Spring 5-18-2018

Variable selection via Lasso with high-dimensional proteomic data

Hongxuan Zhai
Washington University in St. Louis

WASHINGTON UNIVERSITY IN ST. LOUIS

Department of Mathematics

Variable Selection via Lasso with High-dimensional Proteomic Data

by

Hongxuan Zhai

A thesis presented to
The Graduate School
of Washington University in
partial fulfillment of the
requirements for the degree
of Master of Art

May 2018
St. Louis, Missouri

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like first to thank to my families. They supported me to finish my degree and they encouraged me while I studying abroad. I also want to thank my adviser, professor Kuffner, and he provided me with detailed guidance and resources. Thanks to all the professors who have taught me. From their teaching, I obtained knowledge and knew how to solve problems in scientific ways.

Hongxuan Zhai

Washington University in St. Louis

May 2018

ABSTRACT

Variable Selection via Lasso with High-dimensional Proteomic Data

by

Hongxuan Zhai

A.M. in Statistics,

Washington University in St. Louis, 2018.

Professor Todd Kuffner, Chair

Multiclass classification with high-dimensional data is an applied topic both in statistics and machine learning. The classification procedure could be done in various ways. In this thesis, we review the theory of the Lasso procedure which provides a parameter estimator while simultaneously achieving dimension reduction due to a property of the ℓ_1 norm. Lasso with elastic net penalty and sparse group lasso are also reviewed. Our data is high-dimensional proteomic data (iTRAQ ratios) of breast cancer patients with four subtypes of breast cancer. We use the multinomial logistic regression to train our classifier and use the false classification rates obtained from cross validation to compare models.

1. Introduction

The multinomial logistic model is frequently used in analysis of nominal multi-category response variables. The model can be regarded as a generalized linear model (GLM) with a logit link function. To estimate parameters in the multinomial logistic model, the maximum likelihood estimator (MLE) is typically used. However, MLE has the limitation that it will not provide a robust result when there are more parameters to be estimated than observations, or when some of the predictors are highly correlated. In both cases, MLEs' tend to deteriorate rapidly. This has a negative effect on model interpretability. As a result, a variable selection procedure or dimension reduction procedure is needed to obtain a more robust estimation result when the number of predictors,p, is much greater than that of observations,n.

The Lasso, proposed by [1], is an acronym for Least Absolute Shrinkage and Selection Operator, and it has become one of most popular methods for dealing with high-dimensional estimation problems. The data for our application study contains isobaric tags for relative and absolute quantitation (iTRAQ) proteome profiling of 77 breast cancer samples and 3 duplicate breast cancer samples generated by the Clinical Proteomic Tumor Analysis Consortium (NCI/NIH). The iTRAQ reporter ion intensity ratio is used to determine relative abundance of proteins within each sample. Data set contains expression values for 12546 proteins for each sample, with missing values present when a given protein could not be quantified in a given sample. Relating to the proteomic data, the response variable is sub-type of breast cancer given a certain sample. The proteomic data analyses is a "ratio-based" procedure, and the type of predictor variables are continuous while the type of response variable is categorical.

The data set is generated by an isobaric labeling method used in quantitative proteomics by Mass Spectrometry (MS) method, because proteomic analyses are performed on tumor fragments that are different from those used in genomic analysis, a pre-specified sample QC metrics are implemented. All the samples that do not exhibit a unimodal normal distribution are excluded from the study. The original experiment selected samples for proteomic analyses from the subset annotated as having at least 130 mg wet weight residual material, the target amount for proteomics processing between collaborating research teams. After using this criterion, 131 sub-type samples were requisitioned from TCGA, including 28 basal, 20 HER2-enriched, 39 luminal A, and 39 Luminal B. 126 samples were obtained, of which 105 yielded at least the pre-specified minimum of 0.7 mg of total protein after extraction of proteins with 8M urea buffner. Among the 105 samples, there are 28 of tumor samples exhibiting highly skewed protein distribution. Finally, researchers obtain 77 tumor samples as well as 3 replicates that exhibited the expected gaussian unimodal distribution of a log iTRAQ ratio. It was assumed that for proteomic analyses tumor samples should be normalized samples with a log iTRAQ ratio centered at zero. As a result, a normalization scheme was employed that attempted to identify the unregulated proteins and centered the distribution of these log-ratios around zero in order to nullify the effect of systematic MS variation. There exist missing values in 77 samples, so an imputation is needed. In order to diminish the effect of outliers, we use the sample median to impute the missing values. Table (1.1) is the table describing some features of the data and Figure (1.1) is the overlay density plot for 10 of the 80 samples to visualize the centered unimodal density.

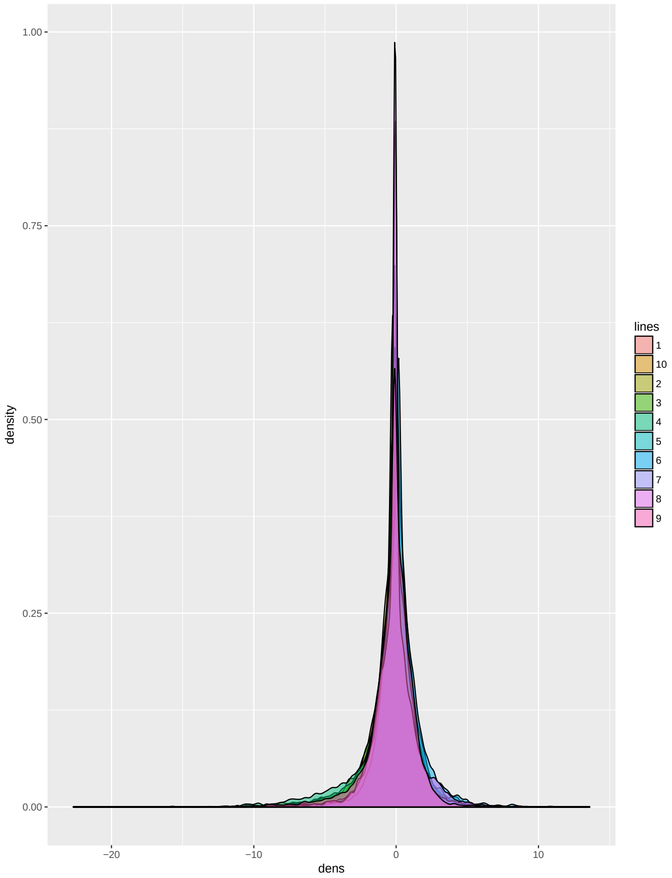


Figure 1.1. Overlay density plot for 10 samples.

There are various well-established methods for variable selection. In the classical linear regression setup, forward/backward strategies have been utilized [2]. However, these methods are unstable and computationally costly [3]. For high-dimensional data, where $p \gg n$, the ordinary least squares estimator (OLSE) is not unique and will heavily overfit the data. Thus, regularized estimation of the regression coefficients is necessary. When we focus on the regulation with ℓ_1 penalty, the parameters in linear regression are estimated with Lasso. By introducing this ℓ_1 regularization method, the Lasso for linear

Table 1.1
Some features of the data.

