plot: cross-validated deviance versus λ. All the genes are used. Right plot: ranking criterion values for each gene of each class. The ones with red circles correspond to the genes selected by PLR RFE.

Before applying PLR, we standardize each sample as is usually done in microarray studies, e.g. Dudoit *et al.* (2002) and Guyon *et al.* (2002), so the mean and standard deviation of the expression levels of each sample are 0 and 1 respectively.

4.1 *Choosing* λ

The regularization parameter λ in (2.4) and (2.10) is chosen by cross-validation without gene selection. The cross-validated deviance is used as the criterion. For example, Figure 1 shows the result for the SRBCT data. The SRBCT data are described in more detail in Section 4.3. The minimum cross-validated deviance corresponds to $\lambda = 1/1024$, hence $\lambda = 1/1024$ is chosen as the regularization parameter. The right panel of Figure 1 plots the values of ranking criterion for each gene of each class based on (3.2) for PLR model. The chosen $\lambda = 1/1024$ is used and all the genes are in the model. We can see the distribution of the ranking criterion values is highly skewed: most of the genes are clustered in the bottom (with small ranking criterion values), only a few are scattered in the top (with large ranking criterion values). This indicates that most of the genes are redundant genes for the classification. The points with red circles correspond to the genes selected by PLR RFE, which is going to be described in Section 4.3. For comparison, we set the regularization parameter for the SVM models the same as that for the PLR models.

4.2 *Leukemia data*

This data set consists of 38 training samples and 34 test samples of two types of acute leukemias, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) (Golub *et al.*, 1999). Each sample is a vector of 7129 genes. 10-fold cross-validation and $\lambda = 1/16$ are used.

The results are plotted in Figure 2 and summarized in Table 1. Notice that for UR, the rankings of genes only depend on the gene expression levels and do not depend on the fitted model, so PLR and the SVM use the same set of genes at each step. However in the RFE, the b_{kj}^2 values depend on the fitted model, so that at each step, PLR and the SVM remove different genes. In the upper part of Figure 2,

