

Statistical Applications in Genetics and Molecular Biology

Volume 6, Issue 1

2007

Article 6

Sparse Logistic Regression with Lp Penalty for Biomarker Identification

Zhenqiu Liu* Feng Jiang[†] Guoliang Tian[‡]
Suna Wang** Fumiaki Sato^{††}
Stephen J. Meltzer^{‡‡} Ming Tan[§]

*University of Maryland, zliu@umm.edu

[†]University of Maryland, fjiang@som.umaryland.edu

[‡]University of Maryland, gtian2@umm.edu

**University of Maryland School of Medicine, swang@medicine.umaryland.edu

^{††}Johns Hopkins University School of Medicine, fsato1@jhmi.edu

^{‡‡}Johns Hopkins University School of Medicine, smeltzer@jhmi.edu

[§]University of Maryland Greenebaum Cancer Center, mtan@umm.edu

Sparse Logistic Regression with L_p Penalty for Biomarker Identification*

Zhenqiu Liu, Feng Jiang, Guoliang Tian, Suna Wang, Fumiaki Sato, Stephen J. Meltzer, and Ming Tan

Abstract

In this paper, we propose a novel method for sparse logistic regression with non-convex regularization L_p ($p < 1$). Based on smooth approximation, we develop several fast algorithms for learning the classifier that is applicable to high dimensional dataset such as gene expression. To the best of our knowledge, these are the first algorithms to perform sparse logistic regression with an L_p and elastic net (L_e) penalty. The regularization parameters are decided through maximizing the area under the ROC curve (AUC) of the test data. Experimental results on methylation and microarray data attest the accuracy, sparsity, and efficiency of the proposed algorithms. Biomarkers identified with our methods are compared with that in the literature. Our computational results show that L_p Logistic regression ($p < 1$) outperforms the L_1 logistic regression and SCAD SVM. Software is available upon request from the first author.

KEYWORDS: sparse logistic regression, L_p penalty, feature selection, microarray analysis

*The authors thank the reviewers and associate editor for their constructive comments and acknowledge partial support by US National Institutes of Health grants CA119758 and CA85069.

1 INTRODUCTION

As massive data are available in bioinformatics and other disciplines, it is very important to develop new methods for feature (variable) selection. Feature selection can help to facilitate the data visualization and data understanding, lessen the measurement and storage requirements, reduce the training and utilizing times, and defy the curse of dimensionality to improve the prediction performance. Therefore feature selection is a common method for improving classifier generalization by counteracting the effects of overfitting. The general idea is to reduce the dimensionality of the dataset before classification is performed. There have been many approaches for feature selection of microarray data. Generally, these fall into three categories: filter, wrapper, or embedded methods. Filter approaches select feature subsets based on properties of the data, independent of the statistical learning method (Bo and Jonassan, 2002; Chow et al, 2001; Kerr et al, 2000; Long et al, 2001; Newton et al, 2001; Yu and Chen, 2005). A weakness of the filter methods is that it examines each feature in isolation, ignoring the possibility that groups of features may have a combined effect which does not necessarily follow from the individual performance of features in the group (Pavlidis et al, 2001). This is a common issue with statistical methods such as the T-test, which examine each feature in isolation.

Wrapper methods, on the other hand, wrap around a particular learning algorithm which is used to assess the selected feature subsets in terms of the estimated classification errors and to build the final classifier (Kohavi and John, 1997, Inza et al 2002). Wrapper methods can notably reduce the number of features and significantly improve the classification accuracy (Rivals and Personnaz, 2003; Monari and Dreyfus, 2000). However one drawback with wrapper methods is their computational intensity.

Embedded methods perform the variable selection as part of the statistical learning procedure. They are much more efficient computationally than wrapper methods with the similar performance. Embedded methods have drawn much of the attention recently in the literature. An important embedded technique called LASSO was proposed by Tibshirani (1996, 1997). LASSO is a penalized least square method imposing a L_1 penalty on regression coefficients. Due to the nature of L_1 penalty, LASSO does both continuous shrinkage and automatic variable selection simultaneously. The LASSO type algorithms have been extended to classification by combining logistic regression with the regularized Laplacian prior (Krishnapuram, *et al.* , 2005). The sparsity promoting property of Laplacian prior is theoretically well justified (Donoho & Elad, 2003) and has been found practically and conceptually useful (Chen, Donoho,

& Saunders, 1998). From the regularization prospective, Laplacian prior is the same as L_1 regularization. It plays a key role for feature selection by encouraging the coefficient estimators be either significantly large or exactly zero, which has the effect of automatically removing the irrelevant features from the model. Theoretically, when sample size $n \gg m$, logistic regression with L_p ($p < 1$) gives asymptotically unbiased estimates of the nonzero parameters while shrinking the estimates of zero (or small) parameters to zero (Knight and Fu, 2000). Hence it in principle outperforms the LASSO type of models with L_1 regularization, which are asymptotically biased. However there is a lack of efficient computational methods for such sparse classification.

In this paper, we address two issues related to sparse logistic regression with nonconvex penalty L_p ($p < 1$). First we propose novel algorithms for feature selection and classification. To our knowledge, these are the first algorithms to deal with L_p regularization in the classification framework. Second, the regularization parameters are decided through maximizing the area under the ROC curve (AUC) of the test data. Statistically speaking, the AUC of a classifier is the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. Larger AUC values indicate better classifier performance across the full range of possible thresholds. AUC is a statistically consistent and a more discriminating measure than prediction error (Ling and Huang, 2003). One of the most important objectives is to check whether L_p logistic regression performs better than L_1 logistic regression and SCAD SVM in the situation of $m \gg n$. Our experiments with gene expression and methylation data show that our algorithms can be applied to biological data of high dimension and low sample size.

