STA232B Final Project

Yanhao Jin, Noah Perry, Mingshuo Liu, Wookyeong Song, Ju-Sheng Hong, Yan-Yu Chen

{yahjin, njperry, mshliu, wksong, jsdhong, ynychen}@ucdavis.edu

Department of Statistics, University of California, Davis.

December 28, 2024

1 Introduction and Data Exploration

In this project, we briefly explore the diabetes dataset and fit the linear regression using shrinkage approach using Lasso and adaptive Lasso. To be specific, we (i) summarize the main results related to Lasso, Least Angle Regression (LARS), and Adaptive Lasso, (ii) implement Lasso and Adaptive Lasso algorithms and apply our algorithm to a dataset containing information about diabetes patients and select models using AIC and BIC; (iii) finally compare results with Efron et al. (2004) paper [EHJT04].

The diabetes data information is related to 442 diabetes patients [EHJT04]. The response y is a measure of disease progression over one year, which is a continuous variable. The 10 predictor variables are listed below.

	AGE	SEX	BMI	BP	Serum measurements						Response
Patient	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y
1	59	2	32.1	101	157	93.2	38	4	4.9	87	151
2	48	1	21.6	87	183	103.2	70	3	3.9	69	75
3	72	2	30.5	93	156	93.6	41	4	4.7	85	141
4	24	1	25.3	84	198	131.4	40	5	4.9	89	206
5	50	1	23.0	101	192	125.4	52	4	4.3	80	135
6	23	1	22.6	89	139	64.8	61	2	4.2	68	97
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
441	36	1	30.0	95	201	125.2	42	5	5.1	85	220
442	36	1	19.6	71	250	133.2	97	3	4.6	92	57

Besides, the correlation plots of the predictors in Diabetes Dataset is given in the left of Figure 1. The correlation figure indicates that some of the predictors in the data are strongly correlated. Besides, the histogram of response variables are given in the right of Figure 1. Based on the histogram, we see that the response variable is right-skewed.

2 Summary of Shrinkage Methods: Lasso

2.1 Introduction and Background

The LASSO approach, firstly developed by [Tib96], is a shrinkage method for estimation in linear models which minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. Formally, suppose we have an input vector $\mathbf{X}^T = (\mathbf{1}, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p)$ and want to predict a real-value output Y. The linear regression model has the form $f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$ where $\mathbf{X}$ is the $N \times (p+1)$ matrix with each row an input vector, and $\mathbf{y}$ is the N-vector of outputs in the training set. The lasso estimator which is is a constrained version of ordinary least squares estimator, is defined by

$$\hat{\beta}^{lasso} = \underset{\beta}{argmin} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \quad s.t. \quad \sum_{j=1}^{p} |\beta_j| \leqslant t \tag{1}$$