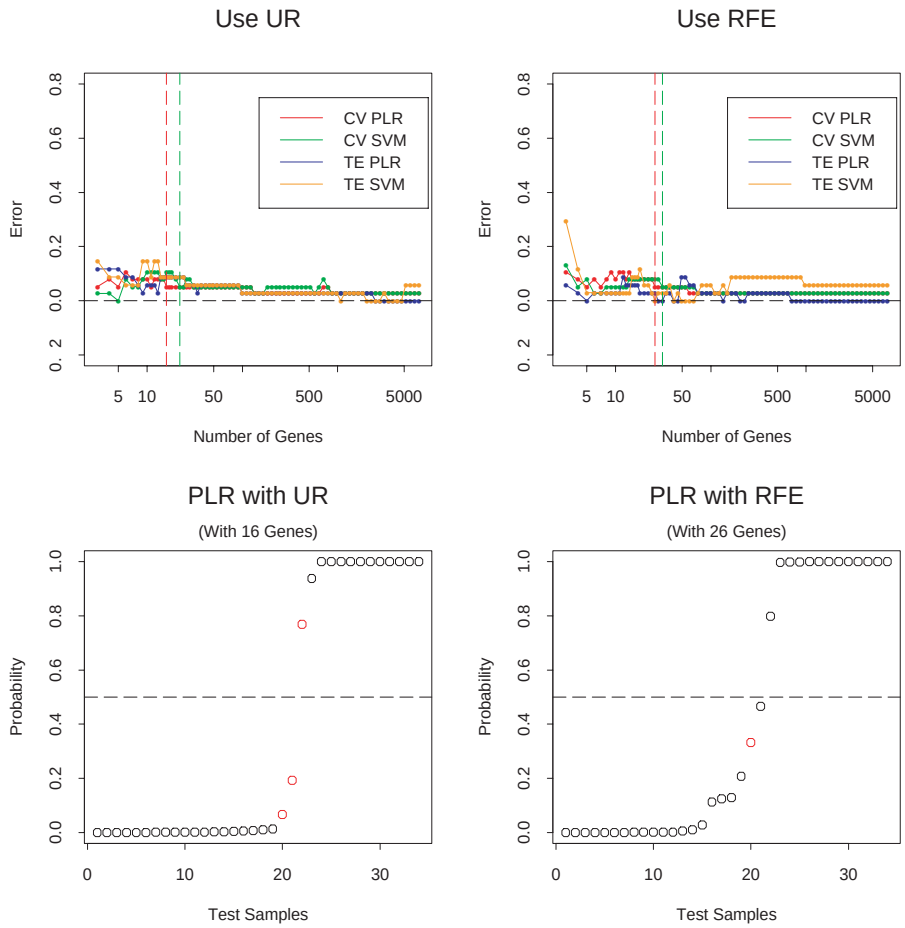

Fig. 2. Leukemia classification. Upper plots: cross-validation and test errors of the SVM and the PLR using the UR and the RFE. Lower plots: estimated probabilities for the test data. The ones with red circles are misclassified ones. The horizontal line is the 0.5 classification decision boundary.

Table 1. *Comparison of leukemia classification methods*

Method	10-fold CV error	Test error	No. of genes
SVM UR	2/38	3/34	22
PLR UR	2/38	3/34	16
SVM RFE	2/38	1/34	31
PLR RFE	2/38	1/34	26

we can see that when using the same set of genes (i.e. using the UR), PLR and the SVM give similar classification results, which agrees with the findings of Rosset *et al.* (2002) and Zhu and Hastie (2002).

The minimum cross-validation error for PLR occurs at 135 genes and 60 genes in the UR and the RFE respectively. To obtain a more manageable set of genes, we sacrifice one more cross-validation error (which only increases the cross-validation error from 1 to 2 in both the UR and the RFE) and this yields 16 genes and 26 genes. The estimated probabilities from the PLR model of the test samples are plotted

in the lower part of Figure 2. The ones with red circles are the misclassified test samples. Apart from the misclassified one, most other samples have good separation between the two classes (i.e. the estimated probability is close to either 1 or 0).

4.3 *SRBCT data*

This data set consists of microarray experiments of small round blue cell tumors (SRBCT) of childhood cancer (Khan *et al.*, 2001). The tumors are classified as Burkitt lymphoma (BL), Ewing sarcoma (EWS), neuroblastoma (NB), or rhabdomyosarcoma (RMS). A total of 63 training samples and 25 test samples are provided. Five of the test samples are actually non-SRBCT. Each sample consists of expression measurements on 2308 genes. 10-fold cross-validation and $\lambda = 1/1024$ are used.

The results are plotted in Figure 3 and summarized in Table 2. As with the leukemia data, we observe that when using the same set of genes (i.e. using the UR), PLR and the SVM give similar classification results, while using the RFE, PLR tends to reduce more genes than the SVM. The PLR model choose 14 genes for using the UR and 8 genes for using the RFE. The estimated probabilities from the PLR model of the test samples are plotted in the lower part of Figure 3. The test error is zero. The ones with violet circles are the maximum estimated probabilities of the five non-SRBCT test samples. Apart from these five samples, most other samples have good separation.

4.4 *Ramaswamy data*

This data set was described in detail by Ramaswamy *et al.* (2001). It consists of 144 training tumor samples and 54 test tumor samples, spanning 14 common tumor classes that account for 80% of new cancer diagnoses in the US. Among these 54 test samples, 8 are metastatic samples. There are 16 063 genes for each sample. 8-fold cross-validation and $\lambda = 1/4$ are used.

The results are plotted in Figures 4 and 5 and summarized in Table 3. The prediction results for this data set are not as good as for the other two data sets. This may be due to the relatively large number of cancer classes ($K = 14$), making it harder to distinguish between classes.

It is worth noting that all four methods choose a large number of genes when compared to the other two data sets, but the RFE method chooses far fewer genes than the UR method. The estimated probabilities of the test samples are plotted in Figure 5. The ones with black circles are the misclassified test samples and the ones with violet circles are the misclassified metastatic test samples. The maximum probabilities of many of these misclassified samples are not far away from their second highest probabilities. We can also see that the breast, renal and pancreas tumor samples are the most difficult samples for classification.

4.5 *Discussion*

PLR with RFE found 26 genes that distinguished AML from ALL with a lower error rate than the methods employed in Golub *et al.* (1999) and Tibshirani *et al.* (2002). The results of Golub *et al.* (1999) and Tibshirani *et al.* (2002) are summarized in Table 4. Our list of 26 genes includes myeloperoxidase and barely missed terminal deoxynucleotidyl transferase. These two genes were not identified in Golub *et al.* (1999) but are known to be excellent markers for AML and ALL, respectively (Tibshirani *et al.*, 2002).

