

FIRST: Combining forward iterative selection and shrinkage in high dimensional sparse linear regression

WOOK YEON HWANG*, HAO HELEN ZHANG†, AND SUBHASHIS GHOSAL‡

We propose a new class of variable selection techniques for regression in high dimensional linear models based on a forward selection version of the LASSO, adaptive LASSO or elastic net, respectively to be called as *forward iterative regression and shrinkage technique* (FIRST), adaptive FIRST and elastic FIRST. These methods seem to work effectively for extremely sparse high dimensional linear models. We exploit the fact that the LASSO, adaptive LASSO and elastic net have closed form solutions when the predictor is one-dimensional. The explicit formula is then repeatedly used in an iterative fashion to build the model until convergence occurs. By carefully considering the relationship between estimators at successive stages, we develop fast algorithms to compute our estimators. The performance of our new estimators are compared with commonly used estimators in terms of predictive accuracy and errors in variable selection.

AMS 2000 SUBJECT CLASSIFICATIONS: Primary 62J05, 62J07; secondary 62J07.

KEYWORDS AND PHRASES: LASSO, elastic net, high dimension, sparsity, variable selection.

1. INTRODUCTION

Variable selection in linear models is a major statistical issue in contemporary data analysis because modern data typically involve a lot of predictors, many of which are nearly irrelevant. Such a sparse structure of the regression function actually allows us to estimate the regression function fairly accurately even when the number of predictors far exceeds the number of available observations. Removing irrelevant variables from the predictive model is essential since the presence of too many variables may cause overfitting and multicollinearity, which lead to poor prediction of future outcomes. Moreover, the presence of too many variables in the regression function makes the relation hard to interpret. Since we are typically interested in situations where the number of predictors p is much larger than the size of

the available sample n (large p , small n or LPSN problems), we simply restrict ourselves to a linear model. Note that predictors are necessarily correlated when $p > n$.

Traditional variable selection methods for linear models include forward selection where variables are added to the effective subset of predictors sequentially, backward selection where variables are gradually removed from the collection, and stepwise selection where variables may be either added or removed depending on predictive performance. Typically, to assess performance and effectiveness in variable selection, the mean squared error (MSE), adjusted R^2 [14], Mallows's C_p [14], prediction sum of squares (PRESS) [14], Akaike information criterion (AIC) [1], Bayesian information criterion (BIC) [15] and so on are used. However, methods based on such criteria generally runs into difficulty in LPSN problems. Nowadays, regularized linear regression methods such as the ridge regression [12], nonnegative garrote [3, 17], least absolute selection and shrinkage operator (LASSO) [16, 21], adaptive LASSO [20] and elastic net [19] are widely used for variable selection in linear models.

We consider the linear regression model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ where $\mathbf{Y} = (Y_1, \dots, Y_n)^T \in R^n$ is a vector of responses, \mathbf{X} is an $n \times p$ matrix for predictors, $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)^T \in R^p$ is a vector for parameters, $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_n)^T$ is an n -dimensional random error, whose components are uncorrelated random variables having mean zero and common variance σ^2 . Without loss of generality we assume the data are centered, so the intercept is not included in the regression model. Throughout this paper, we also assume the columns of X are standardized to length 1. Let $\mathbf{X}_1^T, \dots, \mathbf{X}_n^T$ stand for the rows of \mathbf{X} , so that the i th observation may be decomposed as $Y_i = \mathbf{X}_i^T \boldsymbol{\beta} + \varepsilon_i$, $i = 1, \dots, n$. The model parameter $\boldsymbol{\beta}$ is traditionally estimated by the method of least squares minimizing $(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$, which leads to the best linear unbiased estimator $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$, provided that the model is of full rank, that is, $\text{rank}(\mathbf{X}^T \mathbf{X}) = \text{rank}(\mathbf{X}) = p$. However, the variance-covariance matrix of $\hat{\boldsymbol{\beta}}$ given by $\sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$ becomes very unstable when components of the predictors are nearly linearly related. In particular, if the number of observations n is less than the number of predictors p , $\mathbf{X}^T \mathbf{X}$ is necessarily singular so that a generalized inverse matrix should be considered for $\mathbf{X}^T \mathbf{X}$. In such cases, the least square estimator is not unique and may be

*Corresponding author.

†Research is supported by NSF and NIH/NCI.

‡Research is supported by NSF.

biased, and only certain linear combinations of β can be unbiasedly estimated.

To stabilize the variability of $\hat{\beta}$, [12] proposed the ridge estimator which minimizes a penalized sum of squares $(\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta) + \lambda\|\beta\|^2$ and is given by $\hat{\beta}^R = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{Y}$. The estimator essentially introduces a shrinkage on $\hat{\beta}$ towards zero, and may be viewed as a Bayes estimator with a normal prior on β .

In LPSN problems, typically many variables are insignificant, that is, they have regression coefficient exactly or approximately equal to zero. It is essential to filter out insignificant variables in order to reduce the model to manageable level and reduce prediction errors while estimating the regression coefficients. Thus it is desirable to shrink an insignificant regression coefficient to exactly zero, so that the corresponding variable will be automatically eliminated. [3] suggested the *non-negative garrote* estimator defined by

$$(1) \quad \hat{\beta}^G = \arg \min_{\mathbf{c} \geq \mathbf{0}} \left\{ \|\mathbf{Y} - \mathbf{X}(\hat{\beta} \circ \mathbf{c})\|^2 + \lambda \sum_{j=1}^p c_j \right\},$$

where \circ stands for componentwise (Hadamard) multiplication of vectors and λ is a Lagrangian multiplier. Under the non-negativity restriction, the second term in the expression above stands for the ℓ_1 -norm of the vector \mathbf{c} . Thus the non-negative garrote essentially looks for a minimizer of sum of squares within a weighted ℓ_1 -ball intersected with the cone defined by the least square estimator. The crispy nature of the boundary of ℓ_1 -balls implies that the solution will often be located on the boundary where many of the components are exactly zero. Thus the non-negative garrote shrinks the least square estimator towards zero and sets small coefficients to exactly zero.

