Computers in Biology and Medicine 41 (2011) 1033–1040

Contents lists available at SciVerse ScienceDirect

Computers in Biology and Medicine

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

An experimental comparison of gene selection by Lasso and Dantzig selector for cancer classification

Songfeng Zheng [a],*, Weixiang Liu [b]

[a] Department of Mathematics, Missouri State University, 901 S. National Ave., Springfield, MO 65897, USA
[b] Biomedical Engineering Lab, School of Medicine, Shenzhen University, Shenzhen, Guangdong 518060, China

ARTICLE INFO

Article history:
Received 10 April 2011
Accepted 30 August 2011

Keywords:
Dantzig selector
Lasso
Gene selection
Cancer classification

ABSTRACT

Selecting a subset of genes with strong discriminative power is a very important step in classification problems based on gene expression data. Lasso and Dantzig selector are known to have automatic variable selection ability in linear regression analysis. This paper applies Lasso and Dantzig selector to select the most informative genes for representing the probability of an example being positive as a linear function of the gene expression data. The selected genes are further used to fit different classifiers for cancer classification. Comparative experiments were conducted on six publicly available cancer datasets, and the detailed comparison results show that in general, Lasso is more capable than Dantzig selector at selecting informative genes for cancer classification.

1. Introduction

Since cancer develops a cell's genetic structure due to mutations to cells with uncontrolled growth patterns, extensive research has shown that we can identify cancer by looking at the genetic level [1,2,13,18,23,24,26,27,32,34,35]. In a microarray study we usually obtain the expression levels of different genes, based on which the decision can be made regarding whether or not this is a patient with cancer.

DNA microarray technique allows us to access the expression profiles of thousands of genes. However, due to the high cost of experiments, we usually have a limited number (50–100) of examples. A classifier built on a small training set with very high dimension is prone to overfitting. Fortunately, it is well-known that most genes are irrelevant to the cancer classification problem [38]. Thus, identifying the genes with strong discriminative power is an important step to effective classification based on the gene expression data.

Gene selection methods based on singular value decomposition [3] or principle component analysis [39] do not use class label information, thus the selected genes might be less effective for classification. To select genes with high discriminative power, most commonly used methods employ class label information, and assign each gene a score based on how well the expression of this gene can discriminate different classes and then select the genes with high scores. Popular score functions include correlation coefficients [13], fold change [4,23], absolute expression level [2,18,24,34], standard deviation [5,34], consistency in repeated data [5,23], correlation coefficient with class labels [24,32], t/F-statistics [12,21], Wilcoxon's rank sum test statistics [7,8], the ratio of between-group over within-group sum square [9], partial least-squares [21], to name a few.

The aforementioned methods select genes one by one, and thus share some disadvantages. First, redundant genes may be selected because the mutual information between genes is not considered. Second, the interactions between genes are omitted [9,14]. In order to avoid selecting redundant genes, some methods have been introduced which can select a set of genes simultaneously. For example, Guyon et al. [14] proposed Recursive Feature Elimination method for Support Vector Machine, which recursively eliminates irrelevant genes for classification so that the survived genes are expected to have strong discriminative power.

In regression analysis, it is often very important to identify informative variables in order to explain the obtained model. In the ordinary least-square regression, if we impose L_1 constraint on the regression coefficients, some of the regression coefficients in the model will shrink to zeros, thus we can automatically select *a set of* informative variables simultaneously (i.e. the ones with nonzero coefficients). This technique is called Lasso (least absolute shrinkage and selection operator) [30]. Dantzig selector [6] is a similar technique for linear regression model: instead of minimizing the squared error, it minimizes the L_∞ norm of the gradient vector of the squared error function. With the L_1 constraint on the regression coefficients, the majority of the estimated regression coefficients by Dantzig selector are exactly zeros. As such, Dantzig

* Corresponding author.
E-mail addresses: SongfengZheng@MissouriState.edu (S. Zheng), wxliu@szu.edu.cn (W. Liu).

0010-4825/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compbiomed.2011.08.011

selector can also give a set of informative variables for a linear regression model. The variable selection performances of Lasso and Dantzig selector were compared in [11,20] for linear regression model.

This paper conducts an experimental study to compare the gene selection ability of Lasso and Dantzig selector for cancer classification. We interpret the class label (0/1) as the probability of this example being positive, then use Lasso and Dantzig selector to regress this probability on the gene data, thus simultaneously select informative genes for expressing the probability of being positive with the gene expression data by a linear equation. We apply the selected genes to three linear models: linear regression, linear support vector machines, and logistic regression. On six publicly available cancer datasets, we tested the classification performances based on the selected genes, and the comparison shows that Lasso is in general more powerful than Dantzig selector at selecting informative genes. When feeding the selected genes to nonlinear SVM with Gaussian kernel, we observe the same pattern. Our result about the gene selection performance of Lasso and Dantzig selector for cancer classification is consistent with the conclusion in statistics literature [11,20] for linear regression.

2. Lasso and Dantzig selector for variable selection

Suppose we have dataset $\{(\mathbf{x}_i,y_i),\ i=1,\ldots,N\}$, with the predictor $\mathbf{x}_i \in \mathbf{R}^p$ and the response $y_i \in \mathbf{R}$. Without loss of generality, we assume the predictors and the response are centered, and the predictors are standardized, that is

$$\sum_{i=1}^{N} y_i = 0, \quad \sum_{i=1}^{N} \mathbf{x}_{ij} = 0 \quad \text{and} \quad \sum_{i=1}^{N} \mathbf{x}_{ij}^2 = 1. \tag{1}$$

Consider the linear regression model

$$y = \mathbf{x}'\boldsymbol{\beta} + \varepsilon,$$

where $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_p)'$ is the vector of regression coefficients.

To fit the linear regression model, the ordinary least-square (OLS) estimates are obtained by minimizing the residual squared error, i.e.

$$\hat{\boldsymbol{\beta}}_{\text{OLS}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{N} (y_i - \mathbf{x}_i'\boldsymbol{\beta})^2 = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_{L_2},$$

where $\mathbf{Y} = (y_1,\ldots,y_N)' \in \mathbf{R}^N$ is the response vector, $\mathbf{X} \in \mathbf{R}^{N\times p}$ is the design matrix, and $\|\cdot\|_{L_2}$ represents the L_2 norm, i.e., for vector $\mathbf{r} = (r_1,\ldots,r_n)'$,

$$\|\mathbf{r}\|_{L_2} = \sum_{i=1}^{n} r_i^2.$$

However, when the dimensionality p is large, the OLS estimate $\hat{\boldsymbol{\beta}}_{\text{OLS}}$ has a large portion of non-zero components, making it difficult to interpret the obtained model.