Support vector machines (Vapnik, 1995) has been a popular tool in high dimensional data analysis. Different embedded methods have been introduced for feature selections (Tipping 2001). Bradley and Mangasarian (1998) proposed the L_1 SVM for feature selection. Zhang *et al.* (2005) suggested to combine SVM with SCAD nonconvex penalty for feature selections, The SCAD penalty was first proposed by Fan and Li (2001) and shown to have better theoretical property than the L_1 penalty. Since SCAD SVM performed better than SVM with L_1 penalty as shown by Zhang *et al.* (2005), the performance of our methods is compared with that of SCAD SVM.

2 METHODS

A general binary classification problem may be simply described as follows. Given n samples, $D = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, where \mathbf{x}_i is a multidimensional

input vector with dimension m and class label $y_i \in \{-1, 1\}$, find a classifier $f(\mathbf{x})$ such that for any input \mathbf{x} with class label y , $f(\mathbf{x})$ predict class y correctly. The logistic regression is:

$$P(y = \pm 1 | \mathbf{x}, \mathbf{w}) = g(y\mathbf{w}^T \mathbf{x}) = \frac{1}{1 + \exp(-y\mathbf{w}^T \mathbf{x})},$$

where $\mathbf{w} = (w_1, \dots, w_m)^T$ are the parameters to be estimated. The log likelihood is

$$l(\mathbf{w} | D) = - \sum_{i=1}^n \log(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)).$$

Different prior assumptions in the maximum a-posteriori (MAP) estimation will lead to different regularization terms. The sparse parameter estimates can be achieved with Laplacian *prior* on w_j .

$$P(w_j | \lambda) = \frac{\lambda}{2} \exp(-\lambda |w_j|).$$

We assume that the components of \mathbf{w} are independent and hence the overall prior for \mathbf{w} is the product of the priors for its components. The MAP estimate maximizes:

$$l_{lasso}(\mathbf{w} | D) = l(\mathbf{w} | D) - \lambda \sum_{j=0}^m |w_j|.$$

i.e.,

$$\hat{\mathbf{w}}_{MAP} = \arg \max_{\mathbf{w}} l_{lasso}(\mathbf{w} | D)$$

Frank and Friedman (1993) proposed an *prior* named super Laplacian

$$P(w_j | \lambda) = C \exp(-\lambda |w_j|^p),$$

where C is a normalization factor, can also lead to sparse parameter estimate. However, to the best of our knowledge, there is no algorithm proposed to deal with classification problems up to now. The corresponding MAP estimate maximizes

$$l_p(\mathbf{w} | D) = l(\mathbf{w} | D) - \lambda \sum_{j=0}^m |w_j|^p,$$

with $L_p = \sum_{j=0}^m |w_j|^p$ as the regularization term.

Figure 1 gives some insight into why L_p norms with $p < 1$ favors sparse solutions with only two parameters. The plot in left-down corner shows the level sets of L_p norms of a two-dimensional vector. For a fixed L_2 norm, all

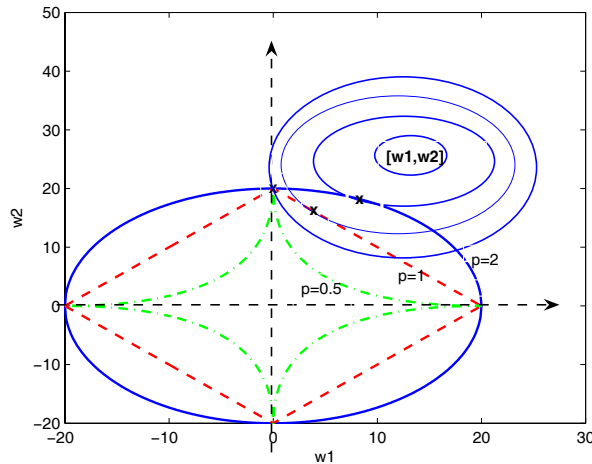


Figure 1: Geometric description of the logistic regression. The plot in left down corner is the L_p with $p = 2, 1, 0.5$ respectively and the upper right corner is the contour of negative log likelihood function. The solution with L_p penalty of different p 's exists at the point where the contours of negative log likelihood and that of the L_p norm first intersect to each other.

vectors lie on the circle with the fixed radius, while L_p norms with $p \leq 1$ are minimized on the coordinate axis. The upper right corner is the contour of negative log likelihood function. The solution with L_p penalty of different p 's exists at the point where the contours of negative log likelihood and that of the L_p norm first intersect. It is observed that the L_p with $p = 0.5$ constrains the coefficients more than LASSO and leads to more sparse solutions for the given plot. Another observation from Figure 1 is that L_p norm is not convex when $p < 1$, so it is not guaranteed to achieve global minima by gradient based optimization methods. Apart from non-convexity, the other difficulty with using the L_p regularization is that it is not differentiable at 0. Therefore we have to consider a differentiable approximation of the L_p norm. Differentiable approximations typically have a parameter which controls the trade-off between the smoothness of the approximation and the closeness of the non-differentiable function which is being approximated. One approximation which works for $p \leq 1$ is

$$L_p = \sum_{j=0}^m (|w_i|^2 + \gamma)^{p/2},$$

where γ is the smoothing parameter. With this differentiable approximation,

we get the following modified MAP function:

$$l_p(\mathbf{w}|D) \approx l_\gamma(\mathbf{w}|D) = l(\mathbf{w}|D) - \lambda \sum_{j=0}^m (|w_j|^2 + \gamma)^{p/2}.$$

Note that $l_\gamma(\mathbf{w}|D) \rightarrow l_p(\mathbf{w}|D)$ as $\gamma \rightarrow 0$.