The non-negative garrote still depends on the least-square estimator in its formulation, which is an inconvenience in LPSN problems. [16] proposed the LASSO, where an ℓ_1 -penalty is imposed, but the vector $\hat{\beta} \circ \mathbf{c}$ is replaced by an arbitrary vector β without any non-negativity constraint. In other words, the LASSO estimate is obtained by solving the non-linear optimization problem

$$(2) \quad \hat{\beta}^L = \arg \min_{\beta} \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \sum_{j=1}^p |\beta_j| \right\},$$

where $\lambda > 0$ is a Lagrangian multiplier which is used to regulate the penalty level.

When the regressors are highly correlated, the LASSO sometimes shows some erratic behavior in selecting variables arbitrarily from a group of correlated variables, and in reversing the sign of the estimate of a regression coefficient as the smoothing parameter varies. To avoid these shortcomings, [19] suggested imposing a quadratic penalty in addition to LASSO's ℓ_1 -penalty and called the resulting

procedure an elastic net (EN). Thus the elastic net estimator with smoothing parameter $\lambda_1 > 0$ and $\lambda_2 > 0$ is defined as $\hat{\beta}^{EN} = (1 + \lambda_2)\tilde{\beta}$, where

$$(3) \quad \tilde{\beta} = \arg \min_{\beta} \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda_1 \sum_{j=1}^p |\beta_j| + \lambda_2 \sum_{j=1}^p \beta_j^2 \right\}.$$

Intuitively, it is preferable to penalize different components differently in LASSO in tune with the size of some estimate of their regression coefficients, leading to the concept of adaptive LASSO [20]. Indeed, it can be shown that such estimators can possess an important oracle property [8, 20]. The adaptive LASSO can be described as the solution of the non-linear optimization problem

$$(4) \quad \hat{\beta}^{AL} = \arg \min_{\beta} \left\{ \|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \sum_{j=1}^p \frac{|\beta_j|}{|\tilde{\beta}_j|^\gamma} \right\}, \quad \lambda > 0, \gamma > 0,$$

where $\tilde{\beta}_j$ is an initial consistent estimator of β_j , $j = 1, \dots, p$. The least square estimator of β is a natural choice for $\tilde{\beta}$, but the choice suffers from LPSN problems where it is not unique because of singularity. The ridge regression estimator is a more sensible choice for $\tilde{\beta}$ [14]. Usually, one chooses $\gamma = 1$ in (4) where it was proven that the adaptive LASSO is the same as the non-negative garrote [17, 20].

While the LASSO and its variants are very useful for variable selection, the LASSO estimator does not have a closed form expression in general. For the non-negative garrote, the non-linear optimization problem (1) is a quadratic programming problem with a linear inequality constraint and can be solved by the technique of Lagrangian multipliers [2]. For the LASSO, the non-linear optimization problem (2) is a quadratic programming problem with linear inequality constraints. [16] described an algorithm based on ordinary least squares problem with 2^p inequality constraints. A faster algorithm for computation of the LASSO is given by the least angle regression (LARS) algorithm recently introduced by [7], which is available in R. A slight modification of the LARS algorithm can compute the elastic net estimator [19]. However, in typical applications where both p and n are large, the LARS solution may take a long time to compute.

The goal of the present paper is to construct LASSO-type estimators through sequential inclusion of variables in the model with a one-dimensional absolute value penalty. Thus our procedure is essentially a forward regression method with an absolute value penalty term, which we call the *forward iterative regression and shrinkage technique* (FIRST). However, unlike the LARS, our method does not give another computing algorithm for the LASSO, but actually gives a new estimator. In one dimension, our estimator agrees with the LASSO, but the two estimators are generally different in higher dimensional cases.

2. METHODOLOGY

2.1 Motivation

In the special case of orthogonal regressor variables, the nonnegative garrote, the LASSO and the elastic net estimator have closed-form expressions. Indeed, the non-negative garrote is given by

$$(5) \quad \hat{\beta}_j^G = \begin{cases} \hat{\beta}_j - \frac{\lambda}{2|\hat{\beta}_j|}, & \text{if } \hat{\beta}_j \geq \sqrt{\lambda/2}, \\ 0, & \text{if } |\hat{\beta}_j| < \sqrt{\lambda/2}, \\ \hat{\beta}_j + \frac{\lambda}{2|\hat{\beta}_j|}, & \text{if } \hat{\beta}_j \leq -\sqrt{\lambda/2}. \end{cases}$$

Similarly, for the LASSO, the corresponding expression is

$$(6) \quad \hat{\beta}_j^L = \begin{cases} \hat{\beta}_j - \frac{\lambda}{2}, & \text{if } \hat{\beta}_j \geq \lambda/2, \\ 0, & \text{if } |\hat{\beta}_j| < \lambda/2, \\ \hat{\beta}_j + \frac{\lambda}{2}, & \text{if } \hat{\beta}_j \leq -\lambda/2, \end{cases}$$

while for the *elastic net*, the corresponding expression is

$$(7) \quad \hat{\beta}_j^{EN} = \begin{cases} \frac{\hat{\beta}_j - \lambda_1/2}{1 + \lambda_2}, & \text{if } \hat{\beta}_j \geq \lambda_1/2, \\ 0, & \text{if } |\hat{\beta}_j| < \lambda_1/2, \\ \frac{\hat{\beta}_j + \lambda_1/2}{1 + \lambda_2}, & \text{if } \hat{\beta}_j \leq -\lambda_1/2. \end{cases}$$

In particular, since orthogonality trivially holds in one-dimension, the LASSO and the elastic net can be explicitly calculated if only one predictor is present. Therefore, if predictors are considered one at a time as in a forward selection procedure, the formulae (6) and (7) can be applied iteratively on the residual of the regression in the previous step. This leads to a new procedure, which we call the *forward iterative regression and shrinkage technique* (FIRST) and denote the resulting estimator by $\hat{\beta}^F$.