Variable	**predictor variables X**	**response variables Y**
number of observations	12546	77
variable type	log-based continuous numeric ratio	multi-categorical

models will be a convex optimization problem, which at the same time achieves variable selection, because of the ℓ_1-geometry [1]. An alternative regularized regression method is elastic net method that employs a combination of an ℓ_1 penalty and ℓ_2 penalty, which is also a convex optimization [4]. Since both methods perform variable selection, we want to compare the performance of both methods for our proteomic data. The penalty parameter will greatly affect the model complexity. A large penalty parameter means that you will have zero or few variables in the model (a very sparse solution), while a small penalty parameter gets you closer to the least-squares solution (with as many predictors as can be estimated from n observations). We also need to do grid searches for the tuning parameters for the two selected methods under certain criterion. Therefore, we search for optimal values of the tuning parameters over a grid of candidate values, where optimality is defined in terms of mean square prediction error (MSPE). We will use the publicly available `R` packages `glmnet` [5] and `caret` [6].

2. Statistical Models and Regulation Methods

2.1 The Multinomial Logistic Model

The multinomial logistic regression model is a particular type of GLM [7] which allows for multi-category response variables, i.e. there are more than two categories for the response variable. Similar to the logistic model with binary response, the multinomial logistic model utilizes the logit link function to model the logarithm of the odds ratio as the linear combination of predictor variables $X = (X_1, X_2, ...X_p)$. Suppose p is a random variable taking values between 0 and 1; The logit link function is defined as

$$logit(p) = \log(\frac{p}{1-p});$$

Since the link function could be regarded as a transformation of the conditional mean $\mathrm{E}(Y|X = x)$, it would naturally give rise to the context of regression. The linear logistic regression model with multicategory response can be regarded as a generalization of the binary response linear logistic regression that extends one logit to multiple logits. Suppose Y is a multicategory response with K levels; A multinomial logistic model with predictors $X = (X_1, X_2, ...X_p)$ is defined as

$$\log \frac{Pr(Y = \ell|x)}{Pr(Y = K|x)} = \beta_{0\ell} + x^T \beta_\ell, \quad \ell = 1, 2, ..., K-1,$$

where $\beta_\ell^T = (\beta_{l1}, ..., \beta_{lp})$ are the regression coefficients.

An equivalent but more symmetric parametrization [4]

$$Pr(Y = \ell|x) = \frac{\exp(\beta_{0\ell} + x^T \beta_\ell)}{\sum_{k=1}^{K} \exp(\beta_{0k} + x^T \beta_k)}$$

Notice that this equivalent parametrization is not estimable since for parameters $(\beta_{0\ell}, \beta_\ell)$, a shifted version $(\beta_{0\ell} - a_0, \beta_\ell - a)$ will generate the same probability measure. The non-estimable property will also make the log-likelihood insensitive to the shifting constant (a_0, a) when we do maximum likelihood estimation [8].

2.2 The Multinomial Logistic Model with Lasso and Elastic-Net Regulation

In the classical linear model context, given a collection of N samples $(x_i, y_i)_{i=1}^N$, the Lasso method finds the solution $(\hat{\beta_0}, \hat{\beta})$ of the optimization problem defined by

$$\underset{\beta_0,\beta}{\text{minimize}} \frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

$$\text{subject to} \sum_{j=1}^{p} |\beta_j| \leq t;$$

Notice that the constraint could also be written as $||\beta||_1 \leq t$, where $\| \cdot \|$ denotes the ℓ_1 norm. Due to Lagrangian duality, the minimization procedure could also be in a Lagrangian form, defined by

$$\underset{\beta_0,\beta}{\text{minimize}} \frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda ||\beta||_1,$$

for some $\lambda \geq 0$. There exists a one-to-one mapping between the solutions for the two constrained problems. In a multinomial logistic regression setup, the negative log-likelihood with ℓ_1 penalization is given by

$$-\frac{1}{N} \sum_{i=1}^{N} \log(Pr(Y = y_i | x_i; (\beta_0, \beta_k)_{k=1}^K) + \lambda \sum_{k=1}^{K} ||\beta_k||_1,$$

where β_k is a vector with components $\beta_{1k}, \ldots, \beta_{pk}$, and that the different β_k correspond to the vectors of coefficients for the K different classes for the response. Since the multinomial probability measure and the corresponding log-likelihood function are invariant with respect to a constant shift of K coefficients, the penalty function is not invariant with respect to a constant shift coefficients [9]. Penalty term could resolve the choice of c_j when $\{\beta_{kj} + c_j\}$ and $\{\beta_{kj}\}$ generate same probability from the likelihood function. As a result, the optimal choice c_j for each candidate set $\{\beta_{kj}\}_{k=1}^{K}$ could be generated by

$$\underset{c \in \mathbb{R}}{\operatorname{argmin}} \sum_{k=1}^{K} |\beta_{kj} - c|$$

and it can be shown that the solution of the objective is the median of $\{\hat{\beta_{1j}}, ..., \hat{\beta_{Kj}}\}$ for each $j = 1, ..., p$.

Estimation of the multinomial logistic model via elastic net corresponds to a penalized estimation problem of the form

$$\underset{\beta_0, \beta}{\text{minimize}} -\frac{1}{N} \log(Pr(Y = y_i | x_i; (\beta_0, \beta_k)_{k=1}^{K}) + \lambda \sum_{j=1}^{p} \rho_j (1-\alpha)\beta_j^2 + \rho_j \alpha |\beta_j|.$$

This penalty could be regarded as a compromise between ℓ_1 and ℓ_2 penalty (also known as ridge penalty) and this penalty is particularly useful in the $p \gg N$ situation [4]. The parameter α is a real number between 0 and 1 serving as the weighting parameter of the Lasso penalty and ρ_j is non-negative quantity serving as a penalty modifier. Notice that elastic net penalty is convex, and hence we can employ convex optimization methods.

2.3 The Uniqueness of Lasso Solutions and Optimization via Coordinate Descent Algorithm

The Lasso estimator is the solution of the optimization problem

$$\underset{\beta_0, \beta}{\text{minimize}} \frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2$$

$$\text{subject to} \sum_{j=1}^{p} |\beta_j| \leq t;$$

It is known that the solution is unique when the rank of the X matrix is equal to the number of columns. Notice that in high-dimensional data analysis, we have data sets where the number of variables exceeds the number of observations. As a result, the Lasso criterion is not strictly convex and there are infinitely many solutions, $\hat{\beta}$, that yield a perfect fit with zero training error. This leads to instability in the estimates, even though the fitted values of $X\hat{\beta}$ are unique. For illustration, consider one simple example where x_1 and x_2 are predictor variables and y is the response variable, and suppose the Lasso solution $\hat{\beta}$ at a certain λ is $(\hat{\beta}_1, \hat{\beta}_2)$. If there is an additional predictor $x_3 = x_2$ in the model, any vector satisfying $\hat{\beta}(\alpha) = (\hat{\beta}_1, \alpha\hat{\beta}_2, (1-\alpha)\hat{\beta}_2)$ for $\alpha \in [0,1]$ produces the same model fit and has the same ℓ_1 norm. Obviously, in this example, there are infinitely many solutions.