Given a small value γ , the gradient can be calculated as:

$$\nabla_{\mathbf{w}} l_\gamma = \sum_i (1 - g(y_i \mathbf{w}^T \mathbf{x}_i)) y_i \mathbf{x}_i - \lambda \nabla_{\mathbf{w}} L_p, \quad (1)$$

where

$$\nabla_{\mathbf{w}} L_p = \text{vec} \left\{ \frac{p w_i}{(|w_i|^2 + \gamma)^{1-p/2}} \right\},$$

where $\text{vec}\{\cdot\}$ represents a vector whose i -th element is given by the expression inside the brackets. The Hessian of the objective function is:

$$H = \nabla_{\mathbf{w}\mathbf{w}} l_\gamma = - \sum_i g(\mathbf{w}^T \mathbf{x}_i) (1 - g(\mathbf{w}^T \mathbf{x}_i)) \mathbf{x}_i \mathbf{x}_i^T - \lambda \nabla_{\mathbf{w}\mathbf{w}} L_p, \quad (2)$$

where

$$\nabla_{\mathbf{w}\mathbf{w}} L_p = \text{diag} \left\{ \frac{p}{(|w_i|^2 + \gamma)^{1-p/2}} + \frac{p(p-2)|w_i|^2}{(|w_i|^2 + \gamma)^{2-p/2}} \right\},$$

where $\text{diag}\{\cdot\}$ is the diagonal matrix whose i -th diagonal element is given by the expression inside the brackets. Let

$$A = \text{diag}\{g(\mathbf{w}^T \mathbf{x}_i) (1 - g(\mathbf{w}^T \mathbf{x}_i))\},$$

we have the matrix form of H:

$$H = -XAX^T - \lambda \nabla_{\mathbf{w}\mathbf{w}} L_p \quad (3)$$

With equation (1) and (3), we may estimate the parameters with Newton's method with

$$\mathbf{w}_{new} = \mathbf{w}_{old} - H^{-1} \nabla_{\mathbf{w}} l_\gamma. \quad (4)$$

We run the iteration until $|\mathbf{w}_{new} - \mathbf{w}_{old}| < \delta$, where $\delta > 0$ is a small number. Other algorithms such as fixed-Hessian, conjugate gradient may also be employed to solve the above problem. The advantage with Newton's method is that it converges very fast when near the optimal solution. This algorithm converges from any initialization and to a local maximum is guaranteed.

The proposed methods can be easily extended to a nonlinear model with the kernel trick. We first transform the input data to a new feature space and then build a linear logistic regression in the feature space. Lineae model in the feature space is equivalent to a nonlinear model in the input space. The probability function with kernel is

$$P(y = \pm 1 | \mathbf{x}, \mathbf{w}) = g \left(\sum_{i=1}^n w_i K(\mathbf{x}_i, \mathbf{x}) \right).$$

We can estimate $\mathbf{w} = [w_1, w_2, \dots, w_n]'$ through maximizing the following objective function:

$$\sum_{j=1}^n \log g \left(\sum_{i=1}^n w_i k(\mathbf{x}_i, \mathbf{x}_j) \right) - \lambda L_p$$

It is observed that the kernel logistic regression leads to sparse in kernels instead of variables and linear classifier often give better performance than nonlinear ones when $m \gg n$ (Hastie *et al.*, 2001), even though nonlinear methods are known to be more flexible. Therefore, we will not explore this topic any further. Readers interested may refer to the paper by Liu *et al.* (2005).

Elastic Net in Sparse Logistic Regression

In gene (feature) selection problem, when genes share the same biological pathway, the correlation between them can be high (Segal and Conklin 2003) and those genes forms a group. The ideal gene selection methods will eliminate the trivial genes and automatically include the whole group into the model once one gene among them is selected. LASSO fails the task as it can only select a small subset of independent genes. Zou and Hastie (2005) proposed an new regularization term named the elastic net penalty $((1 - \alpha)L_1 + \alpha L_2$, where $\alpha \in [0, 1)$), which is a convex combination of the L_1 and L_2 penalty. They showed that a group of high correlated features can be selected with elastic net in the regression framework.

We will adapt this approach into sparse logistic regression with L_p penalty. To be convenient, we call the function $L_e = (1 - \alpha)L_p + \alpha L_2$ the same elastic net penalty. This elastic net penalty function is not differentiable and have the characteristics of both L_p and L_2 . These arguments can be seen clearly from Figure 2.

With the L_e penalty and γ approximation, we have the objective function to be maximized:

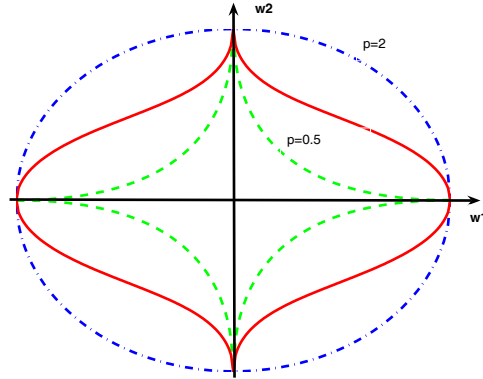


Figure 2: Two-dimensional contour plots (level 1). (---, shape for $p = 2$; - - - -, contour for $p = 0.5$; and —, the contour of elastic net with $\alpha = 0.5$).

$$l_\gamma(\mathbf{w}|D) = l(\mathbf{w}|D) - \lambda(1 - \alpha) \sum_{j=0}^m (|w_j|^2 + \gamma)^{p/2} - \frac{\lambda}{2} \alpha \sum_{j=0}^m |w_j|^2 \quad (5)$$

The first order derivative can be estimated as:

$$\nabla_{\mathbf{w}} l_\gamma = \sum_i (1 - g(y_i \mathbf{w}^T \mathbf{x}_i)) y_i \mathbf{x}_i - \lambda \nabla_{\mathbf{w}} L_e, \quad (6)$$

where

$$\nabla_{\mathbf{w}} L_e = \text{vec} \left\{ (1 - \alpha) \frac{p w_i}{(|w_i|^2 + \gamma)^{1-p/2}} + \alpha w_i \right\}.$$