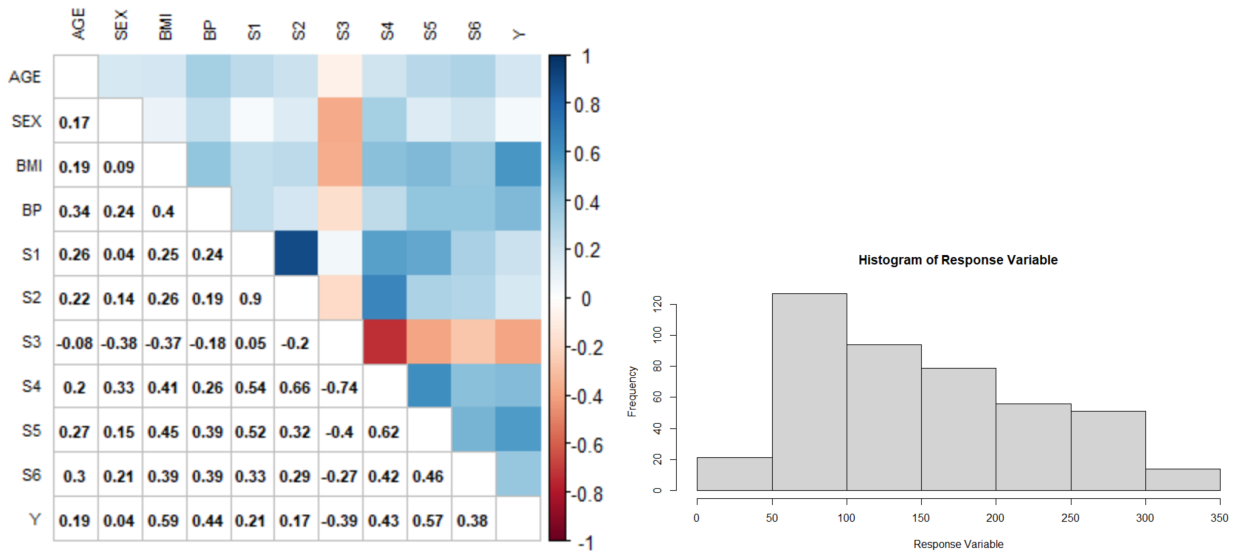

Figure 1: Correlation Plot (Left) and Histogram of the response variable (Right)

The lasso problem in the equivalent Lagrangian form

$$\hat{\beta}^{lasso} = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{2}$$

Similarly to the ridge regression problem: the L_2 penalty in ridge regression problem, $\sum_{j=1}^{p} \beta_j^2$, is replaced by the L_1 lasso penalty $\sum_{j=1}^{p} |\beta_j|$. This latter constraint makes the solutions nonlinear in the y_i, and there is no closed form expression as in ridge regression unless additional assumptions of the input matrix are made.

When the predictors are orthonormal $\mathbf{X}^T\mathbf{X} = \mathbf{I}$ and the ordinary least squared estimate of β is given by $\hat{\beta}^{OLS} = \mathbf{X}^T\mathbf{y}$. If we use the notation $\hat{\beta}_j^{lasso}$ and $\hat{\beta}_j^{OLS}$ to denote the j-th components of lasso estimator and ordinary least square estimator, it can be shown (proof attached in Appendix A.1) that in this case,

$$\begin{aligned} \hat{\beta}_j^{lasso} &= +\frac{\lambda}{2} + \hat{\beta}_j^{OLS} \qquad \hat{\beta}_j^{lasso} < 0 \\ \hat{\beta}_j^{lasso} &= -\frac{\lambda}{2} + \hat{\beta}_j^{OLS} \qquad \hat{\beta}_j^{lasso} > 0 \end{aligned} \tag{3}$$

By using the notation sign that denotes the signature, x_+ denotes the positive part of x and letting $\gamma = \frac{\lambda}{2}$, we will find the estimators given above can be written as

$$sign(\hat{\beta}_j^{OLS})(|\hat{\beta}_j^{OLS}| - \gamma)_+ \tag{4}$$

2.1.1 Geometrical Intuition of Lasso

The reason why Lasso can shrinkage some coefficients to zero comes from the geometrical properties of the penalty terms. Some insights for the case $p = 2$ can be provided by noticing that the target optimization function in regression problem

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - \sum_{j=1}^{p} \beta_j x_{i,j})^2$$

equals the quadratic function

$$\begin{aligned} RSS(\beta) &= (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) = (\mathbf{y} - \mathbf{X}\hat{\beta}^{OLS} + \mathbf{X}\hat{\beta}^{OLS} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\hat{\beta}^{OLS} + \mathbf{X}\hat{\beta}^{OLS} - \mathbf{X}\beta) \\ &= RSS(\hat{\beta}^{OLS}) + (\beta - \hat{\beta}^{OLS})^T\mathbf{X}^T\mathbf{X}(\beta - \hat{\beta}^{OLS}) + 2(\mathbf{y} - \mathbf{X}\hat{\beta}^{OLS})^T(\mathbf{X}\hat{\beta}^{OLS} - \mathbf{X}\beta) \end{aligned} \tag{5}$$