In the SRBCT data set, PLR using RFE found two genes for each class, hence a total of eight genes were identified. This result seems far superior to that of the neural network method of Khan *et al.* (2001), which required 96 genes to obtain zero test error (Table 5). Of our eight genes, seven were also found by the neural network method, except for ‘cold shock domain protein A’. Comparing our eight genes to the 43 genes found in Tibshirani *et al.* (2002), the eight genes are all in the list of Tibshirani *et al.* (2002).

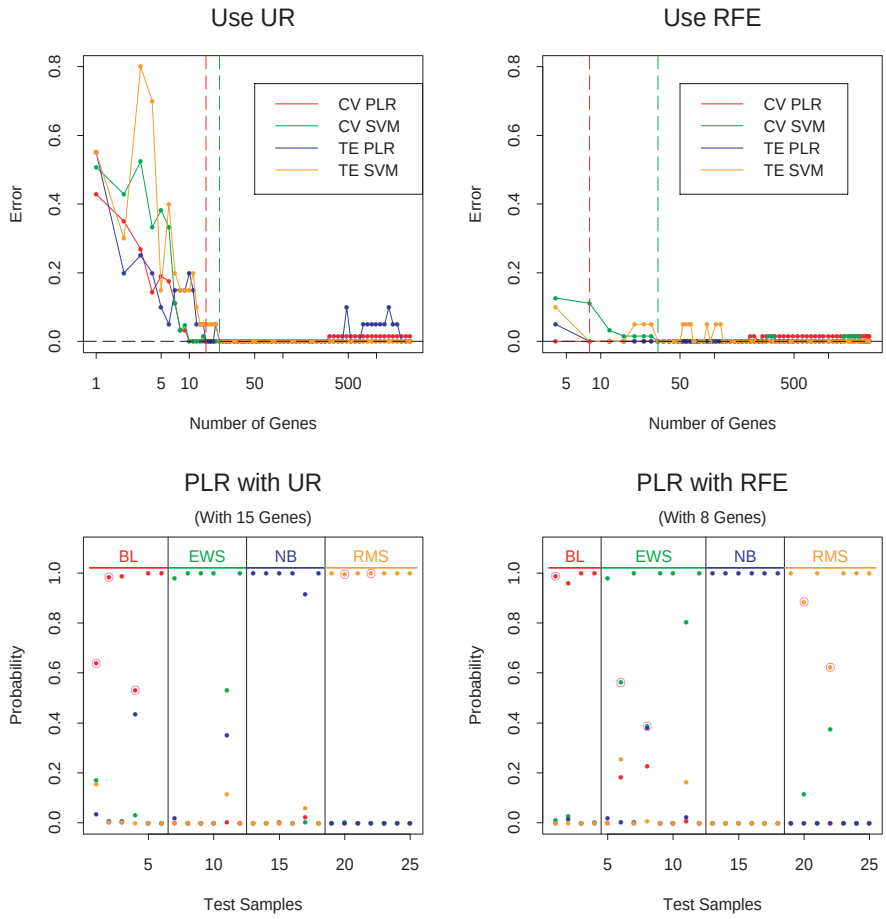

Fig. 3. SRBCT classification. Upper plots: cross-validation and test errors of the SVM and the PLR using the UR and the RFE. Lower plots: estimated probabilities for the test data. The samples are partitioned by the predicted class. All 20 of the test samples are correctly classified. The five non-SRBCT test samples are marked with a circle with its maximum estimated probability. They are below the minimum probabilities of the other test samples in each class.

Table 2. *Comparison of SRBCT classification methods*

Method	10-fold CV error	Test error	No. of genes
SVM UR	0/63	0/20	21
PLR UR	0/63	0/20	15
SVM RFE	0/63	0/20	32
PLR RFE	0/63	0/20	8

This may imply that the simple linear model used by PLR captures the interactions among genes better than the shrunken centroids model.