The columns of the matrix $\mathbf{X}$ are said to be in general position if for $\{x_j\}_{j=1}^{p}$, any affine subspace $\mathbb{L}$ in $\mathbb{R}^N$ of dimension $k < N$ contains at most $k+1$ elements of the set $\{\pm x_1, \pm x_2, ..., \pm x_3\}$, excluding antipodal pairs of points. An affine space is a geometric structure that generalizes the properties of Euclidean spaces in such a way that these are independent of the concepts of distance and measure of angles, keeping only the properties related to parallelism and ratio of lengths for parallel line segments. In a high-dimensional data setting, one can show that if the predictor variables are drawn from a continuous probability distribution, then with probability one, the columns of $\mathbf{X}$ are in general position form and there exists a unique Lasso solution [10].

For a general differentiable convex function f with convex constraint set $\mathbb{C} \in \mathbb{R}^p$, consider the constrained optimization problem defined by

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}}\, f(\beta) \text{ such that } \beta \in \mathbb{C};$$

A necessary and sufficient condition for a vector $\beta^* \in \mathbb{C}$ to be a global optimum is that

$$< \bigtriangledown f(\beta^*), \beta - \beta^* > \geq \; 0;$$

When the constraint set $\mathbb{C}$ can be described as sublevel sets of certain convex constraint functions $g \; : \; \mathbb{R}^p \to \mathbb{R}$, the convex optimization problem can be written in the form of

$$\underset{\beta \in \mathbb{R}^p}{\text{minimize}}\, f(\beta) \; such \; that \; g_j(\beta) \; \geq \; 0 \; for \; j = 1, ..., m;$$

The Lagrangian associated with this problem is

$$L(\beta, \lambda) = f(\beta) + \sum_{j=1}^{m} \lambda_j g_j(\beta),$$

where the weights $\lambda \geq 0$ are Lagrange multipliers. Under technical conditions on f and g_i, Lagrangian duality guarantees the existence of an optimal choice of λ^*. The necessary and sufficient conditions for finding the global optimum β^* related to λ^* are the Karush-Kuhn-Tucker (KKT) conditions.

The Lasso problem involves the ℓ_1 norm, and hence the objective function fails to be differentiable when any of the coordinates β_j is exactly equal to zero. In this situation, the KKT conditions are not directly applicable but there is a generalized notion of the gradient called the subgradient. Based on the property that for a differentiable convex function the first-order tangent approximation always provides a lower bound, the notion of a subgradient of function f at β is defined as a vector $z \in \mathbb{R}^p$ such that

$$f(\beta') \geq f(\beta) + < z, \beta' - \beta > \text{ for all } \beta' \in \mathbb{R}^p;$$

The set of all subgradients of f at β is called the sub-differential, denoted by $\partial f(\beta)$. For absolute value function $f(\beta) = |\beta|$, we define

$$\partial f(\beta) = \begin{cases} 1, & \text{if } \beta \text{ is greater than 0} \\ -1, & \text{if } \beta \text{ is less than 0} \\ [-1, 1], & \text{if } \beta \text{ is 0} \end{cases}$$

We will use the notation that $z \in \text{sign}(\beta)$ to express the idea that z belongs to the sub-differential of the absolute value function at β.As a result, the first-order condition, i.e. the requirement that the gradient is zero for an optimal solution, can be generalized to a condition involving the sub-differential,

$$0 \in \partial f(\beta^*) + \sum_{j=1}^{m} \lambda_j^* \partial g_j(\beta^*);$$

Applying this to the Lasso problem, we have the constraint function g specified as $g(\beta) = \sum_{j=1}^{p} |\beta_j| - R$ for some positive R. Numerical methods are needed to solve such an optimization problem. Newton's method as a second-order method that using knowledge of Hessian, not just first derivative, has a quadratic rate of convergence; However Newton's method has lower computation efficiency. Especially in the multinomial regression setup, newton's method can be tedious. Coordinate descent method is an iterative algorithm that updates parameter β by choosing a single coordinate, and then doing a univariate minimization over the chosen coordinate using first-order method [9]. To be more specific, suppose coordinate k is chosen at iteration t; The update for the chosen coordinate is given by

$$\beta_k^{t+1} = \underset{\beta_k}{\text{argmin}}\, f(\beta_1^t, \beta_2^t, \beta_3^t, ..., \beta_k, \beta_{k+1}^t, ..., \beta_p^t),$$

and $\beta_j^{t+1} = \beta_j^t$ for $j \neq k$. This algorithm solves the optimization problem by cycling through the coordinates in a particular fixed order. One sufficient condition for convergence to the global minimum is that function f is continuously differentiable and strictly convex with respect to each coordinate. It is obvious that when using the ℓ_1 norm penalty, this restrictive condition is not satisfied. In such cases, a separability condition will ensure that coordinate-wise minimization algorithms do not get stuck at sub-optimal values. The separability condition for a cost function f is defined as an additive decomposition

$$f(\beta_1, ..., \beta_p) = g(\beta_1, ..., \beta_p) + \sum_{j=1}^{p} h_j(\beta_j),$$

where function $g : \mathbb{R}^p \rightarrow \mathbb{R}$ is differentiable and convex, while any of the univariate functions $h_j : \mathbb{R} \rightarrow \mathbb{R}$ are convex. [11] shows that for any convex cost function f with separable structure, the coordinate descent algorithm is guaranteed to converge to the global minimizer.

The coordinate descent method is implemented in regularized multinomial regression to get the estimate of coefficients. Followed by [8], the optimization procedure utilizes both partial Newton steps by forming a partial quadratic approximation and the coordinate descent method. As a generalization for regularized logistic regression of binary response, we model the multinomial case by

$$Pr(Y = \ell | x) = \frac{\exp(\beta_{0\ell} + x^T \beta_\ell)}{\sum_{k=1}^{K} \exp(\beta_{0k} + x^T \beta_k)}$$

suggested by [4]. Here the corresponding likelihood function becomes regularized maximum multinomial likelihood. We suppose the categorical response variables to be G, which has $K > 2$ levels, and $g_i \in \{1, 2, ..., K\}$ be the ith response. By introducing Y to be the indicator response matrix with $y_{i\ell} = I(g_i = \ell)$, the explicit form of unregularized log-likelihood is given by

$$\ell(\{\beta_{0\ell}, \beta_\ell\}_1^K) = \frac{1}{N} \sum_{i=1}^{N} [\sum_{\ell=1}^{K} y_{i\ell}(\beta_{0\ell} + {x_i}^T \beta_\ell - \log(\sum_{\ell=1}^{K} e^{\beta_{0\ell} + {x_i}^T \beta_\ell})].$$