which is an oblique elliptical in geometry view. For $p = 2$, the residual sum of squares has elliptical contours, centered at the full least squares estimate. The constraint region for ridge regression is the disk $\beta_1^2 + \beta_2^2 \leqslant t$, while that for lasso is the diamond $|\beta_1| + |\beta_2| \leqslant t$. Both methods find the first point where the elliptical contours hit the constraint region. Unlike the disk, the diamond has corners; if the solution occurs at a corner, then it has one parameter β_j equal to zero. When $p > 2$, the diamond becomes a rhomboid, and has many corners, flat edges and faces; there are many more opportunities for the estimated parameters to be zero.

2.1.2 Standard Errors

Since the lasso estimator is a non-linear and non-differentiable function of the response values even for a fixed value of t, it is difficult to obtain an accurate estimate of its standard error. By writing the penalty $\sum |\beta_j|$ as $\sum |\beta_j|^2/|\beta_j|$, an approximate closed form estimate for the lasso estimate $\tilde{\beta}$ can be derived to be of the form

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{W}^-)^{-1}\mathbf{X}^T\mathbf{y}$$

where $\mathbf{W}$ is a diagonal matrix with diagonal elements $\tilde{\beta}_j$, $\mathbf{W}^-$ denotes the generalized inverse of $\mathbf{W}$ and λ is chosen so that $\sum |\hat{\beta}_j| = t$. The covariance matrix of the estimates may then be approximated by

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{W}^-)^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{W}^-)^{-1}\hat{\sigma}^2$$

where $\hat{\sigma}^2$ is an estimate of the error variance.

2.1.3 Algorithms for Finding Lasso Solutions

Computing the lasso solution is a quadratic programming problem. We fix $t \geqslant 0$ and recall that the lasso is a shrinkage method which is defined by:

$$\hat{\beta}^{lasso} = \underset{\beta}{argmin} \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \quad s.t. \quad \sum_{j=1}^{p} |\beta_j| \leqslant t$$

Therefore, the problem can be expressed as a least squares problem with 2^p inequality constraints, corresponding to the 2^p different possible signs for the β_j's. The naive approach to calculating the Lasso estimator follows from the ingredients for a procedure which solves the linear least squares problem subject to a general linear inequality constraint in [LH95]. Suppose that the linear inequality constraint is given by $\mathbf{G}\beta \leqslant \mathbf{h}$ and $\mathbf{G}$ is an $m \times p$ matrix, corresponding to m linear inequality constraints on the p vector β, let $g(\beta) = \sum_{i=1}^{N}(y_i - \sum_j \beta_j x_{ij})^2$ and δ_i, $i = 1, 2, \ldots, 2^p$ be the p-tuples of the form $(\pm 1, \pm 1, \ldots, \pm 1)$, then the condition $\sum |\beta_j| \leqslant t$ is equivalent to $\delta_i^T\beta \leqslant t$ for all i. For a given β, let

$$E = \left\{ i : \delta_i^T\beta = t \right\} \qquad S = \left\{ i : \delta_i^T\beta < t \right\}$$

The set E is the equality set, corresponding to those constraints which are exactly met, whereas S is the slack set, corresponding to those constraints for which equality does not hold. Denote by $\mathbf{G}_E$ the matrix whose rows are δ_i for $i \in E$. Let $\mathbf{1}$ be a vector of 1s of length equal to the number of rows of $\mathbf{G}_E$.

(a) Start with $E = \{i_0\}$ where $\delta_{i_0} = sign(\hat{\beta}^{OLS})$, $\hat{\beta}^{OLS}$ being the overall least squares estimate.

(b) $\hat{\beta}$ to minimize $g(\beta)$ subject to $\mathbf{G}_E\beta \leqslant t\mathbf{1}$.