2.2 Description

Fix a maximum number of iteration steps, say M . Set initial $Y_{i,1} = Y_i$ for $i = 1, \dots, n$ and initial prediction vector $\hat{\mathbf{f}}_1 = 0$. For $m = 1, \dots, M$, repeat

1. **Standardization step:** Replace X_{ij} by $(X_{ij} - \bar{X}_j) / \{\sum_{l=1}^n (X_{lj} - \bar{X}_j)^2\}^{1/2}$ for $i = 1, \dots, n$ and $j = 1, \dots, p$, where $\bar{X}_j = n^{-1} \sum_{i=1}^n X_{ij}$. Thus from now onwards, we shall assume that $\bar{X}_j = 0$ and $\sum_{i=1}^n X_{ij}^2 = 1$ for all $j = 1, \dots, p$.
2. **Regression step:** Calculate the ordinary least square estimates, $\hat{\beta}_{1,m}, \dots, \hat{\beta}_{p,m}$, by $\hat{\beta}_{j,m} = \frac{\sum_{i=1}^n Y_{i,m} X_{ij}}{\sum_{i=1}^n X_{ij}^2}$, $j = 1, \dots, p$.
3. **Shrinkage step:** Calculate the LASSO estimates,

$$(8) \quad \hat{\beta}_{j,m}^L = \begin{cases} \hat{\beta}_{j,m} - \lambda/2, & \text{if } \hat{\beta}_{j,m} \geq \lambda/2, \\ 0, & \text{if } |\hat{\beta}_{j,m}| < \lambda/2, \\ \hat{\beta}_{j,m} + \lambda/2, & \text{if } \hat{\beta}_{j,m} \leq -\lambda/2. \end{cases}$$

4. **Selection step:** Select $X_{j_m^*}$ as the most effective variable in the m th iteration step, where $j_m^* = \arg_j \min \sum_{i=1}^n (Y_{i,m} - X_{ij} \hat{\beta}_{j,m}^L)^2$, $j = 1, \dots, p$.
5. **Updating stage:** Update m to $m+1$, $Y_{i,m}$ to $Y_{i,m+1} = Y_{i,m} - X_{ij_m^*} \hat{\beta}_{j_m^*,m}^L$, and $\hat{\mathbf{f}}_m$ to $\hat{\mathbf{f}}_{m+1} = \hat{\mathbf{f}}_m + \mathbf{X}_{j_m^*} \hat{\beta}_{j_m^*,m}^L$.
6. **Stopping rule:** Stop and get out of the loop after m steps if $\|\mathbf{Y} - \hat{\mathbf{f}}_m\|^2 - \|\mathbf{Y} - \hat{\mathbf{f}}_{m+1}\|^2 < \epsilon$, where $\epsilon > 0$ is predetermined.

The final estimator is denoted by $\hat{\beta}^F$. In practice, unless λ is very small, the estimator may have significant bias and hence its predictive ability may suffer. Letting the procedure to determine the subset of predictors that are finally used in regression but determining the coefficients by an ordinary least square method using only selected predictors may significantly improve the predictive ability of the estimator while utilizing sparsity in the same way. In other words, the final estimator is given by $\hat{\beta}^{FS} = (\mathbf{X}_S^T \mathbf{X}_S)^{-1} \mathbf{X}_S^T \mathbf{Y}$, where $S \subset \{1, \dots, p\}$ is the subset of predictors selected by FIRST and $\mathbf{X}_S = ((X_{ij}))_{1 \leq i \leq n, j \in S}$. In Section 3, we consider the performance of $\hat{\beta}^{FS}$ along with that of $\hat{\beta}^F$.

In a similar manner, we may consider the forward selection version of the one-dimensional adaptive LASSO (with $\gamma = 1$). The method, which will be called the adaptive FIRST, is obtained by modifying the shrinkage step in the description of FIRST as follows:

$$(9) \quad \hat{\beta}_{j,m}^{AL} = \begin{cases} \hat{\beta}_{j,m} - \frac{\lambda}{2|\hat{\beta}_{j,m}|}, & \text{if } \hat{\beta}_{j,m} \geq \frac{\lambda}{2|\hat{\beta}_{j,m}|}, \\ 0, & \text{if } |\hat{\beta}_{j,m}| < \frac{\lambda}{2|\hat{\beta}_{j,m}|}, \\ \hat{\beta}_{j,m} + \frac{\lambda}{2|\hat{\beta}_{j,m}|}, & \text{if } \hat{\beta}_{j,m} \leq -\frac{\lambda}{2|\hat{\beta}_{j,m}|}, \end{cases}$$

where $\tilde{\beta}$ is an initial consistent estimator of β . The final estimator is denoted by $\hat{\beta}^{AF}$. Also, we can perform an ordinary least square (OLS) analysis after variable selection in the same way as before. Then the final estimator is given by $\hat{\beta}^{AFS} = (\mathbf{X}_S^T \mathbf{X}_S)^{-1} \mathbf{X}_S^T \mathbf{Y}$, where $S \subset \{1, \dots, p\}$ is the subset of predictors selected by the adaptive FIRST and $\mathbf{X}_S = ((X_{ij}))_{1 \leq i \leq n, j \in S}$.