First, we utilize partial Newton steps by performing a partial quadratic approximation to the unregularized log-likelihood and get

$$\ell_{Q\ell}(\beta_{0\ell}, \beta\ell) = -\frac{1}{2N} \sum_{i=1}^{N} w_{i\ell}(z_{i\ell} - \beta_{0\ell} - {x_i}^T \beta_\ell)^2 + C(\{\beta_{0k}, \beta_k\}_1^K)$$

where

$$z_{i\ell} = \tilde{\beta_{0\ell}} + {x_i}^T \tilde{\beta_\ell} + \frac{y_{i\ell} - \tilde{p_\ell}(x_i)}{\tilde{p_\ell}(x_i)(1 - \tilde{p_\ell}(x_i))}$$

$$w_{i\ell} = \tilde{p_\ell}(x_i)(1 - \tilde{p_\ell}(x_i))$$

Second, we utilize the current parameter estimates$(\tilde{\beta_0}, \tilde{\beta})$ and coordinate descent algorithm to solve the problem with penalization, which allows only one element in $(\beta_{0\ell}, \beta_\ell)$ to vary at a time. The problem is described as

$$\underset{\beta_{0\ell}, \beta_\ell}{\text{minimize}} \{-\ell_{Q\ell}(\beta_{0\ell}, \beta_\ell) + \lambda P_\alpha(\beta_\ell)\}.$$

2.4 Sparse Group Lasso

The sparse group lasso method is a combination between the lasso [1] and the group lasso [12]. The sparse group lasso also utilizes a gradient descent method to find the solution to the optimization problem. In the multiclass classification setting, the sparse group lasso method based on a multinomial regression model takes the structured feature of parameters into consideration and generally improves the performance of the classifier in the high-dimensional setting [13]. Compared to the Lasso or group Lasso, the sparse group Lasso also substantially reduces the number of selected variables.

Consider we have p features and we decompose the search space in to m blocks

$$R^p = R^{p_1} \times ... \times R^{p_m}$$

where p_i is the dimension of the group i, with $p = p_1 + p_2 + .. + p_m$. For coefficient vector β we have $\beta = (\beta^{(1)}, ..., \beta^{(m)})$ where $\beta^{(1)} \in R^{p_1}, ..., \beta^{(m)} \in R^{p_m}$. The subvector $\beta^{(J)}$ is the Jth block of β for $J = 1, ..., m$, and we denote β_i^J as the ith coordinate of the Jth block of coefficients.

The sparse group lasso penalty is defined as

$$\Phi(\beta) = (1 - \alpha) \sum_{J=1}^{m} \gamma_J ||\beta^{(J)}||_2 + \alpha \sum_{i=1}^{p} \xi_i |\beta_i|,$$

where $\alpha \in [0, 1]$, $\gamma \in [0, \infty)^m$ are the group weights, and parameter weights $\xi = (\xi^{(1)}, ..., \xi^{(m)}) \in [0, \infty)^p$ for $\xi^{(1)} \in [0, \infty)^{p_1}, ..., \xi^{(m)} \in [0, \infty)^{p_m}$. As with the elastic net method, the tuning parameter α could lead to two different methods by taking $\alpha = 1$ (lasso penalty) or $\alpha = 0$ (group lasso penalty).

The multinomial sparse group lasso classifier problem with K classes, N samples and p features has the $N \times p$ design matrix $X = (x_1, ..., x_N)^T$ and $y_i \in \{1, ..., K\}$ is the categorical response. A symmetric parametrization is used in sparse group lasso with loss function $h(l, \eta) = \dfrac{\exp(\eta_1)}{\sum_{k=1}^{K} \exp(\eta_k)}$; Together with this penalty function, the sparse group lasso problem is expressed as a penalized likelihood criterion,

$$-\sum_{i=1}^{N} \log(h(y_i, \beta^{(0)} + \beta x_i)) + \lambda((1 - \alpha) \sum_{J=1}^{p} \gamma_J ||\beta^{(J)}||_2 + \alpha \sum_{i=1}^{Kp} \xi_i |\beta_i|);$$

For computational aspect, the `R` package `msgl` citepmsgl uses a decreasing sequence of λ and three nested main loops to solve the optimization problem.

2.5 Random Forest Classifier

Decision trees are a non-parametric supervised learning algorithm used for regression and classification. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. Given an input which is usually represented by a feature vector, decision trees make prediction according to function in hypothesis space. Decision trees' performance can be evaluated based on mean square error and mean square prediction error. Given a data, one could use bootstrap scheme to establish multiple decision trees, which is called a random forest. The random forest classifier is defined by $\{h(x, \Theta_k), k = 1, ...\}$ where Θ_k are independent identically distributed random vectors and each tree casts a unit vote for the most "popular" class at each input x [14]. Random forests are composite methods that consist of decision trees generated by bootstrapping training data. The variables at each node are random subsets of the full set of predictor variables, and also each node and the number of nodes in the decision tree are generated randomly. All the decision trees can be grown to the desired size. A large number of trees are generated based on B bootstrap samples from n samples and vote for the most popular class, and such procedure is called random forest.

2.6 Support Vector Machine Classifier

Support vector machines (SVMs) are supervised classification methods which utilize separating hyperplanes. Given a training dataset with pre-specified labels, SVMs give classification outputs based on an optimal hyperplane computed by categorizing new data points. Classical SVMs achieve separation between two classes by making use of a hyperplane $w^T x + b = 0$ with maximum value of $\rho = 2/||w||$. Finding the optimal separating hyperplane is equivalent to solving the following optimization problem in Lagrangian form:

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i [y_i(w^T x_i + b) - 1];$$

Based on the optimum w^*, b^*, α^*, we can obtain the optimal hyperplane, which is defined by ${w^*}^T x + b^* = 0$. With each input vector x, outputs are calculated by the function $\text{sign}({w^*}^T x + b^*)$; Multiclass SVMs are a multiclass generalization of SVMs for binary classification. Multiclass SVM builds and combines several binary class SVMs classifiers, and hence is more computational expensive than binary class SVMs. This method is called the one-against-one method. The procedure builds $k(k-1)/2$ classifiers, with each classifier trained on data obtained from any two classes. The classification problem is defined by minimizing over w, b, ξ

$$\frac{1}{2}(w^{ij})^T w^{ij} + C \sum_t \xi_t^{ij},$$

subject to

$$(w^{ij})^T \phi(x_t) + b^{ij} \geq 1 - \xi_t^{ij}, \quad \text{if } y_t = i$$

$$(w^{ij})^T \phi(x_t) + b^{ij} \leq -1 + \xi_t^{ij}, \quad \text{if } y_t = j$$

$$\xi_t^{ij} \geq 0,$$

where $C \sum_t \xi_t^{ij}$ is the penalty term, and training data is from ith and jth classes. The function ϕ maps training data into a higher-dimensional space.