2.1. Lasso

Lasso [30] is a technique to fit the linear regression model which minimizes the residual squared error with a constraint on the sum of the absolute value of the regression coefficients, i.e.

$$\hat{\boldsymbol{\beta}}_{\text{Lasso}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_{L_2} \quad \text{s.t.} \quad \sum_{j=1}^{p} |\beta_j| \le t. \tag{2}$$

The Lasso penalty in Eq. (2) shrinks the fitted coefficients $\hat{\boldsymbol{\beta}}$ toward zero. If we make t in Eq. (2) small, some of the estimated coefficients $\hat{\beta}_j$'s will be *exactly* zeros. Thus, by tuning the parameter t in Lasso, we can automatically select informative variables (i.e. the ones with nonzero regression coefficients).

Efron et al. [10] proposed Least Angle Regression (LARS) and derived that Lasso is a special case of LARS with simple modification. As analyzed in [10], when there are far more variables than examples, i.e. $p \gg N$, LARS selects at most N variables with $O(N^3)$ operations assuming the training examples are not centralized, otherwise LARS terminates with $N-1$ variables. In this paper, we have far more genes than examples, thus $p \gg N$ is our interested scenario, and we will adopt LARS to find the Lasso solution.

2.2. Dantzig selector

The Dantzig selector [6] estimator of the regression coefficient vector $\boldsymbol{\beta}$ is the solution to

$$\min_{\boldsymbol{\beta} \in \mathbf{R}^p} \sum_{j=1}^{p} |\beta_j| \quad \text{s.t.} \quad \|\mathbf{X}'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\|_{L_\infty} \le s, \tag{3}$$

where $\|\cdot\|_{L_\infty}$ represents the L_∞ norm, i.e., for vector $\mathbf{r} = (r_1,\ldots,r_n)'$, $\|\mathbf{r}\|_{L_\infty} = \max_{1\le i\le n} |r_i|$.

The optimization in Eq. (3) can be recast as a linear programming problem [6]:

$$\min_{\boldsymbol{\beta} \in \mathbf{R}^p, \mathbf{u} \in \mathbf{R}^p} \sum_{i=1}^{p} u_i \tag{4}$$

$$\text{s.t.} \quad -\mathbf{u} \le \boldsymbol{\beta} \le \mathbf{u} \quad \text{and} \quad -s\mathbf{1} \le \mathbf{X}'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) \le s\mathbf{1}, \tag{5}$$

where $\mathbf{u} = (u_1,\ldots,u_p)'$ with $u_i \ge 0$, and $\mathbf{1}$ is a p-dimensional vector of all ones. The above minimization problem produces a large part of estimated coefficients to be exactly 0 in a similar fashion as Lasso and hence can be used as a variable selection tool. Candès and Tao [6] have provided strong theoretical justification for this property, and the Dantzig selector has shown impressive empirical performance on simulated data and real world problems involving large values of p [6,17].

The definition of Dantzig selector can be re-expressed as [11]

$$\hat{\boldsymbol{\beta}}_{\text{DS}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{X}'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\|_{L_\infty} \quad \text{s.t.} \quad \sum_{j=1}^{p} |\beta_j| \le s. \tag{6}$$

Comparing the minimization problem for Lasso in Eq. (2) and that for Dantzig selector in Eq. (6), we can see that with the bound on the absolute sum of the coefficient vector $\boldsymbol{\beta}$, Lasso minimizes the squared error function while Dantzig selector minimizes the maximum component of the gradient of the squared error function. The difference in the problem formulations makes Lasso and Dantzig selector select different variables, and it is claimed in [11,20] that Lasso has slight advantage over Dantzig selector on variable selection for linear regression model. The current paper attempts to compare the two variable selection methods for pattern classification problems.

3. Classifiers

Suppose we have a dataset $\{(\mathbf{x}_i,y_i),\ i=1,\ldots,N\}$, where $\mathbf{x}_i \in \mathbf{R}^p$ and $y_i \in \{0,1\}$. The purpose of supervised learning is to fit a classifier from the training set and apply it to the unseen testing set. In this paper, we consider three linear classifiers, which include Linear Regression as Classifier, linear Support Vector Machines (SVM), and Logistic Regression; we will also apply the selected genes to nonlinear SVM with Gaussian kernel.

3.1. Linear regression as classifier

In the binary classification problems, similar to [29], we treat the class label $y_i \in \{0,1\}$ as the probability of $\mathbf{x}_i$ being a positive example, thus y_i can take continuous values in [0, 1]. We then fit a linear regression model to represent y_i (the probability of being positive) as a linear function of $\mathbf{x}_i$. In the testing stage, we apply the threshold value 0.5 to the predicted value for obtaining the class label.

We notice that [15] suggests name the class label as $y_i \in \{1, -1\}$ and treat it as continuous variable, and use the sign of the predicted value as the class label. There is no difference between the 0/1 labeling system and the ± 1 naming system in the final results because there is a linear relation between them. However, the probabilistic interpretation of the 0/1 labeling system is more natural and easier to understand.