If we incorporate an additional square of the coefficient penalty term in addition to the absolute value term in our objective function so as to minimize $\sum_{i=1}^n (Y_i - X_{ij} \beta_j)^2 + \lambda_1 |\beta_j| + \lambda_2 \beta_j^2$ with respect to β_j and $j = 1, \dots, p$, we obtain a forward selection analog of the elastic net. The estimator, which we shall call the elastic FIRST, is obtained by modifying the shrinkage step in the algorithm for FIRST as follows:

$$(10) \quad \hat{\beta}_{j,m}^{EN} = \begin{cases} \frac{\hat{\beta}_{j,m} - \lambda_1/2}{1 + \lambda_2}, & \text{if } \hat{\beta}_{j,m} \geq \lambda_1/2, \\ 0, & \text{if } |\hat{\beta}_{j,m}| < \lambda_1/2, \\ \frac{\hat{\beta}_{j,m} + \lambda_1/2}{1 + \lambda_2}, & \text{if } \hat{\beta}_{j,m} \leq -\lambda_1/2. \end{cases}$$

The final estimator is denoted by $\hat{\beta}^{EF}$. Naturally, one can perform an ordinary least square analysis after the variable selection stage as before.

2.3 Basic properties

Our method FIRST (and its adaptive and elastic variants) satisfy the following simple properties:

1. The residual sum of squares in every iteration decreases. This happens since choosing the coefficient equal to zero is equivalent to sticking with the estimate obtained in the previous stage. Since the objective criterion is minimized, one always improves the residual sum of squares using the optimizing value instead of zero.
2. The algorithm converges (even when no artificial bound on the maximum number of steps is imposed). Since zero is a lower bound for residual sum of squares which decreases with every iteration, the amount of decrease must be eventually small prompting the algorithm to stop. Indeed, the maximum number of steps are bounded above by $(\sum_{i=1}^n Y_i^2)/\epsilon$, where ϵ is the accuracy level chosen in the stopping rule.
3. If the regressor variables are orthogonal, no variables can be included more than one time in the selection steps. This happens because if one variable were added to the model for the second time, the residual variable will be uncorrelated with that variable, and so the estimated coefficient is zero, prompting the algorithm to terminate.
4. In the orthogonal case, the maximum number of steps cannot be more than the number of variables considered for regression. Since no variables are repeated in the iterations, it is clear that there can be no more than p steps.
5. In the orthogonal case, the FIRST solution coincides with the LASSO solution and the adaptive FIRST solution coincides with the adaptive LASSO solution. In this special case, the ordinary least squares solution is $\hat{\beta}_j = \sum_{i=1}^n Y_i X_{ij}$, $j = 1, \dots, p$. The FIRST selects the variables one by one in the order of $|\hat{\beta}_j|$'s: the larger $|\hat{\beta}_j|$, the larger the reduction in estimation error by including the corresponding variable, the earlier the corresponding variable is added to the model. If $|\hat{\beta}_j| > \frac{\lambda}{2}$, then the variable j is selected by the FIRST at some stage and its effect is estimated as $(|\hat{\beta}_j| - \frac{\lambda}{2})_+ \text{sign}(\hat{\beta}_j)$ provided the set precision level ϵ is sufficiently small. Furthermore, those variables associated with $|\hat{\beta}_j| < \frac{\lambda}{2}$ are never added to the model by the FIRST. Thus the algorithm will stop as soon as all variables j with $|\hat{\beta}_j| > \frac{\lambda}{2}$ are exhausted, and the remaining coefficients are estimated as zero. Similarly, we can show that the adaptive FIRST gives the same solution as the adaptive LASSO.

Based on the model selection result of the LASSO [18], the orthogonal design satisfies the irrepresentable condition, therefore the FIRST is consistent for model selection under the orthogonal design when λ_n is chosen properly. Also, it is known that the adaptive LASSO has the oracle property when the model is tuned properly [20]. Therefore we have the following theorem:

Theorem 2.1. *Assume that the design is orthogonal with p fixed. The FIRST can select the model consistently if $\lambda_n/n \rightarrow 0$ and $\lambda_n/\sqrt{n} \rightarrow \infty$ as $n \rightarrow \infty$. If the initial estimator $\tilde{\beta}$ is \sqrt{n} -consistent, then the adaptive FIRST performs like an oracle if $\lambda_n \rightarrow \infty$ and $\lambda_n/\sqrt{n} \rightarrow 0$ as $n \rightarrow \infty$.*

2.4 Recursive algorithm for FIRST

A substantial savings in computing time of the FIRST is possible by utilizing the following recursive relations between the least square estimates $\hat{\beta}_{j,m}$ for different levels m :

$$(11) \quad \hat{\beta}_{j,m+1} = \hat{\beta}_{j,m} - \hat{\beta}_{j_m^*,m}^L C_{j_m^*,j},$$

where $C_{j_m^*,j}$ is the j th element of the j_m^* th row of the correlation matrix \mathbf{C} , and the (j, k) th element of \mathbf{C} is given by the sample correlation

$$c_{j,k} = \sum_{i=1}^n X_{ij} X_{ik}$$

between the j th and k th predictor. This follows by observing that $Y_{i,m+1} = Y_{i,m} - \hat{\beta}_{j_m^*,m}^L X_{ij_m^*}$ and computing inner products with X_j 's. This leads to the following efficient algorithm for the computation of the FIRST.

Standardize the data first. Then for $m = 1, \dots, M$, repeat

1. **Regression step:** Calculate the ordinary least square estimates, $\hat{\beta}_{1,m}, \dots, \hat{\beta}_{p,m}$, by (11).
2. **Shrinkage step:** Calculate the LASSO estimates as in (8).
3. **Selection step:** Select $X_{j_m^*}$ as the most effective variable in the m th iteration step, where $j_m^* = \arg_j \min\{(\hat{\beta}_{j,m}^L)^2 - 2\hat{\beta}_{j,m}^L \hat{\beta}_{j,m}\}$, $j = 1, \dots, p$.
4. **Updating stage:** Update m to $m+1$ and $\hat{\mathbf{f}}_m$ to $\hat{\mathbf{f}}_{m+1} = \hat{\mathbf{f}}_m + \mathbf{X}_{j_m^*} \hat{\beta}_{j_m^*,m}^L$. Note that we have eliminated the need to compute the residuals.
5. **Stopping rule:** As before.