3. Data Analysis and Model Selection

3.1 Check for Model Assumption

One main feature of Lasso regression is that Lasso regression will do variable selection and model fitting at the same time; However, Lasso regression is also notoriously known as its instability. As a result, we need to further investigate the credibility of our selection algorithm and how stable are our findings so that we could have a more efficient and reasonable statistical procedure. First, we describe the variable selection procedure for Lasso to be

$$\hat{S}(\lambda) = \{j; \hat{\beta}_j(\lambda) \neq 0\}$$

$$for \quad S_0 = \{j; \beta_j^0 \neq 0\},$$

where $\hat{S}(\lambda)$ could be regarded as the selected non-zero coefficient set and S_0 could be regarded as the true non-zero coefficient set. If a variable selection procedure performs well, we expect that there is high probability that the two sets are essentially equal. In order to get a stable solution in Lasso regression, some conditions and problems are worth considering.

- Neighborhood stability condition is restrictive
- Choice of λ will affect the selection of variables
- Small non-zero regression coefficients are hard to detect

Among these conditions and problems, the neighborhood stability condition is restrictive, and it is also a sufficient and necessary condition for consistent model selection with Lasso. For a matrix X, the neighborhood stability condition is that ,in terms of submatrices, there is no strong linear dependence. Since in Lasso regression, the regression coefficients are function of the chosen λ, Empirically, λ is chosen to be λ_{opt} defined as

$$\lambda_{opt} = \underset{\lambda}{\operatorname{argmin}} \mathbb{E}(Y - \sum_{j=1}^{p} \hat{\beta}_j(\lambda) X^{(j)})^2;$$

It can be shown that for prediction optimal λ_{opt}

$$\hat{S}(\lambda opt) \supseteq S_0,$$

which means that the true active set is contained in the estimated active set including the selected variables. This is the screening feature of Lasso regression. Considering our data, the design matrix X has some correlated columns but not strongly correlated since the number of correlated columns is much smaller than the total number of columns. The motivating data satisfy the neighborhood stability condition.

3.2 Data Analysis Using Regularized Multinomial Logistic Regression with Lasso

We label the four breast cancer sub-types (Luminal A, Luminal B, Basal-like, and HER2-enriched) with categorical variables 1 to 4 and treat them as the response variable with respect to each sample. We first choose the tuning parameter λ in the Lasso penalty using cross-validation procedure and choose λ_{opt} to be the one that minimize the deviance with respect to multinomial logistic regression. The cross-validation plot is generated using `R` package `glmnet`. Figure (3.1) illustrated the optimal choice considering the prediction oracle. We could also see the obvious shrinkage effect of Lasso regression

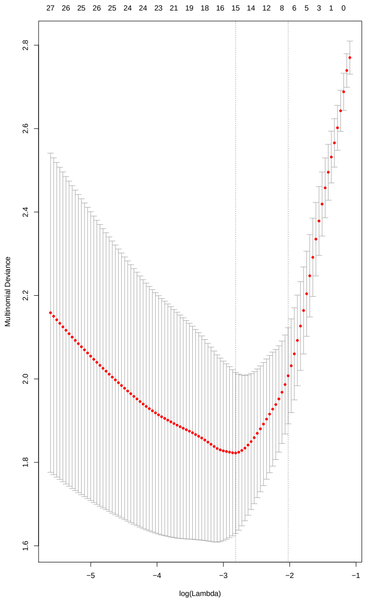


Figure 3.1. Cross-validation plots for λ_{opt}.

with increment of λ in figure (3.2). Larger value of λ will eliminate more variables in the model. Utilizing the λ_{opt} via cross-validation to fit the multinomial logistic model by coordinate descent algorithm, we will do both variable selection and model fitting in one statistical procedure. Notice that `glmnet` package uses the "redundant" parameterization of multinomial logistic regression. As a result, we will get 4 groups of selected variables and their corresponding coefficients. It is obvious that under λ_{opt}, the dimension of the regression problem is greatly reduced. From each category, Lasso regression select 18, 21, 13, 11 variables respectively. As suggested by [8], the intercept coefficients are always not penalized. So that in each model for a certain category, there is always an intercept term. Figure (3.3) displays the coefficient plot for Lasso.

It is worth mentioning that Lasso method is insensitive to the highly correlated data for doing the variable selection, however, the ℓ_2 norm penalty can distinguish the selected

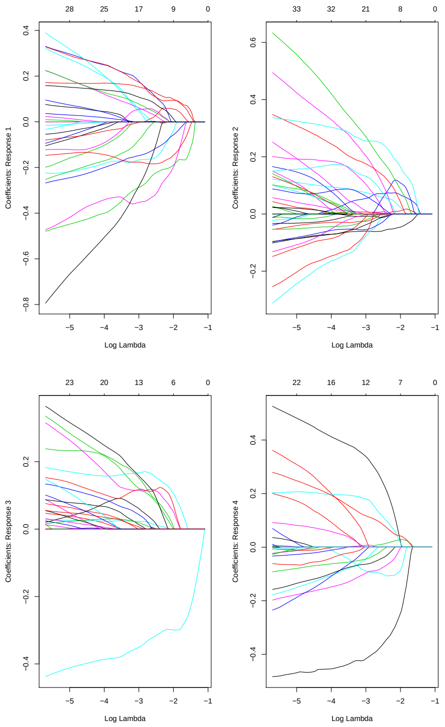


Figure 3.2. Shrinkage effect of Lasso and coefficient path.

variables from correlated variables, which leads us to do a multinomial regression procedure using a different penalty function-elastic net method. In order to get the best tuning parameters in elasticnet model, we need to do a grid search over both the elastic net mixing parameter α and the penalized parameter λ. Since the elastic net mixing parameter $\alpha \in [0,1]$, we only need to have an reasonable interval from which we can have an efficient search over penalized parameter λ. A theorem related to the variable selection with Lasso suggests that under some conditions, some choices of λ will lead to a good variable selection procedure. These conditions are described as:

- the design matrix satisfy the neighborhood stability condition [15]

- data has high-dimensionality

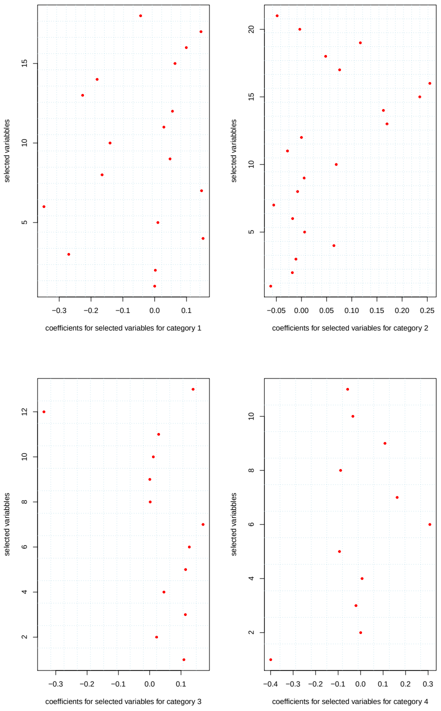


Figure 3.3. Coefficient Plot via Lasso.