(c) While $\left\{ \sum |\hat{\beta}_j| > t \right\}$

(d) Add i to the set E where $\delta_i = sign(\hat{\beta})$. Find $\hat{\beta}$ to minimize $g(\beta)$ subject to $\mathbf{G}_E\beta \leqslant t\mathbf{1}$.

2.1.4 Further Discussion for Lasso

We can generalize ridge regression and the lasso. Consider the criterion for $q \geqslant 0$

$$\tilde{\beta} = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|^q \right\} \tag{6}$$

- The value $q = 0$ corresponds to variable subset selection, as the penalty simply counts the number of nonzero parameters;
- $q = 1$ corresponds to the lasso;
- $q = 2$ corresponds to ridge regression.

The case $q = 1$ (lasso) is the smallest q such that the constraint region is convex; non-convex constraint regions make the optimization problem more difficult. Looking again at the criterion (6), we might try using other values of q besides 0, 1, or 2. Values of $q \in (1, 2)$ suggest a compromise between the lasso and ridge regression. Although this is the case, with $q > 1$, $|\beta_j|_q$ is differentiable at 0, and so does not share the ability of lasso ($q = 1$) for setting coefficients exactly to zero. Partly for this reason as well as for computational tractability, [ZH05] introduced the **elastic-net penalty**

$$\lambda \sum_{j=1}^{p} (\alpha\beta_j^2 + (1 - \alpha)|\beta_j|)$$

a different compromise between ridge and lasso.

2.2 Adaptive Lasso Regression

2.2.1 Introduction and Background

Based on lasso, [ZH05] comes up with the adaptive lasso, which enjoys the oracle properties:

- Identifies the right subset model, $\left\{ j : \hat{\beta}_j \neq 0 \right\} = \mathcal{A}$.
- Has the optimal estimation rate, $\sqrt{n} \left(\hat{\boldsymbol{\beta}}(\delta)_{\mathcal{A}} - \boldsymbol{\beta}^*_{\mathcal{A}} \right) \rightarrow_d \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma}^*)$, where $\boldsymbol{\Sigma}^*$ is the covariance matrix knowing the true subset model.

2.2.2 Definition

[ZH05] has shown that the lasso cannot be an oracle proceduree by giving a counterexample. Authors created new methodology called adaptive lasso assigning different weights to different coefficients. They have shown that if the weights are data-independent and cleverly chosen, then the weighted lasso can have the oracle properties. Suppose that $\hat{\boldsymbol{\beta}}$ is a root-n-consistent estimator to $hat\boldsymbol{\beta}^*$; for example, $\hat{\boldsymbol{\beta}}(\mathrm{ols})$ can be used. Pick $\gamma > 0$, and define the weight vector $\hat{w} = \frac{1}{|\hat{\boldsymbol{\beta}}|^\gamma}$. The adaptive lasso estimates $\hat{\boldsymbol{\beta}}^{*(n)}$ are given by

$$\hat{\boldsymbol{\beta}}^{*(n)} = \underset{\boldsymbol{\beta}}{\arg\min} \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 + \lambda_n \sum_{j=1}^{p} \hat{w}_j \left| \beta_j \right|. \tag{7}$$

2.2.3 Oracle Properties

Suppose that $\lambda_n / \sqrt{n} \rightarrow 0$ and $\lambda_n n^{(\gamma-1)/2} \rightarrow \infty$. Then the adaptive lasso estimates must satisfy the following:

- Consistency in variable selection: $\lim_n P\left(\mathcal{A}_n^* = \mathcal{A}\right) = 1$
- Asymptotic normality: $\sqrt{n} \left(\hat{\boldsymbol{\beta}}^{*(n)}_{\mathcal{A}} - \boldsymbol{\beta}^*_{\mathcal{A}} \right) \rightarrow_d \mathrm{N}\left(\mathbf{0}, \sigma^2 \times \mathbf{C}_{11}^{-1}\right)$.