To see why the selection step 3 is the same as the selection step before, observe that

$$\begin{aligned} & \sum_{i=1}^n (Y_{i,m} - X_{ij} \hat{\beta}_{j,m}^L)^2 \\ &= \sum_{i=1}^n Y_{i,m}^2 - 2\hat{\beta}_{j,m}^L \sum_{i=1}^n X_{ij} Y_{i,m} + (\hat{\beta}_{j,m}^L)^2 \sum_{i=1}^n X_{ij}^2 \\ &= \sum_{i=1}^n Y_{i,m}^2 - 2\hat{\beta}_{j,m}^L \hat{\beta}_{j,m} + (\hat{\beta}_{j,m}^L)^2. \end{aligned}$$

The assertion follows since the first term does not depend on j . In a similar manner, an efficient algorithm for the adaptive FIRST (or the elastic FIRST) can be obtained by replacing $\hat{\beta}_{j_m^*,m}^L$ in (11) by $\hat{\beta}_{j_m^*,m}^{AL}$ (or $\hat{\beta}_{j_m^*,m}^{EN}$) and replacing (8) in the shrinkage step by (9) (or (10)).

2.5 Connections to other methods

Boosting is another iterative approach for building sparse models for high dimensional data. [9] gave a nice interpretation of boosting as a gradient tree boosting for multiple additive regression trees. [4] considered boosting with the squared error loss, called L_2 Boosting, and showed that L_2 Boosting for linear models produces consistent estimates. The L_2 Boosting is a very attractive procedure in the LPSN context, since it just solves a series of simple least squares regression problems. [5] pointed out the link between L_2 Boosting and the LASSO. Both boosting and the FIRST are algorithm-based methods which build the model iteratively, however, they work differently in several ways. Firstly, the L_2 Boosting repeatedly fits ordinary least squares for residuals to update the model, while the FIRST uses the LASSO fitting to update the model. Secondly, these two procedures use different schemes to update the model between iterations. At each step, the L_2 Boosting updates the model by adding the new term multiplying a small step size, say 0.1, which works as a shrinkage parameter to scale down the contribution of the newly added term. By contrast, the FIRST added a new term with the constant step size one, where the new term is subject to a soft-thresholding penalty via LASSO before being added to the model. Thirdly, the number of boosting iterations is an important tuning parameter for the L_2 Boosting, which determines the size of the final model and needs to be selected properly to avoid overfitting [4-6]. In the FIRST, λ is the tuning parameter which automatically incorporates the amount of shrinkage at each step and hence controls the goodness of fit for the final model. Consequently, λ also controls the stopping rule.

It is well-known that the LARS procedure provides a state-of-the-art algorithm to compute the entire solution path for LASSO-type problems. Recently, [10] proposed another fast algorithm, the coordinate-wise descent algorithm (CDA), for solving the regularized regression problems such as the nonnegative garrote, LASSO and EN. The main idea of the CDA is to successively minimize the objective function with respect to one parameter at a time, while holding the remaining parameters at their current values. We use the LASSO to describe the algorithm. At each step, the coordinate-wise descent algorithm solves

$$(12) \quad \arg \min_{\beta_j} \left\{ \sum_{i=1}^n (Y_i - \sum_{k \neq j} X_{ik} \hat{\beta}_k - X_{ij} \beta_j)^2 + \lambda \sum_{k \neq j} |\hat{\beta}_k| + \lambda |\beta_j| \right\},$$

where the minimization is with respect to β_j and all the remaining parameters β_k for $k \neq j$ are fixed at their current values $\hat{\beta}_k$. [10] point out that the CDA converges to the optimal solution and gives very competitive performance with the LARS procedure especially when p is large. Though the FIRST is not an algorithm for computing the LASSO solution, it selects important variables and updates the model iteratively based on one-dimensional LASSO fitting, so it

would be interesting to compare the FIRST with both LARS and the coordinate-wise descent algorithm.

3. SIMULATION

We now demonstrate performance of the FIRST methods under various settings. We focus on large p small n data generated from a high-dimensional sparse linear model

$$Y_i = \mathbf{X}_i^T \beta + \sigma \varepsilon_i, \quad i = 1, \dots, n,$$

where $\mathbf{X}_i \in R^p, p > n$, and ε_i 's are iid errors from $N(0, 1)$. For each method, we generate a training set to fit the model, a validation set to select the tuning parameter, and an independent test set to evaluate prediction accuracy of the resulting estimator. Both the training and tuning sets are of size n , and the test set is of size 1,000. We run 100 simulations for each experiment and report the average results.

We implement six variations of the FIRST methods, including the FIRST, adaptive FIRST (aFIRST), elastic FIRST (eFIRST), FIRST followed by OLS (FIRST+OLS), adaptive FIRST followed by the OLS (aFIRST+OLS), and elastic FIRST followed by the OLS (eFIRST+OLS). For comparison, we also include the LASSO and elastic net (ENET) results. Two algorithms are used to implement the LASSO and ENET: the LARS and the CDA, both available in R. The optimal tuning parameter for each method is chosen by a grid search using the validation set. All simulations are run on Dell Xeon Dual Core 3.6 GHz with 4096 MB RAM.

3.1 Simulation settings

We consider four experiment settings by allowing different sample sizes, error variances (or signal strength), and correlation scenarios among the covariates. Below is the detailed description for the examples.