The Ramaswamy data set has 14 classes. PLR using RFE identified 294 genes. Given the relatively large number of classes ($K = 14$) and the large number of original genes ($p = 16\,063$), this is a sizable

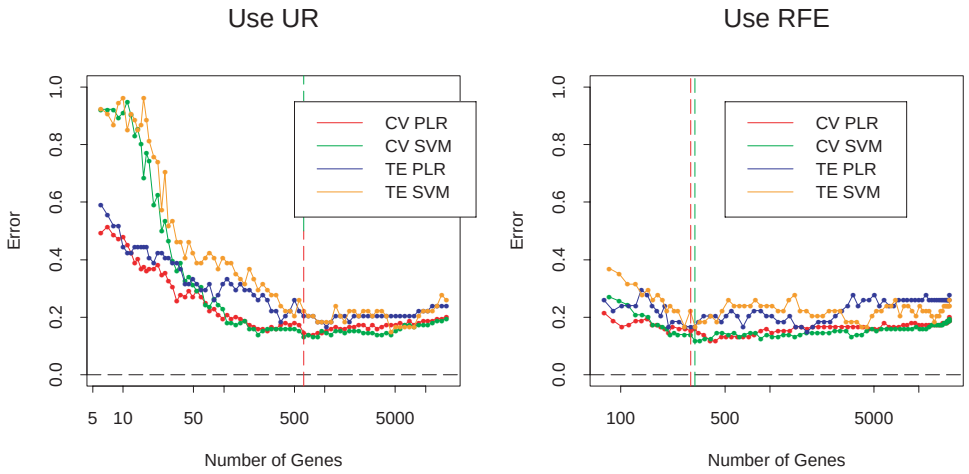

Fig. 4. Ramaswamy classification: cross-validation and test errors of the SVM and the PLR using the UR and the RFE.

reduction. The biological meanings of the genes that were identified need further investigation. Note that Ramaswamy *et al.* (2001) utilized all 16 063 genes in their study to achieve an error rate (12/54).

The tendency that PLR selects fewer genes than the SVM for the studied data sets may be understood heuristically. For the microarray cancer diagnosis problems under study, the training sets are usually linearly separable down to just a few genes. Therefore the solution of the SVM is rather insensitive to the value of the regularization parameter λ in (1.1). Actually when λ is small enough ($\lambda = 1/128$ is usually adequate), the solution of the SVM will be the same for different λ values, which indicates that the regularization does not have much of an effect. This is not the case for PLR. Regularization always shrinks the coefficients b_{kj}, no matter what the value of λ is, and makes the irrelevant genes more likely to be removed. This can be also understood from a fundamental difference between the SVM and PLR. As pointed out in Section 1, the SVM estimates $\text{sign}[p(\vec{x}) - 1/2]$ asymptotically, hence $f(\vec{x})$ is close to -1 or 1, and PLR estimates the logit of probabilities, hence $f(\vec{x})$ is in the range of $-\infty$ and ∞. Therefore, the shrinkage of PLR seems to be more effective.

Although PLR provides an estimate of the class probability, we need to be cautious when using these probability estimates. In the above numerical results, we observe that the misclassified test samples tend to have smaller estimated class probabilities than the correctly classified test samples (see Figures 2 and 5). However, a probability estimate close to 1 does not mean we can be sure about the classification. For example, in Figure 3, we would have expected that the five non-SRBCT samples have relatively small probability estimates (some indeed do), since they do not belong to any of the four tumor classes; but some of these samples have received probability estimates very close to 1. In some sense, estimating the class probability is more difficult than doing classification itself. For example, if we know the true class probability, then we can do classification optimally, but the reverse is not true.

It is also worth noting that, in our paper, the regularization parameter λ is chosen without gene selection. Hence it may not be the optimal regularization when we reduce the number of genes. To search for the optimal regularization parameter λ, we need to consider a much bigger set of possible models, and the number of all possible models grows exponentially fast as the number of iterations increases, which incurs extremely high computational cost. Our approach is a simplified search that reduces the computational cost: i.e. the number of models we consider grows linearly as the number of iterations