The corresponding Hessian matrix is given by:

$$H = \nabla_{\mathbf{w}\mathbf{w}} l_\gamma = H = -XAX^T - \lambda \nabla_{\mathbf{w}\mathbf{w}} L_e \quad (7)$$

where

$$\nabla_{\mathbf{w}\mathbf{w}} L_e = \text{diag} \left\{ (1 - \alpha) \left(\frac{p}{(|w_i|^2 + \gamma)^{1-p/2}} + \frac{p(p-2)|w_i|^2}{(|w_i|^2 + \gamma)^{2-p/2}} \right) + \alpha \right\}.$$

The implementation for elastic net is the same as sparse logistic regression with L_p penalty.

Choice of Parameters

Selecting the parameters associated with L_p penalty is very important for improving algorithm performance. Theoretically the lower value of p would lead to better solutions. However when p is very close to zero, difficulties with convergence arise. Therefore in this paper, we set $p = 0.1$.

The smoothing parameter γ appears in the differentiable approximation to the L_p norm. When γ is too large, the approximation is not a good one and the solution is overly smooth and the sparsity property of L_p will be lost. When γ is very small, the number of iterations required for convergence increases drastically. We have found empirically that a choice of γ which does not require very many iterations, and yet converges to very sharp solutions is around 0.001-0.0000001 for our data. Thus $\gamma = 0.0001$ is used in all the experiments of the paper.

The regular parameter λ controls the model parsimony and data fitting. If λ is too small, there will be overfitting and little sparsity. If λ is too large, the produced classifier will be very sparse but have poor predictability. We will decide the optimal λ with the maximization of AUC in the test data.

To prevent the optimization from sticking to local optimal, we randomly initialize the coefficients 20 times and choose the estimated coefficients with the best AUC value for all of the computational experiments in this paper. Our experiments, however, have shown that the computational results are not sensitive to the parameter initialization and the algorithm converges quickly to the same optimization value most of the time with different parameter initializations.

Model Validation with AUC

ROC curve is widely used for visualization and comparison of performance of binary classifiers (Fawcett, 2004). It is the plot of the probability of correctly classifying the positive examples against the rate of incorrectly classifying negative examples. The curve can be interpreted as a comparison of the classifier performance across the entire range of class distributions and error costs. Each data point on the curve is the true positive and false positive pair.

Area under the ROC curve (AUC) is one of the scalar measures for classifier comparison with its value between (0, 1). Larger AUC values indicate better classifier performance across the full range of possible cutoff value. For datasets with skewed class or cost distribution is unknown as in our applications, AUC performs better than logistic regression with test error of validation data.

Given a binary classification problem with n_p positive class samples and

n_n negative class samples, let $f(\mathbf{x})$ be the score function to rank a sample \mathbf{x} . AUC is the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. Mathematically

$$AUC = \frac{\sum_{i=1}^{n_p} \sum_{j=1}^{n_n} I(\mathbf{x}_i, \mathbf{y}_j)}{n_p n_n},$$

where $I(\mathbf{x}_i, \mathbf{y}_j) = 1$ if $f(\mathbf{x}_i) > f(\mathbf{y}_j)$, otherwise $I(\mathbf{x}_i, \mathbf{y}_j) = 0$. AUC has also been pointed out as being the Wilcoxon-Mann-Witney statistic (Rakotomamonjy, 2004).

Both logistic regression and AUC can be extended to multiclass problem with the one-vs-one scheme. Assume C be the set of all classes and $|C|$ be the number of classes, $AUC(c_i, c_j)$ be the area under the two-class ROC curve involving classes c_i and c_j , then we have

$$AUC_{total} = \frac{2}{|C|(|C| - 1)} \sum_{\{c_i, c_j\} \in C} AUC(c_i, c_j).$$

We can see that the summation is calculated over all pairs of distinct classes, irrespective of order and there are total $|C|(|C| - 1)/2$ such pairs. We will choose the best λ value through maximizing AUC.

3 Computational Results

Simulation Data

We simulate a high-dimensional low sample-size data which contains many redundant variables. Four methods are compared. SCAD SVM (Zhang *et al.* 2006), L_1 logistic regression, L_p logistic regression, and L_e logistic regression. A test data set with the size of 200 is used to tune the optimal parameter λ and evaluate the performance of the classifiers. When the estimated parameter $|w_j| < \varepsilon$ for variable x_j , where ε is a preselected small positive threshold value, we can remove variable x_j . The threshold value for removing the variables is set to 0.001 for all of the experiments.

The simulated dataset is randomly generated with input dimension $m = 200$ and only the first three features are relevant. All other features are random noise generated from $N(0, 1)$. The first three features are drawn from a mixture model with center $[0, 0, 0]$ and $[1.5, 1.5, 1.5]$, mixture prior $[0.4, 0.6]$,

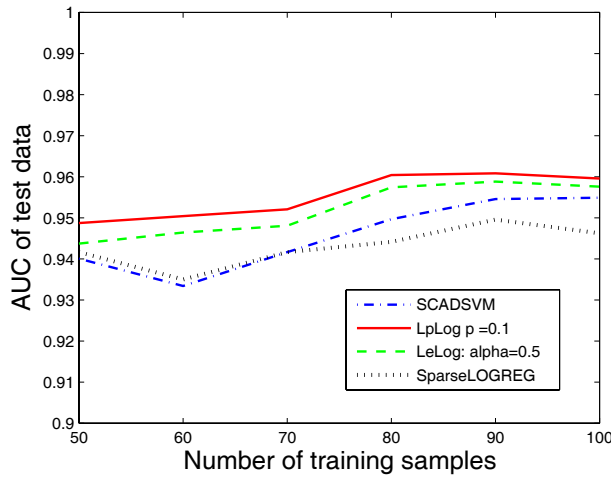


Figure 3: Average AUC values plotted against sample size.