3.2. Support vector machines

In the formulation of Support Vector Machine (SVM), the class label is named as $y_i \in \{1, -1\}$ for convenience. SVM regards the linear classifier sign($\mathbf{x}'\mathbf{w}+b$) as a hyperplane in the p-dimensional space, and chooses the hyperplane so that the distance from it to the nearest data point on each side is maximized. If such a hyperplane exists, it is clearly of interest and is known as the maximum-margin hyperplane. When the training set is not separable, SVM will choose a hyperplane that splits the examples as cleanly as possible, while still maximizing the distance to the nearest cleanly split examples. The method introduces slack variable ξ_i, which measures the degree of misclassification of the datum $\mathbf{x}_i$. The optimization becomes a trade-off between a large margin, and a small error penalty. If the penalty function is linear, the optimization problem becomes

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \tag{7}$$

$$\text{s.t.} \quad y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \xi_i \quad \text{for} \quad 1 \leq i \leq N, \tag{8}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)'$, and C is a parameter to balance the margin of the hyperplane and the errors made on the training set. The optimization can be solved by the dual problem via quadratic programming:

$$\max_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x}_i'\mathbf{x}_j \tag{9}$$

$$\text{s.t.} \quad \sum_{i=1}^{N}\alpha_i y_i = 0 \quad \text{and} \quad 0 \leq \alpha_i \leq C \quad \text{for} \quad 1 \leq i \leq N, \tag{10}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)'$ with α_i being the Lagrangian multiplier of the ith constraint in Eq. (8). Finally, the coefficient vector of the linear classifier can be calculated by

$$\mathbf{w} = \sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i,$$

and b can be estimated from the set of support vectors (with $\alpha_i \neq 0$).

Linear SVM can be extended to nonlinear case by using the kernel trick. Let $\Phi(\mathbf{x})$ be the mapped feature vector for $\mathbf{x}$ in the reproducing kernel Hilbert space, and let $k(\mathbf{x}_i,\mathbf{x}_j) = \Phi(\mathbf{x}_i)'\Phi(\mathbf{x}_j)$ be the kernel function. Nonlinear SVM is learned by fitting the following maximization problem:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i,\mathbf{x}_j), \tag{11}$$

with the same constraints as in Eq. (10). The final nonlinear SVM classifier has the form

$$\hat{y} = \text{sign}\left(\sum_{i=1}^{N}\alpha_i y_i k(\mathbf{x}_i,\mathbf{x}) + b\right),$$

where $\hat{y}$ is the predicted label for the testing object $\mathbf{x}$.

3.3. Logistic regression

For a predictor/feature vector $\mathbf{x}$, we can directly model the posterior probability of the class label $y \in \{0, 1\}$ as logistic response function

$$p(y=1|\mathbf{x}) = \frac{\exp(w_0 + \mathbf{x}'\mathbf{w})}{1 + \exp(w_0 + \mathbf{x}'\mathbf{w})} = p(\mathbf{x}; w_0, \mathbf{w}).$$

The log likelihood function of $\{(\mathbf{x}_i, y_i),\ i = 1, \ldots, N\}$ is

$$l(w_0,\mathbf{w}) = \sum_{i=1}^{N}\{y_i \log p(\mathbf{x}_i; w_0,\mathbf{w}) + (1-y_i)\log(1-p(\mathbf{x}_i; w_0,\mathbf{w}))\}. \tag{12}$$

The parameter $(w_0,\mathbf{w})$ can be obtained by maximizing the log likelihood function $l(w_0,\mathbf{w})$, and the prediction could be made by looking at the sign of $w_0 + \mathbf{x}'\mathbf{w}$.

4. Experiments

This section applies Lasso and Dantzig selector to select the most informative genes for the classifiers introduced in Section 3, and reports the results on six public cancer datasets.

4.1. The datasets

We perform binary classification on six datasets which include DLBCL, Leukemia, Prostate, Colon, Lymphoma, and Estrogen. Table 1 summarizes the basic information about the datasets, and we provide the descriptions of the datasets and the preprocessing procedures as follows:

DLBCL: The DLBCL dataset [26] contains in total 77 examples in two classes, diffuse large B-cell lymphomas (DLBCL) and follicular lymphoma (FL) which have 58 and 19 examples, respectively. The original dataset contains 7129 genes. We thresholded the intensities at 20 and 16,000 units, then we filtered out genes with max/min ≤ 3 or max$-$min ≤ 100. After preprocessing, we obtained a dataset with 77 examples and 6285 genes.

Leukemia: This dataset was first presented in [13], which contains gene expression levels of 72 patients of which 47 with acute lymphoblastic leukemia (ALL) and 25 with acute myeloid leukemia (AML). Following the preprocessing strategy in [9], we processed them by thresholding, filtering, a logarithm transformation, and standardizing each tissue example to have zero mean and unit variance across the genes. The processed data finally contain the expression values of 3571 genes.

Prostate: The Prostate dataset [27] contains in total 102 examples in two classes, i.e. tumor and normal, which have 52 and 50 examples, respectively. The original dataset contains 12,600 genes. In our experiment, intensities were thresholded at 100 and 16,000 units. Then we filtered out the genes with max/min ≤ 5 or max$-$min ≤ 50. After preprocessing, we obtained a dataset with 102 examples and 5966 genes. The DLBCL, Leukemia, and Prostate datasets are available in MATLAB format at http://www.biomedcentral.com/1471-2105/7/228/additional/.

Colon: In this dataset, expression levels of 40 tumors and 22 normal colon tissues for 6500 genes were measured using the Affymetrix gene chip technique. A subset of 2000 genes with

Table 1
The information about the datasets.

Dataset	DLBCL	Leukemia	Prostate	Colon	Lymphoma	Estrogen
# of genes	6285	3571	5966	2000	4026	7129
# of positive	19	25	52	22	62	25
# of negative	58	47	50	40	34	24

highest minimal intensity across the examples has been selected in [2], and the selected subset can be downloaded from http://microarray.princeton.edu/oncology/affydata/index.html. We further process the data as in [8]: applying a logarithm transformation and standardizing each tissue example to have zero mean and unit variance across the genes.

Lymphoma: The Lymphoma dataset [1] contains 62 malignant and 34 normal examples, and each example consists of 4026 gene expression measures. The data are preprocessed to give each feature mean zero and unit standard deviation. This dataset is made publicly available in MATLAB format by [36] at http://www.kyb.tuebingen.mpg.de/bs/people/weston/l0/.