- (a) In Example 1, we have $p = 1,000$ and $n = 100$ or 500. The covariates X_1, \dots, X_p are i.i.d. generated from $N(0, 1)$. The error variance $\sigma^2 = 1$. The true coefficient vector $\beta = (\beta_1, \dots, \beta_p)^T$ contains 10 non-zero coefficients, (3, 3, 3, 3, 1.5, 1.5, 1.5, 2, 2, 2), and the rest are zero coefficients. We let the locations of the non-zero coefficients be equally spaced. In particular, $\beta_1, \beta_{101}, \beta_{201}$ and β_{301} are 3, β_{401}, β_{501} and β_{601} are 1.5, and β_{701}, β_{801} and β_{901} are 2.
- (b) Example 2 is the same as Example 1, except that there is moderate correlation among important covariates. In particular, we assume $\text{corr}(X_i, X_j) = 0.5^{|i-j|}$ if both X_i and X_j are important, and is 0 otherwise.
- (c) Example 3 has the same setting as Example 1, except that all the covariates are moderately correlated: the pairwise correlation between X_i and X_j is $\text{corr}(X_i, X_j) = \rho^{|i-j|}$. Here $\rho = 0.5$. We consider $n = 100$ and two different error variances: $\sigma^2 = 1$ and $\sigma^2 = 4$.
- (d) Example 4 is the same as Example 3, except that the covariates are highly correlated with $\rho = 0.9$.

Table 1. Simulation results for Example 1 ($p = 1000, \sigma^2 = 1$, independent covariates)

Method	Test Error		Selection Error I		Selection Error II		Time	
	$n = 100$	$n = 500$	$n = 100$	$n = 500$	$n = 100$	$n = 500$	$n = 100$	$n = 500$
FIRST	2.46 (0.57)	1.17 (0.06)	0	0	6.24	5.17	14.0	14.6
aFIRST	2.31 (0.60)	1.15 (0.07)	0	0	5.87	3.99	12.4	11.2
eFIRST	2.30 (0.58)	1.16 (0.07)	0	0	5.28	5.37	72.2	75.7
FIRST+OLS	1.38 (0.28)	1.02 (0.04)	0	0	3.20	0.09	14.0	14.6
aFIRST+OLS	1.20 (0.15)	1.02 (0.04)	0	0	0.32	0.04	12.4	11.2
eFIRST+OLS	1.42 (0.22)	1.07 (0.05)	0	0	1.44	0.01	72.2	75.7
LASSO (LARS)	2.59 (0.83)	1.15 (0.05)	0	0	55.03	41.01	1.5	48.4
ENET (LARS)	3.28 (0.90)	1.76 (0.68)	0	0	63.54	24.31	428.8	755.6
LASSO (CDA)	2.57 (0.81)	1.16 (0.05)	0	0	92.01	86.07	0.4	1.1
ENET (CDA)	2.92 (0.95)	1.20 (0.08)	0	0	129.1	67.53	2.7	6.0

Table 2. Simulation results for Example 2 (moderate correlation among important covariates)

Method	Test Error		Selection Error I		Selection Error II		Time	
	$n = 100$	$n = 500$	$n = 100$	$n = 500$	$n = 100$	$n = 500$	$n = 100$	$n = 500$
FIRST	13.47 (6.88)	1.39 (0.11)	0.86	0	28.61	13.43	414.2	423.3
aFIRST	13.34 (6.58)	1.37 (0.10)	1.54	0	15.24	14.75	401.8	401.2
eFIRST	2.80 (0.82)	1.32 (0.10)	0	0	14.41	10.83	417.0	426.3
FIRST+OLS	4.24 (2.74)	1.08 (0.07)	0.96	0	8.16	0.05	414.2	423.4
aFIRST+OLS	6.14 (3.81)	1.08 (0.07)	0	0	6.15	0.02	401.9	401.2
eFIRST+OLS	1.40 (0.25)	1.08 (0.07)	0.01	0	0.58	0.17	417.0	426.3
LASSO (LARS)	1.82 (0.40)	1.14 (0.08)	0	0	26.78	32.99	1.7	47.7
ENET (LARS)	1.95 (0.45)	1.45 (0.49)	0	0	24.14	195.12	454.9	739.6
LASSO (CDA)	1.82 (0.40)	1.14 (0.08)	0	0	26.92	21.70	0.6	1.4
ENET (CDA)	1.82 (0.41)	1.14 (0.08)	0	0	36.02	28.36	3.7	7.4

3.2 Experiments and results

We compare ten different methods with regard to their prediction error, variable selection performance, and computation time. For prediction performance, we report the mean squared error evaluated on the test set. For variable selection, we report two types of selection errors: selection error I defined as the number of non-zero coefficients which are estimated as zero, and selection error II defined as the number of zero coefficients which are not estimated as zero. The computation cost is reported as the average time (in seconds) to obtain the final coefficients, including the tuning process.

Table 1 summarizes the simulation results for the simple independent case, while Table 2 assumes there is moderate correlation among important covariates. We observe that all the FIRST methods give quite competitive performance compared with the LASSO and the ENET in these examples. In summary, the adaptive FIRST followed by the OLS is best for Example 1 in terms of the test error, selection error I and selection error II; the elastic FIRST followed by the OLS is best for Example 2. This is not so surprising since the ENET methods are specially designed for handling correlated covariates. One may wonder why the FIRST followed by the OLS has a smaller selection error when the OLS simply re-estimates the regression coefficients after se-

lection by FIRST. One possible reason is that these two procedures select tuning parameters differently as well as stop at different times. Since a follow-up the OLS step generally improves quality of estimates of regression coefficients by reducing bias significantly, the FIRST followed by the OLS has a better prediction power prompting it to stop sooner without including some of the unimportant variables otherwise selected by the FIRST. A similar explanation applies to the comparison between the adaptive FIRST and the adaptive FIRST followed by the OLS. Interestingly, the LASSO solution obtained by the CDA and that by the LARS give very similar test errors, but the selection performance of the LASSO by the CDA is much worse. It is noted that the LASSO by the CDA tends to retain a large number of redundant variables in the final model.