2.3 Least Angle Regression

The algorithm given in section 2.1.3 is computational inefficiently and in many cases, Least Angle Regression Algorithm can be used to calculate the Lasso Estimator.

2.3.1 Algorithm for LAR

Least angle regression (LAR) is firstly introduced in [EHJT04] and is a modified version of forward stepwise regression. Following the ideas by [EHJT04], LAR is closely related with the lasso and it provides an efficient framework for computing the entire lasso path. The general framework of Least Angle Regression is given below

---

1. Standardize all predictors to have a zero mean and unit variance. Begin with all regression coefficients at zero i.e. $\beta_1 = \beta_2 = \cdots = \beta_p = 0$. The first residual will be $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, since with all $\beta_j = 0$ and standardized predictors the constant coefficient $\beta_0 = \bar{y}$
2. Find the predictor $\mathbf{x}_j$ most correlated with $\mathbf{r}$.
3. Move β_j from 0 towards its least-squares coefficient $\langle \mathbf{x}_j, \mathbf{r} \rangle$, until some other competitor $\mathbf{x}_k$ has as much correlation with the current residual as does $\mathbf{x}_j$.
4. Move β_j and β_k in the direction defined by their joint least squares coefficient of the current residual on $(\mathbf{x}_j, \mathbf{x}_k)$, until some other competitor $\mathbf{x}_l$ has as much correlation with the current residual.
5. Continue in this way until all p predictors have been entered. After $\min(N-1, p)$ steps, we arrive at the full least-squares solution.

In particular, at the first step, LAR algorithm finds the variable most correlated with the response. Instead of fitting this variable completely, LAR moves the coefficient of this variable continuously toward its least squares value and it will cause its correlation with the evolving residual to decrease in absolute value. This moving process is stopped when another variable "catches up" in terms of correlation with the residual and this variable will be put in the active set. After that, their coefficients are moved together in a way that keeps their correlations tied and decreasing. This process is continued until all the variables are in the model, and ends at the full least-squares fit. The details for the framework of the LAR algorithm is given in Appendix

2.3.2 LAR: Lasso Modification

The LAR algorithm can be adapted to calculate the Lasso estimator as suggested in [EHJT04], whose framework is given below.

- If a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction.

The reason behind what happens in the LAR and LAR lasso modification is given in the proposition below:

Proposition 2.1. *Suppose $\mathcal{A}$ is the active set of variables at one certain stage in the LAR algorithm, tied in their absolute inner-product with the current residuals $\mathbf{y} - \mathbf{X}\beta$ in a sense that*

$$\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\beta) = \gamma \cdot s_j, \forall j \in \mathcal{A} \tag{8}$$

where $s_j \in \{-1, 1\}$ is the sign of the inner-product, and γ is the common value. At this step, it holds that $\left|\mathbf{x}_k^T(\mathbf{y} - \mathbf{X}\beta)\right| \leq \gamma$, $\forall k \notin \mathcal{A}$. Now consider the lasso criterion in form of

$$R(\beta) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1 \tag{9}$$

Let $\mathcal{B}$ be the active set of variables in the solution for a given value of λ. For these variables $R(\beta)$ is differentiable, and the stationarity conditions give

$$\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\beta) = \lambda \cdot \operatorname{sign}(\beta_j), \forall j \in \mathcal{B} \tag{10}$$

Comparing (10) with (8), it can be shown that they are identical only if the sign of β_j matches the sign of the inner product. Therefore, LAR algorithm and lasso start to differ when an active coefficient passes through zero; condition (10) is violated for that variable, and it is kicked out of the active set $\mathcal{B}$. Lemma A.2 shows that these equations imply a piecewise-linear coefficient profile as λ decreases. The stationarity conditions for the non-active variables require that

$$\left|\mathbf{x}_k^T(\mathbf{y} - \mathbf{X}\beta)\right| \leq \lambda, \forall k \notin \mathcal{B} \tag{11}$$

which again agrees with the LAR algorithm.