Estrogen: The Estrogen dataset was first presented in [35], and is available at http://data.cgt.duke.edu/west.php. The dataset contains 7129 gene expression values obtained by applying the Affymetrix gene chip technique to 49 breast tumor examples. We thresholded the raw data with a floor of 100 and a ceiling of 16,000 and then applied a logarithm transformation. Finally, each

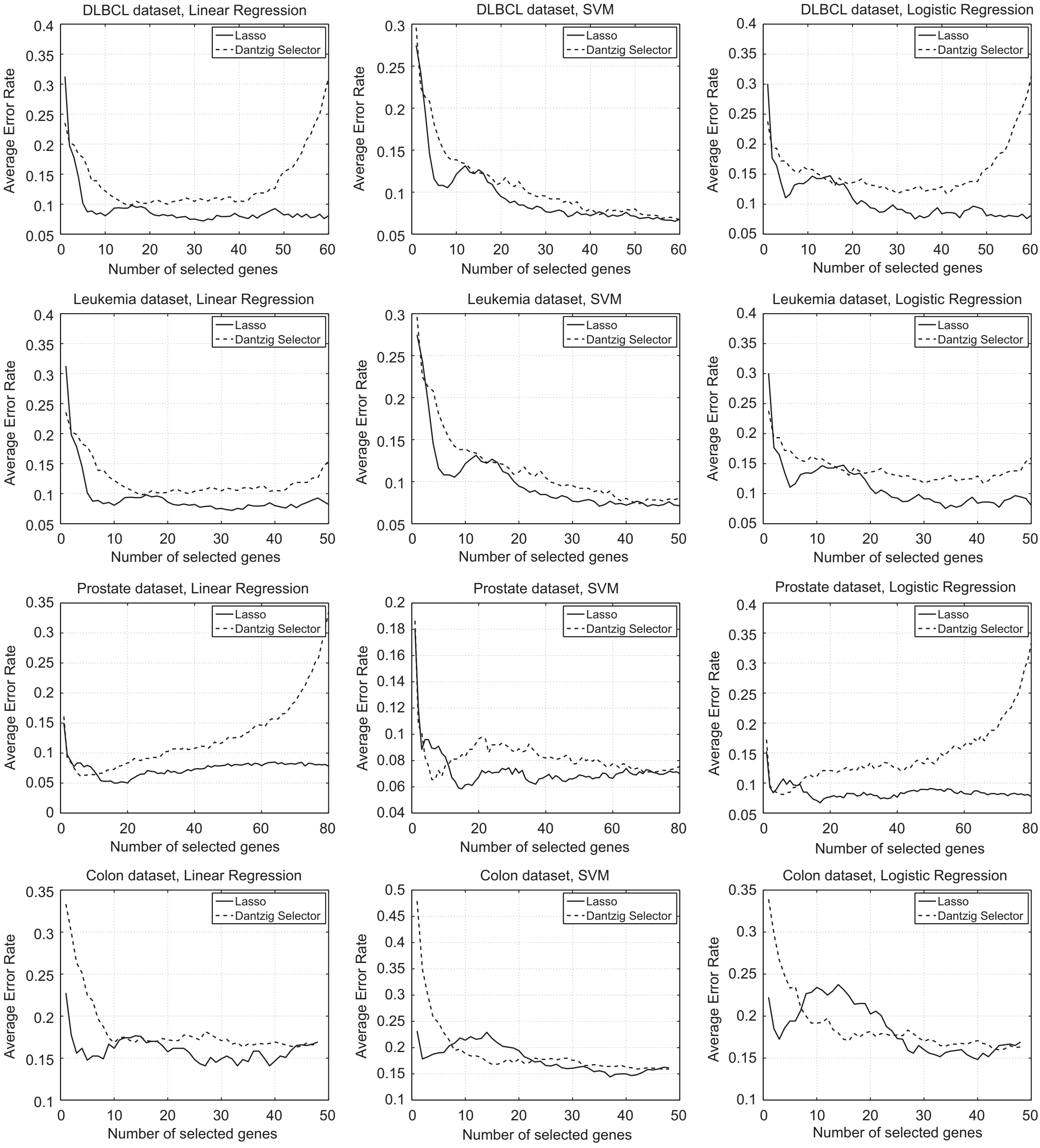


Fig. 1. Average testing error curves with the number of selected genes on the DLBCL, Leukemia, Prostate, and Colon datasets.

tissue example was standardized to have zero mean and unit variance across the genes. The response variable describes the status of the estrogen receptor (ER), and 25 examples are ER+, whereas the remaining 24 examples are ER−.

4.2. Results of linear classifiers

For each dataset, we randomly select 80% of the examples as training set, and the remaining 20% are used for the testing purpose. In the gene selection stage, we let the response $y_i=1$ if the ith patient has cancer, and $y_i=0$ otherwise. As discussed in Section 3.1 (also refer to [29]), we understand the response y_i as the probability of the ith example being cancerous. On the training set, we apply Lasso and Dantzig selector to regress y_i on the gene data, which allow us to select informative genes for expressing the probability of being cancerous with the gene expression data by a linear equation. We then feed the expression values of the selected genes to the training algorithms of the three linear classifiers introduced in Section 3. In the testing stage, only the expression levels of the selected genes are used for decision making. The partition-selection-training-testing process is repeated 100 times.

If Lasso and Dantzig selector pick the same set of genes, the performances of a specific classifier on the gene set selected by Lasso and that selected by Dantzig selector are expected to be the same. In the case of Lasso and Dantzig selector select different sets of genes, this paper studies which gene set can give us better classification result, thus compares the gene selection ability of Lasso and Dantzig selector. To ensure a fair comparison, it is reasonable to fix free parameters in each classifier, rather than tuning the parameters to yield the best result on each dataset. In this paper, we fix the parameter C in SVM as $C=15$.

Changing the parameter t for Lasso in Eq. (2) or the parameter s for Dantzig selector in Eq. (6) can yield different values of the coefficient vectors, thus enables us to observe the evolution of the regression coefficients, which is called the solution path [11,20] of Lasso or Dantzig selector. Starting with $t=0$ or $s=0$, i.e., no gene is selected, we gradually increase the parameters. The continuity of the solution paths of Lasso and Dantzig Selector [11,20] indicates that the number of selected genes increases gradually. That is, if the number of selected genes is fixed, tuning the parameter might improve the performance of the regression, but cannot change the selected genes, thus has no effect on the final classification result because the classifiers are trained on the selected gene set. As such, it is not necessary to tune the parameter (s or t) as long as the number of selected genes by Lasso or Dantzig selector is fixed.