- the estimated active set has sparsity

Then if

$$\lambda = \lambda_n \sim n^{-\frac{1}{2}-\frac{\delta}{2}}(0 < \delta < \frac{1}{2})$$

$$\mathbb{P}[\hat{S}(\lambda) = S_0]1 \ \mathit{even\ for\ relatively\ small}\ n;$$

This gives us a tentative guide for the search interval. Figure (3.4) shows the truncated graph of the grid search with respect to penalized parameter λ under different elasticnet mixing parameter α. After the grid search using the repeated cross-validation criterion, we get the tuning parameters to be $\alpha = 0.24242$ and $\lambda = 0.08636$ respectively. The regression with elasticnet selects 128, 140, 106, 93 variables (include the unpenalized intercept term) respectively for each category. Figure (3.5) illustrate the relative shrinkage effect of elasticnet, from which we conclude that elasticnet method tend to select more variables into the model. Figure (3.6) displays the coefficient plot for elasticnet.

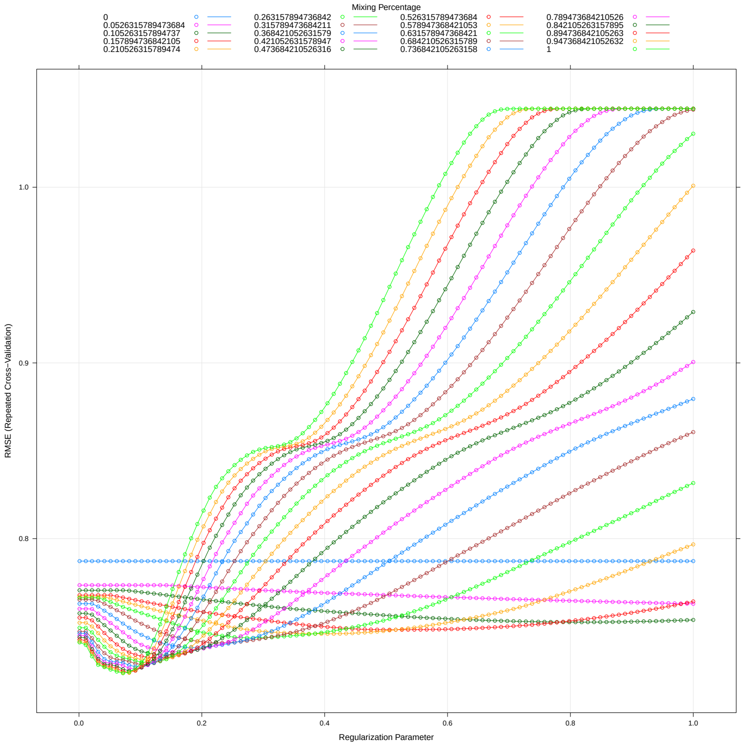


Figure 3.4. Truncated version of grid search plot.

3.3 Data Analysis using Sparse Group Lasso

R package msgl allows users to fit a multinomial logistic regression with sparse group lasso penalty with a sequence of tuning parameters λ. With this setting, users need to prespecify a minimum value of sequence of λ. In this thesis, we set the minimum of sequence of λ to be 0.05, hence msgl fit the model with whole data with a λ sequence ranging from 0.22 to 0.05. The fitted models description is shown in the table. Throughout the table, one can clearly see the sparsity of coefficients of sparse group lasso method based on the number of selected features and the number of estimated parameters. To get

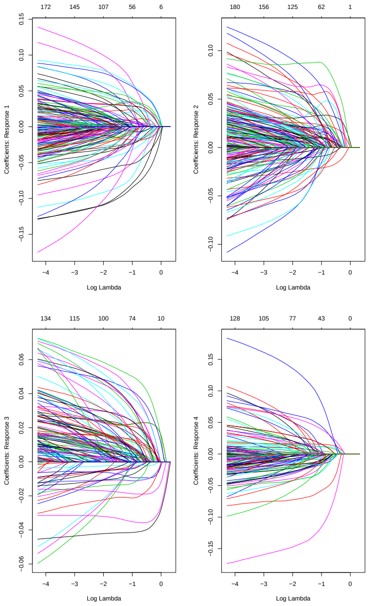


Figure 3.5. Shrinkage effect of elastic net and coefficient path.

the best tuning parameter λ, like the previous method, we do a 10-fold cross validation to select the best tuning parameter. Also for the mean square prediction error of this method, 10-fold cross validation is used and an average is taken among the mean square prediction error.

3.4 Bootstrapping for the tuning parameter

The bootstrap method can provide us with some statistical inference about the estimated parameters when there is little prior information about the distribution of the parameters. The nonparametric resampling method is based on resampling the observed data with replacement. The vector resampling is one method of nonparametric bootstrap that generates the new data as a sample from a certain bivariate distribution. Considering acquiring the distribution of tuning parameters, we have the bootstrap procedure designed as the follow,

Table 3.1
Description of fitted models.

Index	**lambda**	**features**	**Parameters**
1	1.00	1	4
20	0.75	6	17
40	0.55	22	55
60	0.41	31	75
80	0.30	50	117
100	0.22	57	135