3 Model Selection Criteria

Model selection refers to the fact that, while we may consider including all available predictors in an $n \times p$ design matrix $\mathbf{X}$ to predict the response Y through the linear model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \varepsilon, \quad \text{with } \varepsilon \sim N_n\left(\mathbf{0}, \sigma^2\mathbf{I}_n\right),$$

it may happen that the model involving a smaller subset of the predictors can fit the data better, in terms of greater predictive accuracy or better estimation risk. And the submodel is expressed as the following,

$$\mathbf{Y} = \mathbf{X}_{(k)}\boldsymbol{\beta}_{(k)} + \varepsilon, \quad \text{with } \varepsilon \sim N_n\left(\mathbf{0}, \sigma^2\mathbf{I}_n\right)$$

where k is used as an index to identify individual submodels. There are three principal approaches to model selection in the above context,

- Approach based on unbiased estimation of risk.
- Approach based on estimating the predictive risk.
- Likelihood-based approaches.

Here, two dominant criterion Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) are used in our data analysis part. Detailed descriptions are given in the following sections.

3.1 Degrees-of-Freedom Formula for LAR and Lasso

When AIC and BIC are considered, one needs to calculate the degree of freedom of Lasso. One traditional approach introduced in [Tib96] is to approximate the degree of freedom by

$$p(t) = \operatorname{tr}\left\{\mathbf{X}\left(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda\mathbf{W}^{-}\right)^{-1}\mathbf{X}^{\mathrm{T}}\right\}$$

where $\mathbf{W}$ is a diagonal matrix with diagonal elements $\tilde{\beta}_j$, $\mathbf{W}^{-}$ denotes the generalized inverse of $\mathbf{W}$ and λ is chosen so that $\sum|\hat{\beta}_j| = t$. However, [Tib96] did not provide the formal definition of the degree of freedom. The degree of freedom of Lasso is discussed in [ZHT07]. Formally,

Definition 3.1. The degrees of freedom of the fitted vector $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N)$ are given by

$$\operatorname{df}(\hat{\mathbf{y}}) = \frac{1}{\sigma^2}\sum_{i=1}^{N}\operatorname{Cov}(\hat{y}_i, y_i) \tag{12}$$

Here $\operatorname{Cov}(\hat{y}_i, y_i)$ refers to the sampling covariance between the predicted value $\hat{y}_i$ and its corresponding outcome value y_i.

The intuition behind the degree of freedom is very straight forward. The harder that we fit to the data, the larger this covariance and hence $\mathrm{df}(\hat{\mathbf{y}})$. As shown in [EHJT04] and [ZHT07], after the k-th step of the LAR procedure, the effective degrees of freedom of the fit vector is exactly k. Now for the lasso, the (modified) LAR procedure often takes more than p steps, since predictors can drop out. Hence the definition is a little different; for the lasso, at any stage $\mathrm{df}(\hat{\mathbf{y}})$ approximately equals the number of predictors in the model.

[ZHT07] provided a useful bootstrap approach to calculate the degree of freedom following the ideas from [Ste81]. Given a model fitting method δ, let $\hat{\boldsymbol{\mu}} = \delta(\mathbf{y})$ represent its fit. It is assumed that given the $\mathbf{x}$'s, $\mathbf{y}$ is generated according to $\mathbf{y} \sim \left(\boldsymbol{\mu}, \sigma^2\mathbf{I}\right)$, where $\boldsymbol{\mu}$ is the true mean vector and σ^2 is the common variance. Then the degrees of freedom of δ is

$$df(\hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} \mathrm{cov}\left(\hat{\mu}_i, y_i\right)/\sigma^2 \tag{13}$$

Suppose that $\mathbf{y}$ is used to fit an ordinary least square model. We compute the OLS estimates $\hat{\beta}_{\mathrm{ols}}$ and $\hat{\sigma}^2_{\mathrm{ols}}$. Then we consider a synthetic model,

$$\mathbf{y}^* = \mathbf{X}\beta + N(0,1)\sigma \tag{14}$$

where $\beta = \hat{\beta}_{\mathrm{ols}}$ and $\sigma = \hat{\sigma}_{\mathrm{ols}}$. Given the synthetic model (14), the degrees of freedom of the lasso can be numerically evaluated by Monte Carlo methods.