For each number of selected genes, we calculate the average testing error rate for every classifier on each of the six datasets. Figs. 1 and 2 show the average testing error rate curves of the 100 runs for every classifier on the six datasets. From the figures, we observe that on particular datasets (e.g. Colon), for particular classifiers, Dantzig selector has advantage when the number of selected genes is in a certain range, while in all other cases, Lasso based classifiers have better performance in terms of average testing error rate. This shows that in general, compared to Dantzig selector, Lasso is more efficient at picking informative genes for linear classifier.

4.3. Results of nonlinear support vector machines

We also ran the experiment in Section 4.2 with nonlinear SVM with Gaussian kernel function

$$k(\mathbf{x}_i,\mathbf{x}_j)=\exp\left(-\frac{\|\mathbf{x}_i-\mathbf{x}_j\|^2}{2\sigma^2}\right),$$

with $\sigma=1$. As in the experiments in Section 4.2, we fix the penalty parameter $C=15$ in the nonlinear SVM for a fair comparison to linear SVM. Fig. 3 shows the average testing error curves of the 100 runs with different number of selected genes. From Fig. 3, we observe that, same as for linear classifiers, in general, the

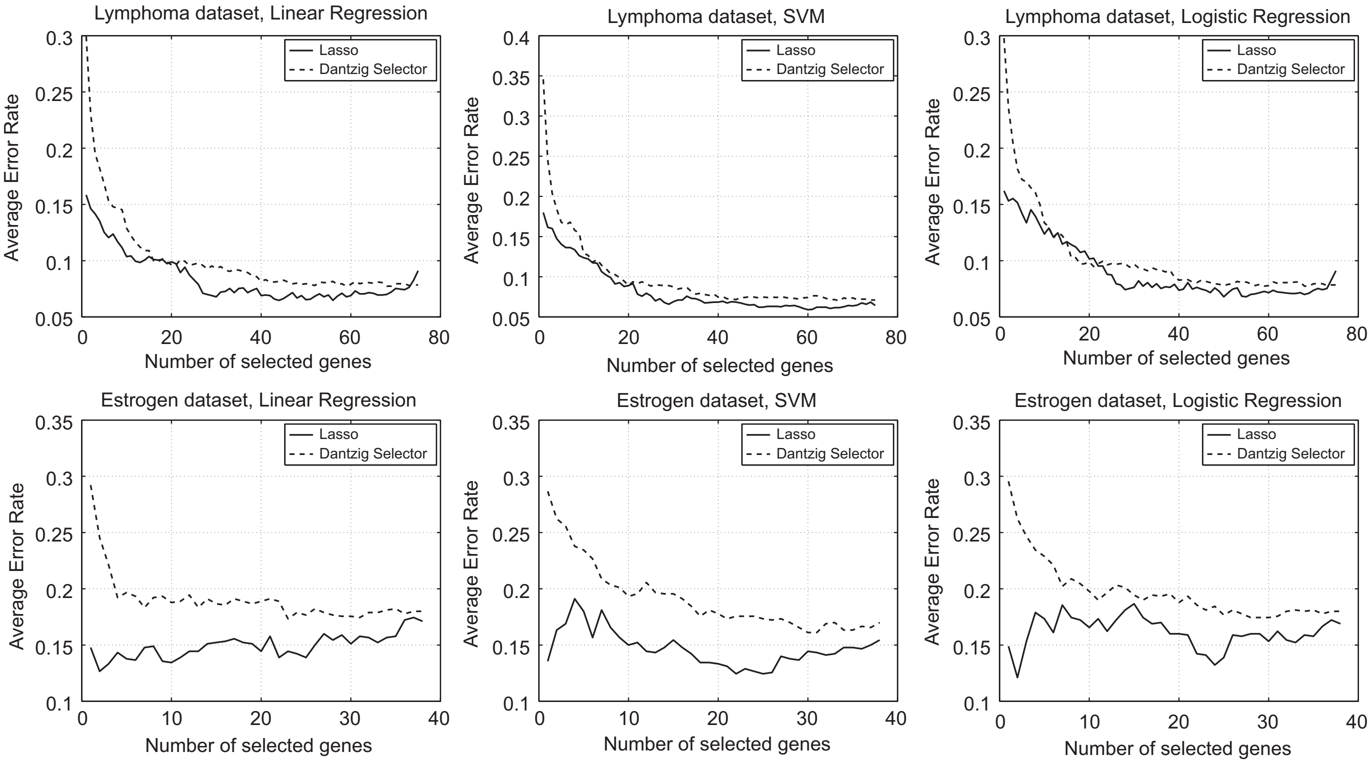


Fig. 2. Continuing Fig. 1: average testing error curves on the Lymphoma and Estrogen with the number of selected genes.