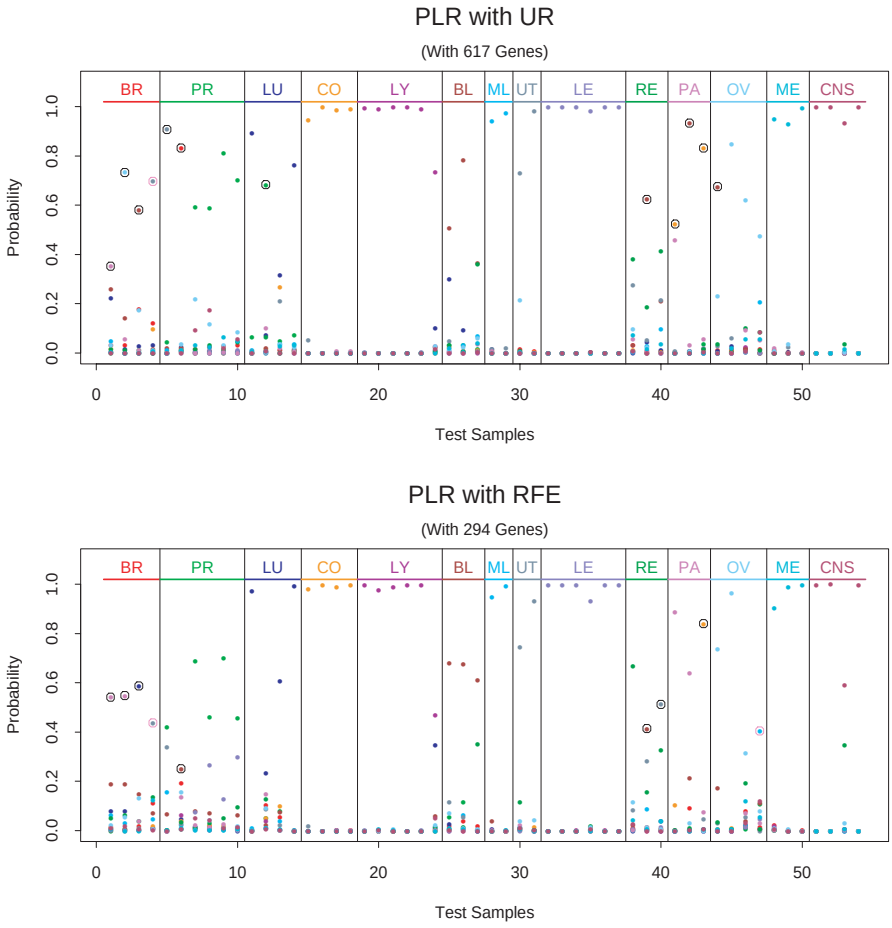

Fig. 5. Ramaswamy classification: estimated probabilities for the test data. The samples are partitioned by the true class. The misclassified test samples are marked with black circles with their maximum estimated probabilities and the two misclassified metastatic test samples are marked with violet circles. The 14 classes correspond to: Breast, Prostate, Lung, Colorectal, Lymphoma, Bladder, Melanoma, Ulterus, Leukemia, Renal, Pancreas, Ovary, Mesothelioma and CNS tumors.

increases. An improvement to our approach might be the following: start with a regularization parameter λ chosen using all the genes, reduce the number of genes, then search for a better regularization parameter using the remaining genes and iterate. We leave this as an open issue that we will investigate in our future work.

5. CONCLUSION

We have proposed penalized logistic regression (PLR) for the microarray cancer diagnosis problem. The empirical results on three data sets show that when using the same set of genes, PLR and the SVM perform similarly in classification, but in addition PLR provides an estimate of the underlying probability. When using the recursive feature (gene) elimination method to select relevant genes for cancer classification, PLR tends to reduce more genes than the SVM.

Table 3. *Comparison of Ramaswamy data classification methods*

Method	8-fold CV error	Test error	No. of genes
SVM UR	19/144	12/54	617
PLR UR	20/144	12/54	617
SVM RFE	17/144	9/54	315
PLR RFE	17/144	9/54	294

Table 4. *Comparison of leukemia classification methods*

Method	10-fold CV error	Test error	No. of genes
Golub *et al.* (1999)	3/38	4/34	50
Tibshirani *et al.* (2002)	2/38	2/34	21
PLR RFE	2/38	1/34	26

Table 5. *Comparison of SRBCT classification methods*

Method	10-fold CV error	Test error	No. of genes
Khan *et al.* (2001)	0/63	0/20	96
Tibshirani *et al.* (2002)	0/63	0/20	43
PLR RFE	0/63	0/20	8

ACKNOWLEDGEMENTS

We thank Susan Holmes, Saharon Rosset, Dylan Small, John Storey and Rob Tibshirani for their helpful comments. Ji Zhu and Trevor Hastie are supported by NSF grant DMS-9803645 and NIH grant ROI-CA-72028-01. Ji Zhu was also partially supported by the Stanford Graduate Fellowship.