- For $b = 1, 2, \ldots, B$, we independently simulate $\mathbf{y}^*(b)$ from (14).
- Compute
$$\widehat{\mathrm{cov}}_i = \frac{\sum_{b=1}^{B}\left(\hat{\mu}_i(b) - a_i\right)\left(y_i^*(b) - (\mathbf{X}\beta)_i\right)}{B}$$
- Finally, $df = \sum_{i=1}^{n} \widehat{\mathrm{cov}}_i/\sigma^2$. Typically $a_i = 0$ is used in Monte Carlo calculation. In this work we use $a_i = (\mathbf{X}\beta)_i$, for it gives a Monte Carlo estimate for df with smaller variance than that given by $a_i = 0$.

3.1.1 Efron's Conjecture.

In [EHJT04], Efron first considered deriving the analytical form of the degrees of freedom of the lasso. In particular, following conjecture on the degrees of freedom of the lasso is presented

Conjecture 3.1. Starting at step 0, let m_k^{last} be the index of the last LARS-lasso sequence containing exactly k nonzero predictors. Then $df\left(\hat{\boldsymbol{\mu}}_{m_k^{\mathrm{last}}}\right) = k$.

In general, [ZHT07] argues that the conjecture is true under the so-called "positive cone condition". Without the positive cone condition the conjecture can be wrong, although k is a good approximation of degree of freedom. In particular, [ZHT07] shows that conjecture works appropriately from the model selection perspective. If we use the conjecture to construct AIC (or BIC) to select the lasso fit, then the selected model is identical to that selected by AIC (or BIC) using the exact degrees of freedom results.

3.2 AIC

AIC [Boz87] is derived from the perspective of Kullback-Leibler (KL) divergence. The KL divergence ($KL_f(k)$) of $f_k\left(\cdot \mid \widehat{\boldsymbol{\theta}}_k\right)$ with respect to $f(\cdot)$ can be expressed as

$$KL_f(k) = \mathbb{E}_f(\log f(\mathbf{Y})) - \mathbb{E}_f\left(\log f_k\left(\mathbf{Y} \mid \widehat{\boldsymbol{\theta}}_k\right)\right) \tag{15}$$

And the goal of AIC is to minimize the $KL_f(k)$ between the true model and submodels. Under certain assumptions that

- For every k, there is a unique parameter value $\boldsymbol{\theta}_k^0$ such that $-\mathbb{E}_f\left(\log f_k\left(\mathbf{Y} \mid \boldsymbol{\theta}_k\right)\right)$ is minimized over the parameter space Θ_k at $\boldsymbol{\theta}_k^0$;

- The density $f(\cdot)$ can be approximated by the density $f_k\left(\cdot \mid \boldsymbol{\theta}_k^0\right)$.

The final form of AIC can be expressed as

$$AIC(k) = n \log \widehat{\sigma}_k^2 + 2p_k \tag{16}$$

where p_k is the degree of freedom of the submodel. Notice that, in this expression, the first term is simply $-2 \log f_k\left(\mathbf{Y} \mid \widehat{\boldsymbol{\theta}}_k\right)) + n - n \log(2\pi)$. Ignoring the constant term, $-2 \log f_k\left(\mathbf{Y} \mid \widehat{\boldsymbol{\theta}}_k\right)) + 2p_k$ is being used as an estimator of $-2\mathbb{E}_f\left(\log f_k\left(\mathbf{Y} \mid \widehat{\boldsymbol{\theta}}_k\right)\right)$. Justification of this estimator as a good (meaning, consistent) estimator can be given through the large-sample theory of maximum likelihood estimators.