With regard to the computation cost, the LASSO by the CDA is the fastest. One may note another anomaly that the computing time for the adaptive FIRST and the adaptive FIRST followed by the OLS in Tables 1 and 2 actually decreased when the sample size increased from $n = 100$ to $n = 500$. It may be noted that the role of n in computing time is limited only to the standardization step and the first step, while the dimension p plays a far more important role. Since in our examples n is much smaller than p , there may not be any significant impact of a larger value of n on computing time compared to computational burden

Table 3. Simulation results for Example 3 ($p = 1000, n = 100$, correlated covariates with $\rho = 0.5$)

Method	Test Error		Selection Error I		Selection Error II		Time	
	$\sigma^2 = 1$	$\sigma^2 = 4$	$\sigma^2 = 1$	$\sigma^2 = 4$	$\sigma^2 = 1$	$\sigma^2 = 4$	$\sigma^2 = 1$	$\sigma^2 = 4$
FIRST	2.85 (1.17)	9.58 (4.43)	0	0.09	32.69	34.11	460.7	477.4
aFIRST	2.48 (1.12)	8.55 (4.45)	0	0.14	14.17	12.49	440.1	462.2
eFIRST	2.67 (1.12)	9.20 (3.85)	0	0.05	29.35	32.74	464.7	481.1
FIRST+OLS	1.73 (0.67)	7.94 (3.28)	0	0.22	6.82	14.34	460.7	477.4
aFIRST+OLS	1.40 (0.24)	5.94 (2.34)	0	0	0.49	1.87	440.1	462.2
eFIRST+OLS	1.61 (0.54)	7.20 (2.61)	0.01	0.14	4.97	12.35	464.7	481.1
LASSO (LARS)	2.95 (1.12)	10.81 (3.68)	0	0.08	52.8	51.28	2.6	2.8
ENET (LARS)	3.03 (1.17)	10.84 (3.71)	0	0.07	49.87	52.21	624.2	630.1
LASSO (CDA)	2.96 (1.14)	10.84 (3.71)	0	0.08	79.86	60.8	0.9	1.0
ENET (CDA)	2.96 (1.14)	10.84 (3.71)	0	0.08	79.86	60.8	5.4	5.5

Table 4. Simulation results for Example 4 ($p = 1000, n = 100$, dependent covariate with $\rho = 0.9$)

Method	Test Error		Selection Error I		Selection Error II		Time	
	$\sigma^2 = 1$	$\sigma^2 = 4$	$\sigma^2 = 1$	$\sigma^2 = 4$	$\sigma^2 = 1$	$\sigma^2 = 4$	$\sigma^2 = 1$	$\sigma^2 = 4$
FIRST	7.97 (5.15)	13.53 (4.98)	1.76	2.58	31.30	31.84	464.0	439.1
aFIRST	8.18 (5.47)	13.18 (5.38)	2.36	3.07	18.33	14.98	448.8	424.1
eFIRST	5.47 (3.08)	11.42 (3.49)	0.55	1.47	27.38	31.05	470.1	444.0
FIRST+OLS	4.08 (2.87)	10.51 (3.94)	1.56	2.74	13.56	15.26	464.0	439.1
aFIRST+OLS	4.73 (3.76)	9.99 (3.95)	0	0	5.84	7.08	448.8	424.1
eFIRST+OLS	2.07 (1.02)	7.83 (2.30)	0.38	1.49	13.36	15.09	470.1	440.0
LASSO (LARS)	2.60 (0.73)	9.27 (2.01)	0	0.54	49.14	48.88	3.8	3.3
ENET (LARS)	2.66 (0.75)	9.32 (2.07)	0	0.56	50.08	48.06	625.4	557.8
LASSO (CDA)	2.61 (0.73)	9.28 (2.00)	0	0.54	50.24	49.79	1.0	0.9
ENET (CDA)	2.61 (0.73)	9.29 (2.01)	0	0.54	50.24	50.19	5.8	5.9

in subsequent steps, in each of which $O(p)$ calculations are needed. Moreover, as our iterative procedures are dependent on stopping times and relatively sooner convergence is expected for a more accurate procedure associated with higher sample size, it is possible to observe shorter computing time for larger sample size. Finally, computing time can vary up to some extent randomly depending on the number of jobs running on the server where all our programs ran. We also notice that as n increases from 100 to 500, the computation time of both the LASSO and the ENET given by the LARS algorithm seriously deteriorates, while other algorithms are not so significantly affected by the sample size. As expected by us, the performance of all the methods get better when the sample size increases and get worse when the number of redundant predictors increases.

Tables 3 and 4 respectively consider the scenarios where all the covariates are either moderately or highly correlated. We notice that the aFIRST+OLS works best when the covariates are moderately correlated, while the eFIRST+OLS is best in the high correlation case. The LASSO and the ENET results given by the CDA are among the fastest, but their selection error II is worse than other methods. Overall speaking, the FIRST algorithms generally lead to much leaner models with a better prediction performance than the LASSO.