APPENDIX A

An SMO algorithm for solving multi-class PLR

The spirit of the multi-class SMO algorithm follows that of the two-class SMO algorithm Keerthi *et al.* (2002), but the details are different.

To minimize (2.4) under (2.2), we can rewrite it as

$$\min \quad P = C\sum_{i=1}^{n} g(\vec{\xi}_i) + \frac{1}{2}\sum_{k=1}^{K} \|\vec{b}_k\|^2, \tag{A.1}$$

$$\text{subject to : } \xi_{ik} = b_{k0} + \vec{b}_k^T \vec{x}_i, \quad \forall i, k, \tag{A.2}$$

$$\sum_{k=1}^{K} b_{k0} = 0, \tag{A.3}$$

where $C = 1/\lambda$ is the regularization parameter, and

$$g(\vec{\xi}_i) = -(y_{i1}\xi_{i1} + \cdots + y_{iK}\xi_{iK}) + \ln(e^{\xi_{i1}} + \cdots + e^{\xi_{iK}}).$$

The Lagrangian for this problem is:

$$L = C\sum_{i=1}^{n} g(\vec{\xi}_i) + \frac{1}{2}\sum_{k=1}^{K}\|\vec{b}_k\|^2 + \sum_{k=1}^{K}\sum_{i=1}^{n} a_{ik}[\xi_{ik} - b_{k0} - \vec{b}_k^T\vec{x}_i] + a_0\sum_{k=1}^{K} b_{k0}, \tag{A.4}$$

where a_{ik} and a_0 are the Lagrangian multipliers. Therefore the KKT optimality conditions are given by

$$\frac{\partial L}{\partial b_{k0}} = a_0 - \sum_{i=1}^{n} a_{ik} = 0, \quad \forall k, \tag{A.5}$$

$$\nabla_{\vec{b}_k} L = \vec{b}_k - \sum_{i=1}^{n} a_{ik}\vec{x}_i = 0, \quad \forall k, \tag{A.6}$$

$$\frac{\partial L}{\partial \xi_{ik}} = C\left(-y_{ik} + \frac{e^{\xi_{ik}}}{\sum_{k'=1}^{K} e^{\xi_{ik'}}}\right) + a_{ik} = 0, \quad \forall i, k, \tag{A.7}$$

which allow us to express $\vec{b}_k$ and ξ_{ik} as functions of a_{ik}:

$$\vec{b}_k = \sum_{i=1}^{n} a_{ik}\vec{x}_i, \quad \forall k, \tag{A.8}$$

$$\xi_{ik} = \ln\left(y_{ik} - \frac{a_{ik}}{C}\right) - \frac{1}{K}\sum_{k'=1}^{K}\ln\left(y_{ik'} - \frac{a_{ik'}}{C}\right), \quad \forall i, k. \tag{A.9}$$

Here we enforce $\sum_{k=1}^{K}\xi_{ik} = 0,\ \forall i$. Note that (A.7) also indicates

$$\sum_{k=1}^{K} a_{ik} = 0, \quad \forall i. \tag{A.10}$$

Hence we have $a_0 = 0$ and $\sum_{i=1}^{n} a_{ik} = 0,\ \forall k$.

Now we apply Wolfe's duality theory to (A.1). The Wolfe dual corresponds to the maximization of L subject to (A.5)–(A.7). Using (A.8) and (A.9) we can simplify the Wolfe dual as

$$\min \quad D = C\sum_{i=1}^{n} G\left(\frac{\vec{a}_i}{C}\right) + \frac{1}{2}\sum_{k=1}^{K}\|\vec{b}_k\|^2, \tag{A.11}$$

$$\text{subject to : } \sum_{k=1}^{K} a_{ik} = 0, \quad \forall i, \tag{A.12}$$

$$\sum_{i=1}^{n} a_{ik} = 0, \quad \forall k, \tag{A.13}$$

where $\vec{a}_i = (a_{i1}, \ldots, a_{iK})^T$, $\vec{b}_k$ is given by (A.8), and

$$G\left(\frac{\vec{a}_i}{C}\right) = \sum_{k=1}^{K}\left(y_{ik} - \frac{a_{ik}}{C}\right)\ln\left(y_{ik} - \frac{a_{ik}}{C}\right).$$

Once the dual problem is solved for a_{ik}, the primal variables $\vec{b}_k$ and ξ_{ik} can be obtained using (A.8) and (A.9).