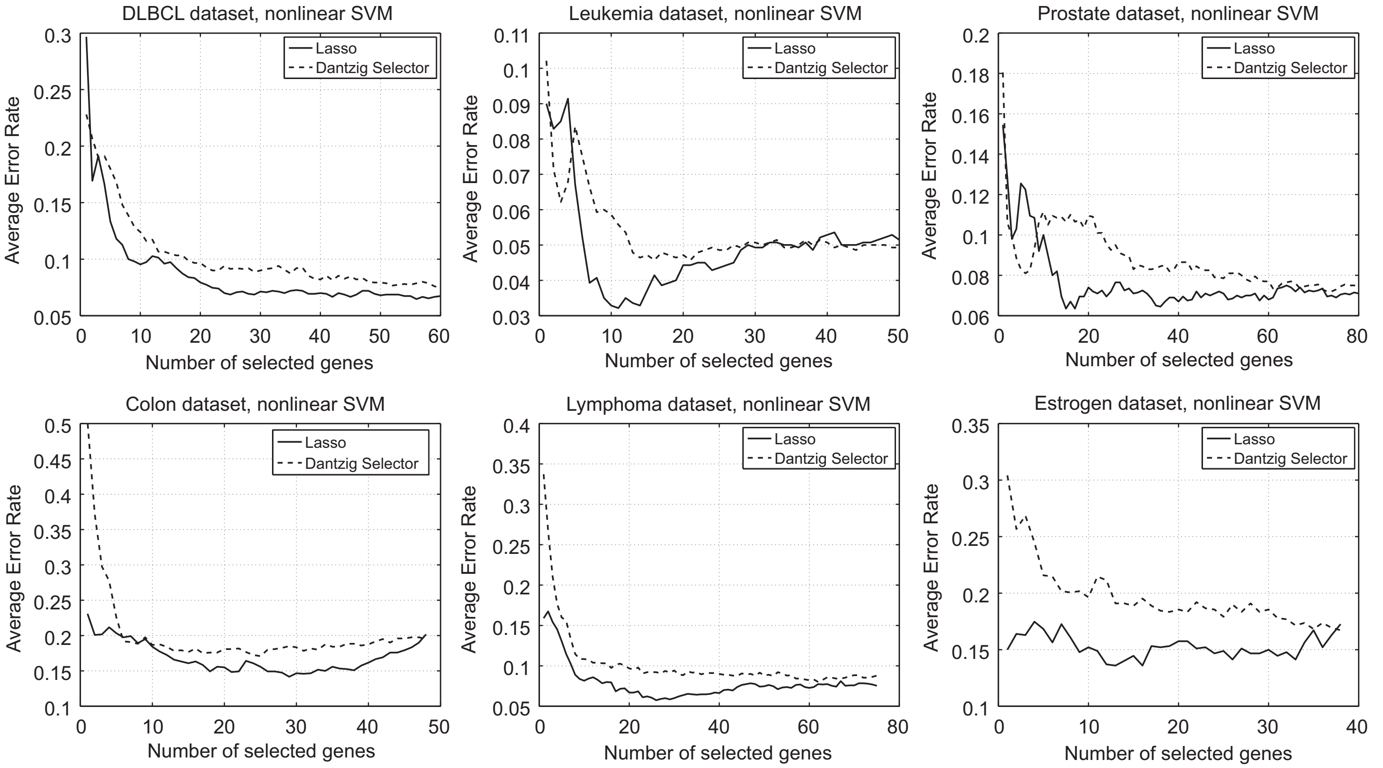


Fig. 3. Average testing error curves of the nonlinear SVM with the number of selected genes on the six datasets.

genes selected by Lasso are more informative than those selected by Dantzig selector for nonlinear SVM.

Comparing the testing error curves of nonlinear SVM in Fig. 3 to those of linear SVM in Figs. 1 and 2, we see that for the same set of selected genes by Lasso or Dantzig selector, on Leukemia dataset, nonlinear SVM has slightly better performance than linear SVM, but on other datasets, the performances of the two types of classifiers are similar.

In summary, from the performance curves on six public datasets with three linear classifiers and nonlinear SVM, we conclude that in general, the genes selected by Lasso are more informative than those selected by Dantzig selector for cancer classification.

4.4. Comparing Lasso and Dantzig selector to elastic net

There are variable selection methods closely related to Lasso, for example, fused Lasso [31], adaptive Lasso [40], and Elastic net [41]. Under the assumption in Eq. (1) and assuming the linear model, these methods could be respectively summarized as

$$\hat{\boldsymbol{\beta}}_{\text{fused}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_{L_2} + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|, \tag{13}$$

$$\hat{\boldsymbol{\beta}}_{\text{ada}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_{L_2} + \lambda \sum_{j=1}^{p} w_j |\beta_j| \tag{14}$$

and

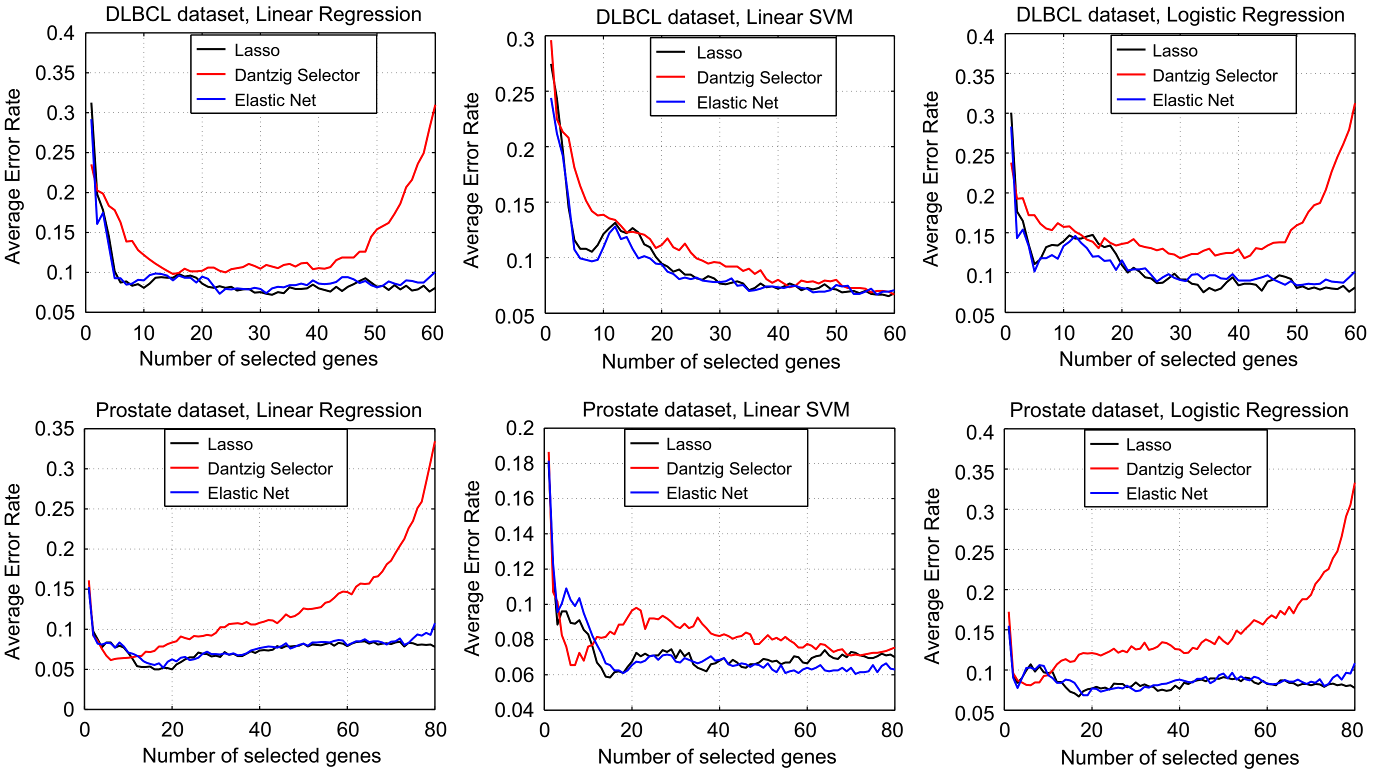
$$\hat{\boldsymbol{\beta}}_{\text{Enet}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|_{L_2} + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2. \tag{15}$$

However, it is discussed in [31] that the computational speed of fused Lasso is a practical difficulty, especially when the parameters in fused Lasso are determined by 5 or 10-fold cross validation. In the adaptive Lasso, the weights w_j are determined from the ordinary least-square (OLS) regression [40], however, the OLS might be unstable in the case of high dimensional data. By using a data augmentation technique, Elastic net can be solved efficiently in the framework of LARS [41]. A variation of the Elastic net was proposed in [19] for genomic data analysis. As such, we experimentally compare the performance of Elastic net in gene selection.