and a common covariance structure

$$\Sigma = \begin{pmatrix} 0.5 & 0.10 & 0.10 \\ 0.1 & 0.5 & 0.45 \\ 0.1 & 0.45 & 0.5 \end{pmatrix}.$$

Features 2 and 3 are highly correlated. We consider various setting for training size $n = 50, 60, 70, 80, 90, 100$. The experiments will repeat 30 times and the average AUC values are plotted. We want to evaluate if the proposed methods can select correct features and if they can select the set of highly correlated variables (grouped variables) together. We run thirty replicates and plot the average AUC values of the test data in Figure 3. Figure 3 shows that sparse logistic regression with L_p ($p=0.1$) penalty consistently outperforms the SCAD SVM, elastic net with $\alpha = 0.5$, and L_1 logistic regression (Shevade & Keerthi, 2003), although the differences become smaller when sample size increases. The software package SparseLOGREG for L_1 logistic regression was downloaded at <http://guppy.mpe.nus.edu.sg/~mpessk/SparseLOG.shtml>.

The AUC values of the test data under different parameter setting are given in Figure 4. The upper left panel in Figure 4 shows the smaller the p the better the performance, although it is not monotonic increase. The upper right panel of Figure 4 shows that the optimal $\lambda = 3 - 4$, where the maximum AUC is achieved. The bottom left panel of Figure 4 gives some insight about how the AUC varies with the number of variables selected. It is clear that the optimal number of variables is 3. The bottom right panel shows the $\log_{10}(\gamma)$

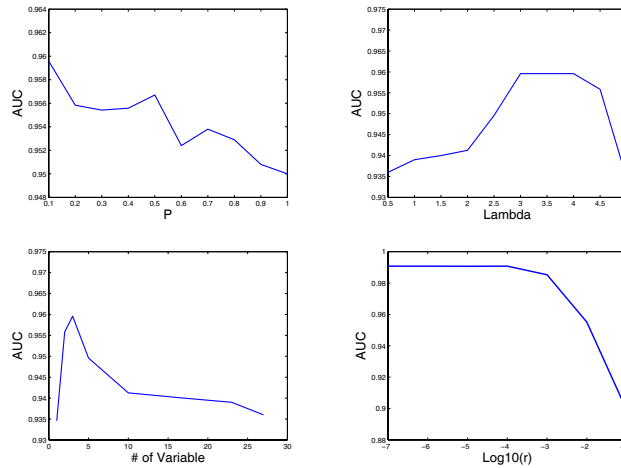


Figure 4: Upper left panel: AUC with different p ; Upper right panel: AUC with different λ ; Bottom left panel: AUC with different number of selected variables; Bottom right panel: AUC with $\log_{10}(\gamma)$.

and corresponding AUC. Apparently the best range of γ to achieve the sparsity and convergence quickly is $0.001 - 0.0000001$.

One of our objectives is to evaluate if the proposed methods can select highly correlated variables simultaneously. Table 1 shows that the average number of variables selected in 30 runs for each method. The values in the parentheses are the standard errors of the corresponding mean values. It is observed that sparse logistic regression with L_p penalty performs the best. This method can find the highly correlated feature 2 and feature 3 simultaneously with reasonable sample size. When $\alpha = 0.5$, L_e tends to select more features than necessary, when sample size is small. SCAD SVM and SparseLOGREG software is able to select the least features on average. The number of features selected with SCAD SVM is diverse according to their standard deviations. 1–5 features are selected in different simulations with the same parameter setting. Obviously both SCAD SVM and SparseLOGREG can not guarantee to select highly correlated features simultaneously, they have the same drawbacks as other LASSO type algorithms.

The frequency of selecting correct features (1-4) in thirty runs is presented in Table 2. Table 2 clearly shows that the sparse logistic regression with L_p penalty performs better than SCAD SVM and SparseLOGREG when the pair-wise correlations are high. SparseLOGREG is implemented using Gauss-Seidel method and selects variables in a forward fashion, it can only select

Table 1: Number of variables selected by various methods and different parameter setting.

| Methods | $n = 50$ | $n = 60$ | $n = 70$ | $n = 80$ | $n = 90$ | $n = 100$ |
|----------------------------|----------|----------|----------|----------|----------|-----------|
| L_p ($p = 0.1$) | 3 | 3 | 2.90 | 3 | 3 | 3 |
| ($\lambda_{opt} = 4$) | (0) | (0) | (0.31) | (0) | (0) | (0) |
| L_e ($\alpha = 0.5$) | 4.65 | 4.43 | 4.27 | 3.98 | 3.17 | 3.08 |
| ($\lambda_{opt} = 26$) | (1.73) | (1.15) | (0.96) | (0.71) | (0.38) | (0.21) |
| SCAD SVM | 1.93 | 1.98 | 2.11 | 2.26 | 2.32 | 2.37 |
| ($\lambda_{opt} = 0.12$) | (0.76) | (0.39) | (0.25) | (0.45) | (0.64) | (0.86) |
| SparseLOGREG | 1.87 | 1.40 | 1.73 | 1.88 | 1.67 | 1.67 |
| | (0.76) | (0.93) | (0.69) | (0.86) | (0.76) | (0.76) |

Table 2: Frequency of selecting exact first three variables in 30 runs.