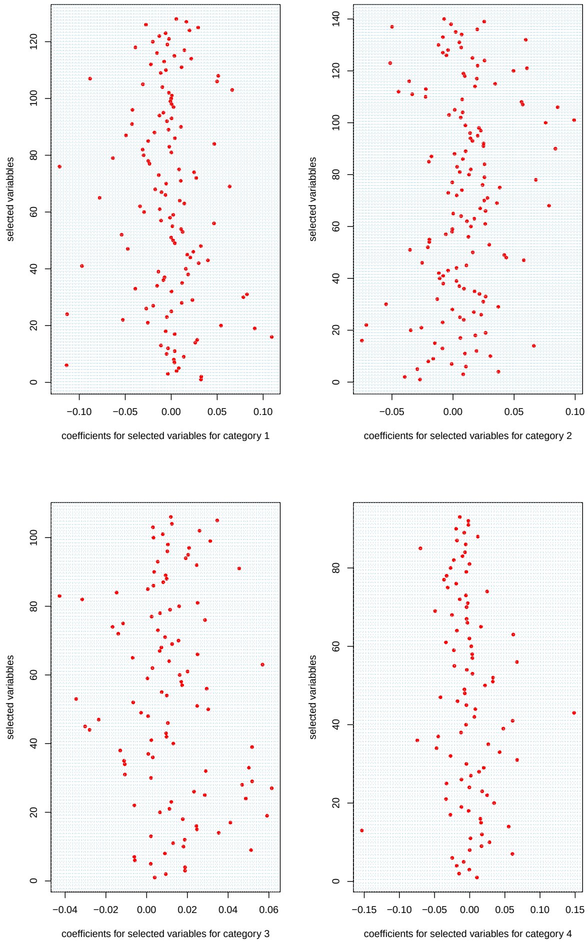


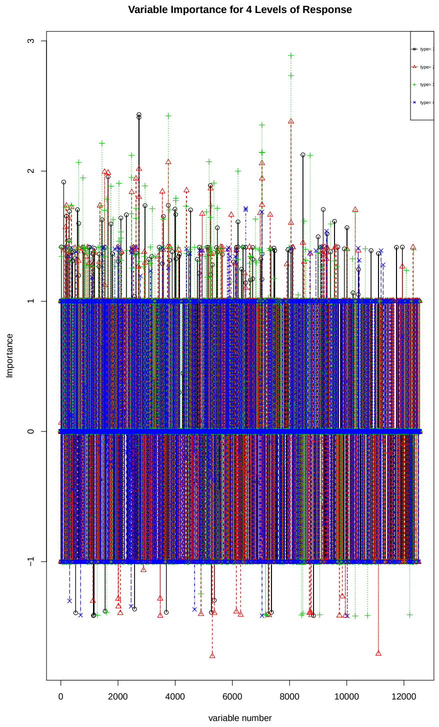
Figure 3.6. Coefficient plot via elastic net.

- Sampling with replacement from the original data set of (X,Y), and generate bootstrapped pairs and combine them as the bootstrapped sample

- Based on the bootstrapped sample, get the best estimate for the tuning parameters

- Repeat step one and step two

Based on this statistical procedure, we obtain the sample distribution for the tuning parameters under both "Lasso" model and "elastic net" model that are displayed as figure(3.7) and figure(3.8). Through the Q-Q plots of the estimates for tuning parameters,



we can conclude that in "Lasso" model the sample distributions for both λ_{min} and λ_{1se} are asymptotic normal. However, the sample distributions for tuning parameters α and λ in elastic net model are highly skewed and do not display normal distribution characteristics. As a results the bootstrap confidence interval for those tuning parameters are constructed under different assumptions. For the tuning parameters of "Lasso" model, we construct such interval based on the asymptotic normal-distributed feature of the sample distribution while for the tuning parameters in "elastic net" model we use the percentile method. The confidence intervals are described as table (3.2)

It is worth mentioning that based on the bootstrap method, λ_{opt} obtained by cross validation does not lie in the confidence interval of λ_{opt} obtained by bootstrap resampling but it does lies in the confidence interval of $\lambda - 1se$. And for "elastic net" model, bootstrap method does not perform very well since α obtained by grid search training does not lie in

Table 3.2
Confidence interval obtained by bootstrap.

Lasso:λ_{opt}	**Lasso:**λ_{1se}	**Elsticnet:**λ	**Elsticnet:**α
(0.0001362765,0.0280269155)	(0.02507274,0.09970878)	(0.001, 7.515212)	(0,0.7368421)

the confidence interval obtained by bootstrap method. It may be conclude that the basic bootstrap method under the brute-force searching algorithm does not yield relatively satisfying results in the context of statistics for high-dimensional data.

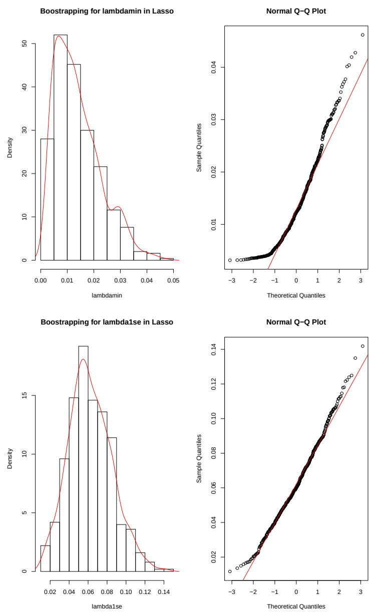


Figure 3.8. Boostrapping for lambda in Lasso.

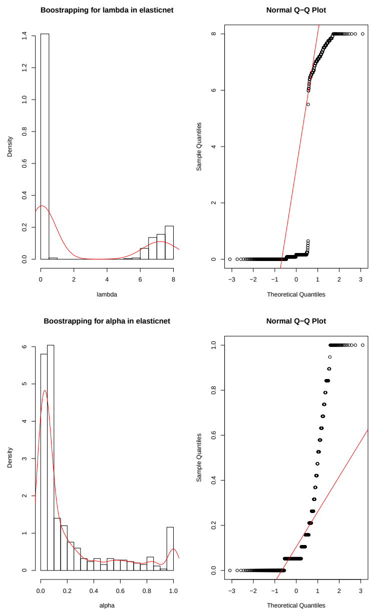


Figure 3.9. Boostrapping for lambda and alpha in elasticnet.

3.5 Prediction accuracy and model comparison

Since both regressions with Lasso or elastic net are optimization problems, we have relative fewer criterion to test that whether the fitted model is good. One criterion that we can use is the false classification rate of the fitted model. Table (3.3) shows the false classification rate for 5 methods of classification in high dimensional data setup. In terms of prediction accuracy, we use 10 fold cross validation to get the false classification rate. Under our motivated data, the multinomial logistic regression model with sparse group lasso has the best prediction accuracy, and multinomial logistic regression models with lasso and elastic net are less powerful in prediction compared with the previous model. Based on these corresponding results, multinomial logistic regression model with Lasso provide us with a better fit and a more parsimonious model since Lasso method selects less variables into the model compared to elastic net method. Due to the fundamental difference between ℓ_1 geometry and ℓ_2 geometry, under certain choice of λ, the Lasso method could guarantee that the coefficients of some groups of variables are exactly zero.

But there are also some issues regarding the criterion and procedure for model selection and model prediction. First, the false classification rate could be sensitive to the sample size. In our motivated data, since there are 77 different samples that is relatively small, the false classification rate might not be representative enough. Second, since in general the problem of Lasso could be regarded as optimization problem that yield little statistical test like confidence interval and p-value, we have relatively less information to combine Lasso with classic statistical test. As a result, we have less information about the uncertainty of our selected model and hypothesis testing, and the most direct message we can utilize is the prediction power of our model. Third, since the prediction error is

Table 3.3
False classification rate.

Lasso:λ_{opt}	**Elsticnet:**λ_{opt}	**Sparse group lasso**	**random forest**	**SVMs**
25.3%	27.5%	24.5%	29.2%	30.2%

almost zero with Lasso method, we can not rule out the possibility of overfitting phenomenon. Also, in the aspect of optimization algorithm, there exists no test regarding the convergence.

`glmnet` pointed out that the code provided in `glmnet` package does not implement any checks for divergence, since this check would slow down the computing for the solution, where the fast speed of computation with coordinate descent is the main advantage against other optimization algorithm. The coordinate descent algorithm have a closed form of expression for the starting solutions and subsequent solutions are warm-started from the previous close-by solutions that generally make the quadratic approximations very accurate. There is no documented example of divergence problem so far.