To solve (A.11), we first replace a_{iK} with $-\sum_{k=1}^{K-1} a_{ik}$, hence (A.11) becomes

$$\min \quad D = C\sum_{i=1}^{n} G\left(\frac{\vec{a}_i}{C}\right) + \frac{1}{2}\sum_{k=1}^{K-1}\|\vec{b}_k\|^2 + \frac{1}{2}\|\vec{b}_K\|^2, \tag{A.14}$$

$$\text{subject to}: \sum_{i=1}^{n} a_{ik} = 0, \quad k = 1, \ldots K-1, \tag{A.15}$$

where $\vec{b}_K = -\sum_{i=1}^{n}\left(\sum_{k=1}^{K-1} a_{ik}\right)\vec{x}_i$ and

$$G\left(\frac{\vec{a}_i}{C}\right) = \sum_{k=1}^{K-1}\left(y_{ik} - \frac{a_{ik}}{C}\right)\ln\left(y_{ik} - \frac{a_{ik}}{C}\right) + \left[1 - \sum_{k=1}^{K-1}\left(y_{ik} - \frac{a_{ik}}{C}\right)\right]\ln\left[1 - \sum_{k=1}^{K-1}\left(y_{ik} - \frac{a_{ik}}{C}\right)\right]. \tag{A.16}$$

Now we can write down the optimality conditions for (A.14). The Lagrangian for (A.14) is then

$$\tilde{L} = C\sum_{i=1}^{n} G\left(\frac{\vec{a}_i}{C}\right) + \frac{1}{2}\sum_{k=1}^{K-1}\|\vec{b}_k\|^2 + \frac{1}{2}\|\vec{b}_K\|^2 - \sum_{k=1}^{K}\left[\beta_k \sum_{i=1}^{n} a_{ik}\right].$$

Define

$$\begin{aligned} F_{ik} &\equiv \vec{b}_k^T \cdot \vec{x}_i - \vec{b}_K^T \cdot \vec{x}_i + C\frac{\partial G}{\partial a_{ik}} \\ &= (\vec{b}_k - \vec{b}_K)^T \cdot \vec{x}_i - \left(\ln\left(y_{ik} - \frac{a_{ik}}{C}\right) - \ln\left[1 - \sum_{k=1}^{K-1}\left(y_{ik} - \frac{a_{ik}}{C}\right)\right]\right). \end{aligned}$$

Then the KKT conditions for (A.14) are

$$\frac{\partial \tilde{L}}{\partial a_{ik}} = F_{ik} - \beta_k = 0 \quad i = 1, \ldots, n; k = 1, \ldots, K-1. \tag{A.17}$$

Define

$$\begin{aligned} i\_up(k) &= \text{argmax}_i F_{ik}, \quad k = 1, \ldots, K-1, \\ i\_low(k) &= \text{argmin}_i F_{ik}, \quad k = 1, \ldots, K-1. \end{aligned}$$

Then the optimality conditions will hold at given a_{ik} if and only if

$$F_{i\_up(k),k} = F_{i\_low(k),k} \quad k = 1, \ldots, K-1. \tag{A.18}$$

Note that to make the F_{ik} appropriately defined, the a_{ik} must satisfy

$$\begin{cases} 0 < a_{ik} < C & \text{if } y_{ik} = 1 \\ -C < a_{ik} < 0 & \text{if } y_{ik} = 0, \end{cases} \tag{A.19}$$

and

$$0 < \sum_{k=1}^{K-1}\left(y_{ik} - \frac{a_{ik}}{C}\right) < 1 \tag{A.20}$$

If there exists an index pair (i, i') such that

$$F_{ik} \neq F_{i'k} \tag{A.21}$$

for some k, then we can decrease D (A.14) (while maintaining the equality constraint $\sum_{i=1}^{n} a_{ik} = 0$ (A.15)) by adjusting a_{ik} and $a_{i'k}$ only. To see this, we define

$$\begin{aligned}\tilde{a}_{ik}(t) &= a_{ik} + t,\\ \tilde{a}_{i'k}(t) &= a_{i'k} - t,\\ \tilde{a}_{i'k''}(t) &= a_{i'k''} \quad \text{for all others.}\end{aligned}$$