| Methods | $n = 50$ | $n = 60$ | $n = 70$ | $n = 80$ | $n = 90$ | $n = 100$ |
|--------------------------|----------|----------|----------|----------|----------|-----------|
| L_p ($p = 0.1$) | 30 | 30 | 27 | 30 | 30 | 30 |
| L_e ($\alpha = 0.5$) | 0 | 2 | 10 | 18 | 25 | 27 |
| SCAD SVM | 0 | 1 | 2 | 5 | 10 | 9 |
| SparseLOGREG | 0 | 0 | 2 | 1 | 1 | 0 |

Table 3: The selected CpG regions and model performance.

| Methods | Selected CpG regions | Average AUC (std) |
|--|----------------------|-------------------|
| L_p ($p = 0.1$ & $\lambda_{opt} = 10$) | {1, 2, 3, 5, 6, 7} | 0.8724(0.42) |
| L_e ($\alpha = 0.5$ & $\lambda_{opt} = 3$) | {1, 2, 3, 5, 6, 7} | 0.8683 (0.34) |
| SCAD SVM ($\lambda_{opt} = 0.31$) | {1, 2, 3, 6} | 0.8595 (0.37) |
| SparseLOGREG ($\lambda = 6.5$) | {1, 2, 3, 6, 7} | 0.8561 (0.51) |

independent variables. Elastic net tends to select more features. SCAD SVM makes less than half times of correct selections in all cases and tends to select the mutually independent variables. On average, it selects 2 features more than 20 times out of 30 simulations in different sample size settings.

Real Methylation Data

This methylation data are from 7 CpG regions and 87 lung cancer cell lines (Virmani *et al.* 2002, Siegmund *et al.* 2004). 41 lines are from small cell lung cancer and 46 lines from non-small cell lung cancer. The proportion of positive values for the different regions ranges from 39 to 100% for the small cell lung cancer and from 65 to 98% for the non-small cell lung cancer. The data are available at <http://www-rcf.usc.edu/kims/SupplementaryInfo.html>. We utilize the two-fold cross validation scheme to choose the best λ and test our algorithms. We randomly split the data into two roughly equal-sized subsets and build the classifier with one subset and test it with the other. To avoid the bias arising from a particular partition, the procedure is repeated 100 times, each time splitting the data randomly into two folds and doing the cross validation. The AUC is estimated after each cross validation. The average AUC is given in Table (3). Obviously our models select more genes than do SCAD SVM and SparseLOGREG. Those 6 out of 7 CpG regions selected by Lp logistic regression have been proved to be predictive of lung cancer subtype (Siegmund *et al.* 2004). Our studies have shown that the discarded CpG region (MTHFR) has the lowest correlation coefficient with the cell type (0.22) and lowest par-wise correlation with other CpG regions (0.17 on average). The prediction power (AUC) of the model has increased roughly 5% without CpG region 4 (MTHFR). SCAD SVM selected 4 out of 7 CpG regions most of the time and is more stingy in gene selection.

Table 4: Classification performance (AUC) of different methods for colon data.

| Methods | # of genes selected | $AUC_{total}(std)$ |
|--------------------------|---------------------|---------------------|
| L_p ($p = 0.1$) | 12 | 0.988(± 0.03) |
| L_e ($\alpha = 0.5$) | 23 | 0.980(± 0.05) |
| SCAD SVM | 6 | 0.974(± 0.08) |
| SparseLOGREG | 8 | 0.968(± 0.10) |

Colon Microarray Data

The colon microarray data set (Alon *et al.*, 1999) has 2000 features (genes) per sample and 62 samples which consisted 22 normal and 40 cancer tissues. The task is to distinguish tumor from normal tissues. The data set was first normalized for each gene to have zero mean and unit variance. The transformed data was then used for all the experiments. We employed a same two-fold cross validation scheme to evaluate the model. This computational experiments are repeated 100 times. The *relevance count* concept proposed by Shevade & Keerthi (2003) was utilized to count how many times a gene is selected in the cross validation. Clearly the maximum *relevance count* for a gene is 200 with the two-fold cross validation and 100 repeating. The AUC was calculated after each cross validation. The computational results for performance comparison are reported in Table 4 and Table 5.

It can be seen L_p logistic regression achieves the largest AUC value. Genes selected with L_p logistic regression have 5 genes in common with SCAD SVM and 4 genes in common with SparseLOGREG. The main reason for that genes selected with different methods being partially different is that the classification hypothesis needs not be unique as the samples in gene expression data lie in a high-dimensional space.

Multiclass Gene Expression Data

This gene expression profile data (Wang & Meltzer 2006) comprises 24 normal esophagi (NE), 19 Barrett's esophagus (BE), and 9 esophageal adenocarcinoma (EAC) and there are total 6648 genes. Barrett's esophagus has been long recognized as a key precursor lesion of EAC. The objectives of their research were to clarify the relationship between the stages of neoplastic progression in EAC and pre-EAC and identify the potential biomarkers from BE to EAC and from NE to BE. The data set was pre-processed using standard procedure. Each gene was standardized across the samples to have mean zero and unit

Table 5: Selected top 12 genes with their relevance counts

| Selected Genes | Relev-Count |
|--|-------------|
| Human CRP gene, exons 5 and 6 | 200 |
| H.sapiens mRNA for GCAP-II/uroguanylin precursor | 191 |
| MYOSIN HEAVY CHAIN, NONMUSCLE (Gallus gallus) | 200 |
| human gene for heterogeneous hnRNP core protein A1 | 186 |
| MINERALOCORTICOID RECEPTOR (Homo sapiens) | 168 |
| Human aspartyl-tRNA synthetase alpha-2 subunit mRNA, complete cds. | 174 |
| COLLAGEN ALPHA 2(XI) CHAIN (Homo sapiens) | 198 |
| Human vasoactive intestinal peptide (VIP) mRNA, complete cds. | 165 |
| MYOSIN REGULATORY LIGHT CHAIN 2, SMOOTH MUSCLE ISOFORM (HUMAN) | 158 |
| GELSOLIN PRECURSOR, PLASMA (HUMAN) | 199 |
| Human desmin gene, complete cds. | 175 |
| COMPLEM-FACTOR D PRECURSOR (Homo sapiens) | 169 |