4. Conclusion

In this thesis, we review basic theory of lasso (least absolute shrinkage and selection operator) regression method and its generalization method that is elastic net. We utilize such methods to analyze our motivated data that is really a high-dimensional data with the number of predictors is much greater than the number of samples. Both methods perform an obvious dimension-reduction effect and overcome some drawbacks that ordinary least square method and maximum likelihood estimator intrinsically have. In theory, the property of ℓ_1 geometry guarantee that the Lasso method could do the variable selection and model fitting at the same time, and the elastic net method as an compromise between Lasso and ridge regression method could also serve as a variable selection procedure. Under Lasso and elastic net regression method, the fitted models based on our motivated data display different complexity. The model fitted by Lasso tends to select fewer variables compared to the model fitted by elastic net, in which Lasso provides us with a more parsimonious model. Considering the model selection criterion based on prediction accuracy, Lasso outperforms elastic net method in the analysis of our motivated data. The best model with smallest false classification rate is multiclass logistic regression with sparse group lasso penalty that allows sparsity within each selected feature. Our motivated data has very few samples but huge number of features, and it is highly possible that among those features exist large amounts of noises. Since all the penalized regression methods have relatively high false classification rate, we also utilized machine learning algorithms to compare models. It turns out that machine learning methods provides us with an even worse classification power. Thus, the penalized regression methods (Lasso, elastic net, and sparse group lasso) achieve performance and results as good as possible. Also the basic bootstrap method for getting the confidence interval for tuning parameter does not yield good results and in the future I want to investigate more about the construction of confidence interval for statistical model based on high-dimensional data and do more post inference in statistics for high-dimensional data.

APPENDIX

A. Some R code

Lasso:

```
Fold=function(Z=10,w,D,seed=7777){
  n=nrow(w)
  d=1:n;dd=list()
  e=levels(w[,D])
  T=length(e);set.seed(seed)
  for(i in 1:T){
    d0=d[w[,D]==e[i]];j=length(d0)
    ZT=rep(1:Z,ceiling(j/Z))[1:j]
    id=cbind(sample(ZT,length(ZT)),d0);dd[[i]]=id}
mm=list();for(i in 1:Z)
{u=NULL;for(j in 1:T)u=c(u,dd[[j]][dd[[j]][,1]==i,2])
mm[[i]]=u}
return(mm)}  # generate cross validation set
w=read.csv(file="/Users/tekinosei/Downloads/data1␣(1).csv",
header=TRUE, sep=",")
w=w[,-1]
con<-as.factor(w[,12547])
w=w[,-12547]
```

```
w=cbind(w,con) # load data


# 10-fold CV

library(glmnet)

D=12547;Z=10;n=nrow(w);mm=Fold(Z,w,D,1202)

Z=10

E=rep(0,Z)

for(i in 1:Z){m=mm[[i]]

n1=length(m)

w1<-as.matrix(w[,-12547])

cvlambda<-cv.glmnet(w1[-m,],w[-m,D],family="multinomial")

a=predict(cvlambda,newx=w1[m,],s=cvlambda$lambda.min,type="class")

E[i]=sum(w[m,D]!=a)/n1

show(i)}

mean(E)
```

Elastic-net:

```
library(glmnet)

Fold=function(Z=10,w,D,seed=7777){

  n=nrow(w)

  d=1:n;dd=list()

  e=levels(w[,D])

  T=length(e);set.seed(seed)

  for(i in 1:T){
    d0=d[w[,D]==e[i]];j=length(d0)
    ZT=rep(1:Z,ceiling(j/Z))[1:j]
    id=cbind(sample(ZT,length(ZT)),d0);dd[[i]]=id}
mm=list();for(i in 1:Z){u=NULL;for(j in 1:T)
u=c(u,dd[[j]][dd[[j]][,1]==i,2])
mm[[i]]=u}
return(mm)}  # generate cross validation set
w=read.csv(file="/Users/tekinosei/Downloads/data1␣(1).csv",
header=TRUE, sep=",")
w=w[,-1]
con<-as.factor(w[,12547])
w=w[,-12547]
w=cbind(w,con) # load data
D=12547;Z=10;n=nrow(w);mm=Fold(Z,w,D,1202)
Z=10
E=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
n1=length(m)
w1<-as.matrix(w[,-12547])
glmlambda<-glmnet(w1[-m,],w[-m,D],family="multinomial",
alpha = 0.2424242,lambda = 0.08636364)
a=predict(glmlambda,newx=w1[m,],s=0.08636364,type="class")
E[i]=sum(w[m,D]!=a)/n1
```

```
show(i)}

mean(E)
```

Sparse group lasso:

```
library(msgl)

w=read.csv(file="/Users/tekinosei/Downloads/data1␣(1).csv",

header=TRUE, sep=",")

w=w[,-1]

con<-as.factor(w[,12547])

w=w[,-12547]

w=cbind(w,con) # load data

D=12547;Z=10;n=nrow(w);mm=Fold(Z,w,D,1202)

Z=10

E=rep(0,Z)

for(i in 1:Z){m=mm[[i]]

n1=length(m)

w1<-as.matrix(w[,-12547])

lseq<-lambda(w1[-m,],w[-m,D],lambda.min = 0.05)

a=fit(w1[-m,],w[-m,D],lambda =lseq)

E[i]=sum(min(Err(a,w1[m,],w[m,D])))

show(i)}

mean(E)
```

Random forest:

```
D=12547;Z=10;n=nrow(w);mm=Fold(Z,w,D,1202)
library(randomForest)
set.seed(1202)
Z=10
E=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
n1=length(m)
a=randomForest(con~.,w[-m,])
E[i]=sum(w[m,D]!=predict(a,w[m,]))/n1
show(i)}
mean(E)
```

Support vector machine:

```
library(e1071)
D=12547;Z=10;n=nrow(w);mm=Fold(Z,w,D,1202)
Z=10
E=rep(0,Z)
for(i in 1:Z){m=mm[[i]]
n1=length(m)
a=svm(con~.,data=w[-m,],kernal="sigmoid")
E[i]=sum(w[m,D]!=predict(a,w[m,]))/n1
show(i)}
mean(E) # False classification rate
```

REFERENCES

[1] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[2] I. Guyon and A. Elisseeff. An introduction to variable and feature selection., 2003.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[4] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

[5] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software*, 39(5):1–13, 2011.

[6] Max Kuhn. The caret package, 2009.

[7] P. McCullagh and J.A. Nelder. *Generalized Linear Models, Second Edition*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1989.

[8] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent, 2009.

[9] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015.

[10] Ryan J. Tibshirani. The lasso problem and uniqueness. *Electron. J. Statist.*, 7:1456–1490, 2013.

[11] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494, June 2001.

[12] Lukas Meier, Sara van de Geer, and Peter Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society. Series B*, 70(1):53–71, 2008.

[13] Vincent, M., Hansen, and N. R. Sparse group lasso and high dimensional multinomial classification. *Computational Statistics Data Analysis*, 71:771–786, 2014.

[14] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[15] Peng Zhao and Bin Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, December 2006.