In the Elastic net formulation Eq. (15), there are two tuning parameters, λ_1 and λ_2. For each fixed λ_2, we solve an augmented L_1 constrained regression problem using LARS. In our work, similar to [19], λ_2 is selected from $\{0.001, 0.01, 0.1, 1, 10, 100\}$ by 10-fold cross validation on the training set, and the chosen λ_2 is the one giving the smallest cross validation error. For the chosen λ_2, we repeat the experiments in Sections 4.2 and 4.3. For illustration purpose, we performed the experiments on the DLBCL and Prostate datasets.

Fig. 4 shows the average testing error rate curves of the 100 runs for every linear classifier using the genes selected by Lasso, Dantzig selector, and Elastic net. From Fig. 4, we observe that in most cases, Dantzig selector performs worse than Lasso and Elastic net at picking discriminative genes for linear classifiers, and the Elastic net performs very similar to Lasso. For Gaussian kernel SVM, Fig. 5 shows the average testing error curves of the 100 runs with different number of selected genes. For nonlinear SVM, we observe that, same as for linear classifiers, in general, Lasso and Elastic net have similar gene selection ability, and both are more effective than Dantzig selector.

Although Elastic net can be solved by using the time-efficient LARS algorithm, the optimization problem has a large size due to the use of augmented data, thus Elastic net takes more time than Lasso. Furthermore, Elastic net has two parameters, and we employed cross validation to select λ_2, which consumes extra time. Overall, compared to Lasso, Elastic net is more time expensive although they have the similar gene selection performance.


DLBCL dataset, Linear Regression

DLBCL dataset, Linear SVM

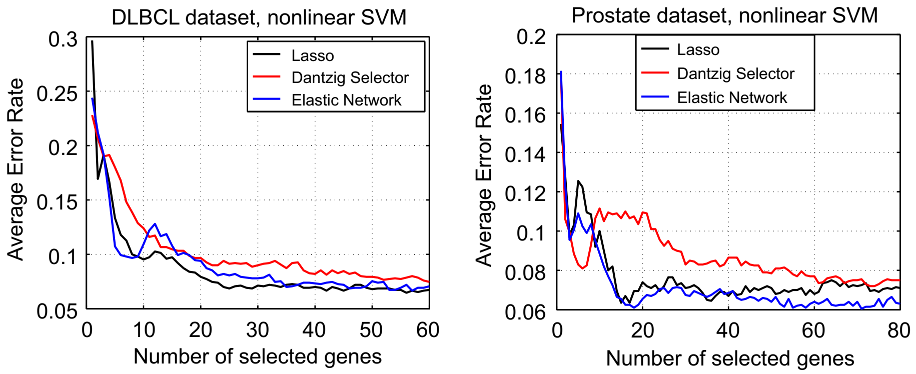
DLBCL dataset, Logistic Regression

Prostate dataset, Linear Regression

Prostate dataset, Linear SVM

Prostate dataset, Logistic Regression

Fig. 4. Average testing error curves with the number of selected genes on the DLBCL and Prostate datasets.


DLBCL dataset, nonlinear SVM

Prostate dataset, nonlinear SVM

Fig. 5. Average testing error curves with the number of selected genes on the DLBCL and Prostate datasets.

5. Conclusion and discussion

In linear regression models, Lasso and Dantzig selector impose upper bound on the L_1-norm of the regression coefficients, such that a part of the regression coefficients are shrunk to zeros, therefore Lasso and Dantzig selector implicitly perform variable selection. This paper applies Lasso and Dantzig selector to regress the class label (which is interpreted as the probability of being positive example) on the gene expression data, thus selecting informative genes. The selected genes are further used by different classifiers for the purpose of cancer classification. On six public cancer datasets, we conducted a comparative study on the gene selection ability of Lasso and Dantzig selector. For three linear classifiers and nonlinear SVM, the testing error rate curves demonstrate that, in general, Lasso is more capable at selecting informative genes than Dantzig selector.

The result presented in this paper is consistent with the conclusion in the statistics literature [11,20]. However, the researches in [11,20] study the variable selection performance of Lasso and Dantzig selector for *linear regression* model, while this paper compares the feature selection ability of Lasso and Dantzig selector for *pattern classification* problems.

The proposed approach has two distinct stages, i.e., gene selection *then* classification. There are strategies which combine feature (gene) selection and classification in a unified framework by adding the L_1 [22,28,25,33,37] or L_∞ constraint [16] directly to the classification models such as logistic regression or support vector machines. By this way, the classification performance might be improved because the selected genes the most suited for the specified classifier (e.g., SVM, logistic regression). This paper chose the two-stage strategy because we intend to compare the gene selection ability of Lasso and Dantzig selector *in general*, not for a particular classifier. Moreover, the two-stage method is easy to understand and implement. Nevertheless, it is instructive to compare classification performances of the two-stage strategy and the unified strategy, and it is part of our future project.

Conflict of interest statement

None.

Acknowledgement

The authors thank the anonymous reviewers for their comments. S. Zheng was supported by a 2011 Summer Faculty Fellowship awarded by Missouri State University. W. Liu was supported by SZU R/D Fund(201054) and the National Natural Science Foundation of China (60903113).

References

[1] A.A. Alizadeh, M.B. Eisen, et al., Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling, Nature 403 (6769) (2000) 503–511.