4. REAL EXAMPLE

We consider the gene expression data and approaches used in [13]. There are totally 31,099 probe sets and 120 observations in this dataset. For high dimensional data like this, it is a common practice to use pre-screening to make the computation more manageable. Two stages of pre-screening were applied in our analysis. In the first pre-screening, we removed 3,815 probe sets whose maximum expression values are not greater than the 25th percentile of the entire probe set. In the second stage, we selected 3,000 probe sets with the largest variances among the remaining 27,283 probe sets. Then in our analysis, we used these 3,000 probe sets the predictors. Since these 3,000 predictors are very likely to be correlated with each other, we implemented the elastic FIRST and the elastic FIRST followed by the OLS, with $\epsilon = 0.001$ and $M = 200$. The ENET was also implemented respectively by the LARS and the CDA. We randomly select 100 training data and 20 test data. Five-fold cross validation is conducted with 100 training data in R 2.72. The test error, the number of non-zero estimates, and computation time are shown in Table 5. We observe that the elastic FIRST (eFIRST) gives the smallest test error, the ENET by the CDA is the second best, and the ENET by the LARS is the worst. In terms of the model size, the eFIRST followed by the OLS gives the most sparse model of size 7, the

Table 5. Real example results

Method	Test Error	Selected Genes	Time (min.)
eFIRST	0.00828	36	52.94
eFIRST+OLS	0.01057	7	52.94
ENET (LARS)	0.01216	31	1098.32
ENET (CDA)	0.00866	2095	3.22

eFIRST and the ENET produce similar model sizes 36 and 31, while the ENET by the CDA gives the largest model of size 2,095. With regard to computation time, the ENET by the LARS struggles with the problem and take almost 20 times longer than that of the elastic FIRST, and the ENET by the CDA is again the fastest.

5. DISCUSSION

We propose a new class of variable selection approaches for high dimensional sparse regression models and the recursive algorithms for computational reduction. Basically, the FIRST is a combination of one dimensional LASSO and forward selection. The FIRST takes an advantage of the closed-form solutions for the one-dimensional LASSO and forward selection of fitting residuals repeatedly. Furthermore, we extend the FIRST to the adaptive FIRST and the elastic FIRST by applying the same concepts to the adaptive LASSO and the ENET. We also consider an ordinary least square after applying the FIRST, the adaptive FIRST and the elastic FIRST. We finally derive a recursive algorithm from the relations between the successive least square estimates and residuals, which leads to substantial savings in computing time. Throughout the simulation study and a real data example, we show that our algorithms generally show very competitive performance in model prediction and selection accuracy when compared with the LASSO and the ENET. Our algorithms have reasonable computational cost, which make the methods useful for analyzing high dimensional sparse data especially if the sample size is also large. As a future work, we will study how to tune parameters more efficiently so that additional savings in computation can be achieved.

Received 6 May 2009

REFERENCES

[1] AKAIKE, H. (1973). Maximum likelihood identification of gaussian autoregressive moving average models. *Biometrika* **60** 255–265. [MR0326953](#)

[2] BAZARAA, M. and SHETTY, C. (1979). *Nonlinear Programming*. John Wiley. [MR0533477](#)

[3] BREIMAN, L. (1995). Better subset selection using the non-negative garrote. *Technometrics* **37** 373–384. [MR1365720](#)

[4] BÜHLMANN, P. (2006). Boosting for high-dimensional linear models. *Annals of Statistics* **34** 559–583. [MR2281878](#)

[5] BÜHLMANN, P. and HOTHORN, T. (2007). Boosting algorithms: regularization, prediction, and model fitting. *Statistical Science* **22** 447–505.

[6] BÜHLMANN, P. and YU, B. (2005). Boosting, model selection, lasso and nonnegative garrote. Technical report, ETH Zürich. Available at citeseer.ist.psu.edu/757058.html.

[7] EFRON, B., HASTIE, T., JOHNSTONE, I., and TIBSHIRANI, R. (2004). Least angle regression. *Annals of Statistics* **24** 407–499. [MR2060166](#)

[8] FAN, J. and LI, R. (2002). Variable selection for Cox’s proportional hazards model and frailty model. *Annals of Statistics* **30** 74–99. [MR1892656](#)

[9] FRIEDMAN, J. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis* **38** 367–378. [MR1884869](#)

[10] FRIEDMAN, J., HASTIE, T., HOFLING, H., and TIBSHIRANI, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics* **1** 302–332. [MR2415737](#)

[11] FRIEDMAN, J., HASTIE, T., and TIBSHIRANI, R. (2008). Regularization paths for generalized linear models via coordinate descent. Technical Report, Department of Statistics, Stanford University.

[12] HOERL, E. and KENNARD, W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* **12** 55–67.

[13] HUANG, J., MA, S., and ZHANG, C.-H. (2008). Adaptive Lasso for sparse high-dimensional regression models. *Statistica Sinica* **18** 1603–1618. [MR2469326](#)

[14] RAWLINGS, J., PANTULA, S., and DICKEY, D. (2001). *Applied Regression Analysis*. Springer, New York. [MR1631919](#)

[15] SCHWARZ, G. (1978). Estimating the dimension of a model. *Annals of Statistics* **6** 461–464. [MR0468014](#)

[16] TIBSHIRANI, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of Royal Statistical Society, Series B* **1** 147–169. [MR1379242](#)

[17] YUAN, M. and LIN, Y. (2007). On the nonnegative garrote estimator. *Journal of Royal Statistical Society, Series B* **69** 143–161. [MR2325269](#)

[18] ZHAO, P. and YU, B. (2006). On model selection consistency of Lasso. *Journal of Machine Learning Research* **7** 2541–2563. [MR2274449](#)

[19] ZOU, H. and HASTIE, T. (2005). Regularization and variable selection via the elastic net. *Journal of Royal Statistical Society* **67** 301–320. [MR2137327](#)

[20] ZOU, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association* **476** 1418–1429. [MR2279469](#)

[21] ZHU, J., ROSSET, S., HASTIE, T., and TIBSHIRANI, R. (2004). L1-norm support vector machines. *Advances in Neural Information Processing Systems*.

Wook Yeon Hwang
 Department of Statistics
 North Carolina State University
 Raleigh, NC 27695-8203
 E-mail address: whwang@ncsu.edu

Hao Helen Zhang
 Department of Statistics
 North Carolina State University
 Raleigh, NC 27695-8203
 E-mail address: hzzhang@stat.ncsu.edu

Subhashis Ghosal
 Department of Statistics
 North Carolina State University
 Raleigh, NC 27695-8203
 E-mail address: ghosal@stat.ncsu.edu