variance. The transformed data was then used for all experiments. Because the tissue samples is relatively small, we would like to make full use of all available samples in gene selection with cross-validation. Ten-fold cross validation was utilized to evaluate the performance of the algorithms. To avoid biases arising from a particular split of the training data, this procedure is repeated 50 times, each time splitting the data differently into 10-fold, using the training samples derived from omitting one of the 10 folds (subsets), and testing the classifier on the samples from the subset which was omitted during training. This procedure gave an unbiased estimate of our gene selection method (Ambrose and McLachlan, 2002). To deal with this multiclass classification problem, we utilized the one-vs-one scheme, which allows us to build classifiers for NE vs BE, NE vs EAC, and BE vs EAC. The AUC was calculated with the prediction value of the whole samples after each 10-fold cross validation. The overall performance was measured with the average AUC of all classifiers. Since it is possible different genes may be selected in the cross validation procedure, we evaluated the performance of the classifiers with the most common selected genes. Table (6) clearly shows that sparse logistic regression has the better performance, although all the methods performed reasonable well. It is very interesting to identify genes responsible for the progressive transition from BE

Table 6: Classification performance (AUC) of different methods.

| Methods | NE vs BE | NE vs EAC | BE vs EAC | AUC_{total} |
|--------------------------|---------------------|---------------------|---------------------|---------------|
| L_p ($p = 0.1$) | 0.969(± 0.12) | 0.992(± 0.05) | 0.988(± 0.1) | 0.983 |
| # of Genes | 35 | 11 | 3 | |
| L_e ($\alpha = 0.5$) | 0.955(± 0.16) | 0.978(± 0.07) | 0.984(± 0.12) | 0.972 |
| # of Genes | 35 | 11 | 5 | |
| SCAD SVM | 0.952(± 0.08) | 0.981(± 0.04) | 0.937(± 0.08) | 0.957 |
| # of Genes | 19 | 14 | 1 | |
| SpLOGREG | 0.947(± 0.05) | 0.976(± 0.10) | 0.953(± 0.07) | 0.958 |
| # of genes | 22 | 13 | 2 | |

to EAC. In the original paper of Wang *et al.* (2006), the classification error rate is 25% with over 50 genes identified. Our proposed methods give AUCs over 98% with only 3-5 genes. The three genes found with L_p logistic regression are KIAA0423, CYR61, YES1. Both CYR61 and YES1 are known biomarkers for esophageal adenocarcinoma. Gene KIAA0423 found with all four methods can be a potential biomarker.

4 Conclusions and Remarks

We have introduced sparse logistic regression with L_p and elastic net nonconvex penalties. Both our simulation and real data analyses show sparse logistic regression with L_p penalty consistently yields a higher AUC value than competing approaches. One interesting finding with our simulation is that sparse logistic regression with L_p penalty is able to select highly correlated features simultaneously. Sparse logistic regression with elastic net may select more features than necessary. This finding seems to be different from LASSO. LASSO can only select independent features. This may be explained by how the algorithm is implemented. LASSO type algorithms are a two-stage procedure: first they find the ridge regression coefficients, and then they do the LASSO type shrinkage along the lasso coefficient paths. It appears that highly correlated features will be filtered out in second stage. Our gradient based algorithms will give the same features the same coefficients with a coordinate-wise Newton update. Elastic net type algorithms deteriorate the classification performance by choosing unnecessary features.

Our proposed methods are superior to SCAD SVM and SparseLOGREG

in the limited experiments. This can be explained from two aspects. First, SCAD SVM is a two-stage implementation similar to LASSO. It first finds the initial coefficients with standard SVM and then shrinks some coefficients to zero through solving a series of linear equations. However the initial values of the algorithm are very important in nonconvex optimization and their initialization with standard SVM might not be optimal. Second, L_p with small p may be superior to SCAD penalty. SparseLOGREG is implemented using Gauss-Seidel method and selects variables in a forward fashion. It can only select independent variables.

In the ‘large m , small n ’ problem, the ‘group of highly correlated variables’ situation is of particularly important concern in the current literature. Our proposed algorithms achieve a simultaneous classification and feature selection with better performance in our limited experiments. Even though there is no method performs universally better than other methods, our methods still have certain advantages as discussed. For example, they are easy to implement without using any additional packages and show great potential for applications in cancer research and association study.

References

- Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ. (1999). “Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays.” *Proceedings of National Academy of Sciences of the United States of American*, 96(12):6745-50.
- Ambrose C, McLachlan GJ. (2002). “Selection bias in gene extraction on the basis of microarray gene-expression data.” *Proc Natl Acad Sci U S A*. 99(10):6562-6.
- Bo T, Jonassen I (2002). “New feature subset selection procedures for classification of expression profiles.” *Genome Biology*, 3(4):0017.
- Bradley PS, Mangasarian OL (1998). “Feature Selection via Concave Minimization and Support Vector Machines.” *ICML 1998*: 82-90.
- Chen SS, Donoho DL, and Saunders MA (1998). “Atomic decomposition by basis pursuit, ” *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61.
- Chow M.L, Moler E.J, and Mian I.S. (2001). “ Identifying marker genes in transcription profiling data using a mixture of feature relevance experts.”