[2] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, A.J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. USA 96 (1999) 6745–6750.

[3] O. Alter, P.O. Brown, D. Botstein, Singular value decomposition for genome-wide expression data processing and modeling, Proc. Natl. Acad Sci. USA 97 (18) (2000) 10101–10106.

[4] A. Antoniadis, S. Lambert-Lacroix S, F. Leblanc, Effective dimension reduction methods for tumor classification using gene expression data, Bioinformatics 19 (2003) 563–570.

[5] A. Bhattacharjee, W.G. Richards, J. Staunton, et al., Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses, Proc. Natl. Acad. Sci. USA 98 (2001) 13790–13795.

[6] E. Candès, T. Tao, The Dantzig selector: statistical estimation when p is much larger than n, Ann. Statist. 35 (6) (2007) 2313–2351.

[7] M. Dettling, BagBoosting for tumor classification with gene expression data, Bioinformatics 20 (2004) 3583–3593.

[8] M. Dettling, P. Bühlmann, Boosting for tumor classification with gene expression data, Bioinformatics 19 (9) (2003) 1061–1069.

[9] S. Dudoit, J. Fridlyand, T.P. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, J. Am. Stat. Assoc. 97 (457) (2002) 77–87.

[10] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Stat. 32 (2) (2004) 407–499.

[11] B. Efron, T. Hastie, R. Tibshirani, Discussion: the Dantzig selector: statistical estimation when p is much larger than n, Ann. Statist. 35 (6) (2007) 2358–2364.

[12] D. Ghosh, Singular value decomposition regression models for classification of tumors from microarray experiments, Pac. Symp. Biocomput. (2002) 18–29.

[13] T.R. Golub, D.K. Slonim, P. Tamayo, et al., Molecular classification of cancer: class discovery and class prediction by gene expression, Science 286 (1999) 531–537.

[14] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (1–3) (2002) 389–422.

[15] X. Huang, W. Pan, Linear regression and two-class classification with gene expression data, Bioinformatics 19 (16) (2003) 2072–2078.

[16] G. James, P. Radchenko, A generalized Dantzig selector with shrinkage tuning, Biometrika 2 (2009) 323–337.

[17] G. James, P. Radchenko, J. Lv, DASSO: connections between the Dantzig selector and Lasso, J. R. Stat. Soc. B 71 (2009) 127–142.

[18] J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, et al., Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nat. Med. 7 (2001) 673–679.

[19] C. Li, H. Li, Network-constrained regularization and variable selection for analysis of genomic data, Bioinformatics 24 (9) (2008) 1175–1182.

[20] N. Meinshausen, G. Rocha, B. Yu, A Tale of three cousins: LASSO, L_2 Boosting and Dantzig, Ann. Statist. 35 (6) (2007) 2372–2384.

[21] D.V. Nguyen, D.M. Rocke, Tumor classification by partial least squares using microarray gene expression data, Bioinformatics 18 (2002) 39–50.

[22] M.Y. Park, T. Hastie, L_1-regularization path algorithm for generalized linear models, J. R. Stat. Soc. B 69 (2007) 659–677.

[23] C.M. Perou, T. Sorlie, M.B. Eisen MB, M. van de Rijn, S.S. Jeffrey, et al., Molecular portraits of human breast tumors, Nature 406 (2000) 747–752.

[24] A. Potti, S. Mukherjee, R. Petersen, H.K. Dressman, A. Bild, et al., A genomic strategy to refine prognosis in early-stage non-small-cell lung cancer, N. Engl. J. Med. 355 (2006) 570–580.

[25] W. Shi, G. Wahba, S. Wright, K. Lee, R. Klein, B. Klein, LASSO-Patternsearch algorithm with application to ophthalmology and genomic data, Stat. Interface 1 (2008) 137–153.

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

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

[28] S.K. Shevade, S.S. Keerthi, A simple and efficient algorithm for gene selection using sparse logistic regression, Bioinformatics 19 (17) (2003) 2246–2253.

[29] J. Song, Z. Yuan, H. Tan, T. Huber, K. Burrage, Predicting disulfide connectivity from protein sequence using multiple sequence feature vectors and secondary structure, Bioinformatics 23 (2007) 3147–3154.

[30] R. Tibshirani, Regression shrinkage and selection via the Lasso, J. R. Stat. Soc. B 58 (1) (1996) 267–288.

[31] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, K. Knight, Sparsity and smoothness via the fused lasso, J. R. Stat. Soc. B 67 (2005) 91–108.

[32] L.J. van't Veer, H. Dai, M.J. van de Vijver, Y.D. He, A.A. Hart, et al., Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (2002) 530–536.

[33] L. Wang, J. Zhu, H. Zou, Hybrid huberized support vector machines for microarray classification and gene selection, Bioinformatics 24 (3) (2008) 412–419.

[34] J.B. Welsh, P.P. Zarrinkar, L.M. Sapinoso, S.G. Kern, C.A. Behling, et al., Analysis of gene expression profiles in normal and neoplastic ovarian tissue samples identifies candidate molecular markers of epithelial ovarian cancer, Proc. Natl. Acad. Sci. USA 98 (2001) 1176–1181.

[35] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J.R. Marks, J.R. Nevins, Predicting the clinical status of human breast cancer using gene expression profiles, Proc. Natl. Acad. Sci. USA 98 (2001) 11462–11467.

[36] J. Weston, A. Elisseeff, B. Schölkopf, M. Tipping, Use of the zero norm with linear models and kernel methods, J. Mach. Learn. Res. 3 (2003) 1439–1461.

[37] T.T. Wu, Y.F. Chen, T. Hastie, E. Sobel, K. Lange, Genome-wide association analysis by Lasso penalized logistic regression, Bioinformatics 25 (2009) 714–721.

[38] M. Xiong, X. Fang, J. Zhao, Biomarker identification by feature wrappers, Genome Res. 11 (11) (2001) 1878–1887.

[39] M. Xiong, L. Jin, W. Li, E. Boerwinkle, Computational methods for gene expression-based tumor classification, Biotechniques 29 (6) (2000) 1264–1268.

[40] H. Zou, The adaptive Lasso and its oracle properties, J. Am. Stat. Assoc. 101 (476) (2006) 1418–1429.

[41] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. B 67 (2005) 301–320.