Then one can verify that

$$\frac{\mathrm{d}D}{\mathrm{d}t} = F_{ik} - F_{i'k},$$

where F_{ik} and $F_{i'k}$ are evaluated at t. Since by (A.21), $F_{ik} - F_{i'k} \neq 0$ at $t = 0$, a decrease in D is possible by choosing t suitably away from 0.

The SMO algorithm can now be described as follows:

(B1) Choose a^0 satisfying conditions (A.15) and (A.19). Set $r = 1$.
(B2) If a^r satisfies (A.18), stop. If not, let

$$a_{i\_low,k^*} = a^r_{i\_low,k^*} + t, \tag{A.22}$$
$$a_{i\_up,k^*} = a^r_{i\_up,k^*} - t, \tag{A.23}$$
$$a_{i,k} = a^r_{i,k} \quad \text{for other } i, k. \tag{A.24}$$

Find t^* which minimizes the dual D (A.14).
(B3) Update a^{r+1} according to (A.22)–(A.24). Go back to step (B2).

Notice that the step (B2) is a univariate optimization problem, and updating formulae can be used to calculate F_{ik} and D.

REFERENCES

DUDOIT, S., FRIDLYAND, J. AND SPEED, T. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association* **97**, 77–87.

GOLUB, T., SLONIM, D., TAMAYO, P., HUARD, C., GAASENBEEK, M., MESIROV, J., COLLER, H., LOH, M., DOWNING, J. AND CALIGIURI, M. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**, 531–536.

GUYON, I., WESTON, J., BARNHILL, S. AND VAPNIK, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning* **46**, 389–422.

HASTIE, T., TIBSHIRANI, R. AND FRIEDMAN, J. (2001). *The Elements of Statistical Learning*. Berlin: Springer.

KEERTHI, S. S., DUAN, K., SHEVADE, S. K. AND POO, A. N. (2002). A fast dual algorithm for kernel logistic regression. *19th International Conference on Machine Learning*.

KHAN, J., WEI, J., RINGNER, M., SAAL, L., LADANYI, M., WESTERMANN, F., BERTHOLD, F., SCHWAB, M., ANTONESCU, C. AND PETERSON, C. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* **7**, 673–679.

KIMELDORF, G. AND WAHBA, G. (1971). Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 82–95.

LEE, Y. AND LEE, C.K. (2002). Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Technical Report 1051*. Department of Statistics, University of Wisconsin, Madison, WI.

MUKHERJEE, S., TAMAYO, P., SLONIM, D., VERRI, A., GOLUB, T., MESIROV, J. AND POGGIO, T. (1999). Support vector machine classification of microarray data. *Technical Report AI Memo 1677, MIT*.

PLATT, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. *Technical Report MSR-TR-98-14, Microsoft Research*.

RAMASWAMY, S., TAMAYO, P., RIFKIN, R., MUKHERJEE, S., YEANG, C., ANGELO, M., LADD, C., REICH, M., LATULIPPE, E., MESIROV, J., POGGIO, T., GERALD, W., LODA, M., LANDER, E. AND GOLUB, T. (2001). Multiclass cancer diagnosis using tumor gene expression signature. *Proceedings of the National Academy of Sciences* **98**, 15149–15154.

ROSSET, S., ZHU, J. AND HASTIE, T. (2002). Boosting as a regularized path to a maximum margin classifier. *Technical Report*. Department of Statistics, Stanford University, Stanford, CA 94305.

TIBSHIRANI, R., HASTIE, T., NARASIMHAN, B. AND CHU, G. (2002). *Proceedings of the National Academy of Sciences* **99**, 6567–6572.

ZHU, J. and HASTIE, T. (2002) Kernel logistic regression and the import vector machine. *Advances in Neural Information Processing Systems* **14**.

[*Received November 20, 2002; revised January 14, 2004; accepted for publication January 16, 2004*]