- Physiol. Genomics, Mar; 5:99111.
- Donoho DL and Elad M (2003). “Maximal sparsity representation via l_1 minimization.” *Proc. Nat. Acad. Sci.* 100, pp. 2197–2202.
- Fan J. and Li R. (2001). “Variable selection via nonconcave penalized likelihood and its oracle properties.” *Journal of the American Statistical Association* 96: 1348-1360.
- Fawcett T. (2004). “ROC graphs : Notes and practical considerations for researchers,” Technical report, HP Laboratories, MS 1143, 1501 Page Mill Road, Palo Alto CA 94304, USA.
- Frank IE and Friedman JH (1993). “A statistical view of some chemometrics regression tools.” *Technometrics* 35, pp. 109-148.
- Hastie T., Tibshirani R, and Friedman J. (2001). *The Elements of Statistical Learning*. Springer, New York.
- Inza B, Sierra R, Blanco, and Larra naga P. (2002). “ Gene selection by sequential search wrapper approaches in microarray cancer class prediction.” *Journal of Intelligent and Fuzzy Systems*.
- Kerr MK, Martin M, Churchill GA (2000). “Analysis of variance for gene expression microarray data.” *Journal of Computational Biology*, 7:819-837.
- Knight K. and Fu WJ (2000). “Asymptotics for Lasso-type estimators.” *Annals of Statistics*, 28:1356-1378.
- Kohavi R & John GH (1998). “The Wrapper Approach, in Feature Selection for Knowledge Discovery and Data Mining.” Liu & Motoda (eds.), Kluwer Academic Publishers, pp33-50
- Krishnapuram B, Carin L, Figueiredo M, and Hartemink A (2005). “Sparse multinomial logistic regression: fast algorithms, and generalization bounds.” *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 27, No. 6, pp. 957-968
- Ling X, Huang J. and Zhang H (2003). “ AUC: a Statistically Consistent and more Discriminating Measure than Accuracy.” *Proceedings of IJCAI 2003*.
- Liu Z, Chen D, Xu Y, Li J (2005). “Logistic support vector machines and their

- application to gene expression data.” *International Journal of Bioinformatics Research and Applications* - Vol. 1, No.2 pp. 169 - 182.
- Long A, Mangalam H, Chan B, Toller L, Hatfield G, and Baldi P. (2001). “Improved statistical inference from DNA microarray data using analysis of variance and a Bayesian statistical framework.” *J. Biol. Chem.*, 276, pp.19937-19944.
- Malioutov DM, etin M, and Willsky AS (2005). “ A Sparse Signal Reconstruction Perspective for Source Localization with Sensor Arrays.” *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 3010-3022
- Monari G and Dreyfus G. (2000). “Withdrawing an example from the training set: an analytic estimation of its effect on a nonlinear parameterized model.” *Neurocomputing Letters*, vol. 35, pp.195-201.
- Newton MA, Kendzierski CM, Richmond CS, Blattner FR, Tsui KW (2001). “On differential variability of expression ratios: improving statistical inference about gene expression changes from microarray data.” *Journal of Computational Biology*, 8(1):37-52
- Pavlidis P, Noble WS (2001). “Analysis of strain and regional variation in gene expression in mouse brain.” *Genome Biology*, 2(10): research0042.1-0042.15
- Rakotomamonjy A (2004). “Optimizing AUC with Support Vector Machine (SVM).” *Proceedings of European Conference on Artificial Intelligence Workshop on ROC Curve and AI*, Valencia, 2004.
- Rivals I, Personnaz L (2003). “MLPs (Mono-Layer Polynomials and Multi-Layer Perceptrons) for Nonlinear Modeling.” *Journal of Machine Learning Research* 3: 1383-1398
- Segal MR, Dahlquist KD, Conklin BR. (2003). “Regression approaches for microarray data analysis.” *Journal of Computational Biology*. 10:961-980.
- Siegmund KD, Laird PW, Laird-Offringa IA (2004). “A comparison of cluster analysis methods using DNA methylation data.” *Bioinformatics* 2004, 20:1896-1904.
- Shevade SK, Keerthi SS (2003). “A simple and efficient algorithm for gene

- selection using sparse logistic regression.” *Bioinformatics*. 19(17):2246-53.
- Tibshirani R. (1996). “Regression shrinkage and selection via the lasso.” *J. Royal. Statist. Soc B.*, Vol. 58, No. 1, pages 267-288).
- Tibshirani, R. (1997). “The lasso method for variable selection in the Cox model.” *Statistics in Medicine* 16, 385-395.
- Tipping, M. E. (2001). “Sparse Bayesian learning and the relevance vector machine.” *Journal of Machine Learning Research* 1, 211244.
- Vapnik, V. (1995). “The Nature of Statistical Learning Theory.” Springer-Verlag, New York.
- Virmani AK, Tsou JA, Siegmund KD, Shen LYC, Long TI, Laird PW, Gazdar AF, Laird-Offringa IA (2002). “Hierarchical clustering of lungcancer cell lines using DNA methylation markers.” *Cancer Epidemiology, Biomarkers & Prevention* 2002, 11:291-297
- Wang S, Zhan M, Yin J, Abraham JM, Mori Y, Sato F, Xu Y, Olaru A, Berki AT, Li H, Schulmann K, Kan T, Hamilton JP, Paun B, Yu MM, Jin Z, Cheng Y, Ito T, Mantzur C, Greenwald BD, Meltzer SJ (2006). “Transcriptional profiling suggests that Barrett’s metaplasia is an early intermediate stage in esophageal adenocarcinogenesis.” *Oncogene*. 25(23) : 3346-56.
- Yu J. and Chen X-W. (2005). “Bayesian Neural Network Approaches to Ovarian Cancer Identification from High-resolution Mass Spectrometry Data.” *Bioinformatics*, vol. 21 (suppl-1), pp. i487-i494.
- Zhang HH, Ahn J, Lin X, Park C. (2006). “Gene selection using support vector machines with non-convex penalty.” *Bioinformatics*. 1;22(1):88-95.
- Zou H and Hastie T. (2005). “Regularization and Variable Selection via the Elastic Net.” *JRSSB* 67(2) 301-320.