3.3 BIC

BIC [Sch78] is formulated from the Bayesian perspective. K is defined to be the random variable that takes value over the space of model indexed. And the Bayesian frame for BIC is given as,

- Let π_k denote the prior probability of the event $K = k$.
- Given $K = k$, let $g_k(\cdot)$ denote the prior density of the parameter $\boldsymbol{\theta}_k$, now treated as a random vector.
- Assume that the π_k 's are essentially equal (uniform prior on the model space), and $g_k\left(\boldsymbol{\theta}_k\right)$ is constant on Θ_k (i.e., prior for $\boldsymbol{\theta}_k$ is "flat" or "noninformative").
- Using Bayes' Theorem, the marginal posterior probability of submodel k, given observed data $\mathbf{Y} = \mathbf{y}$ is equal to

$$\mathbb{P}(K = k \mid \mathbf{Y} = \mathbf{y}) = \frac{\pi_k}{m(\mathbf{y})} \int_{\Theta_k} g_k\left(\boldsymbol{\theta}_k\right) f_k\left(\mathbf{y} \mid \boldsymbol{\theta}_k\right) d\boldsymbol{\theta}_k \tag{17}$$

  for a function $m(\mathbf{y})$ such that $\sum_k \mathbb{P}(K = k \mid \mathbf{y}) = 1$ for all $\mathbf{y}$, where the sum is taken over all possible values of k.

Then by the Laplace approximation method for integrals and ignore the contant term, $-2 \log \mathbb{P}(K = k \mid \mathbf{Y} = \mathbf{y})$ can be estimated by

$$-2 \log \int_{\Theta_k} f_k\left(\mathbf{Y} \mid \boldsymbol{\theta}_k\right) d\boldsymbol{\theta}_k \approx -2 \log f_k\left(\mathbf{Y} \mid \widehat{\boldsymbol{\theta}}_k\right) + (\log n) p_k \tag{18}$$

Excluding additive terms not depending on k in the right hand side of the last display, BIC can be expressed as

$$BIC(k) = n \log \widehat{\sigma}_k^2 + (\log n) p_k \tag{19}$$

3.4 Selecting λ by Cross Validation

For changing λ the number of variables with coefficients different from 0 changes. Which of the model should we choose. One could make the decision based on AIC or BIC. Another possibility is to apply **Cross Validation** for making this choice.

For Cross Validation the sample is split into K folds, $F_1, \ldots, F_k$, of equal size (or as close as possible). Then the model is fit K times: each time omitting one of the folds (let's say i) for estimating the model parameters.

The outcome of this estimation is then cross validated by comparing the actual measurements with their prediction for all data in the omitted fold k:

$$e_i(\lambda) = \sum_{j \in F_i} (y_j - \hat{y}_j(\lambda))^2$$

The CV error (also called the cross validation MSE) is the error averaged over the K folds.

$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^{K} e_i(\lambda)$$

When $K = n$ we call this the leave-one-out cross validation. More common choice for K are five or ten.

A plot of the cross-validation error against λ usually shows first a decrease in $CV(\lambda)$ and then an increase. Choose λ such that the cross validation error is smallest.

$$\hat{\lambda} = \underset{\lambda}{argmin}\, CV(\lambda)$$

Experimenters have found this choice very conservative not eliminating "sufficiently many" predictors from the model.

4 Data Analysis

4.1 Coefficients Analysis

In this part, we will give 5 plots of estimated coefficients according to the algorithms described above (LASSO in 'lars' package, LARS in 'lars' package, Stagewise in 'lars' package, our LASSO and our Adaptive LASSO). In figures, x axis is defined as the fraction of L_1 norm of estimated coefficients divided by the maximum of all L_1 norm. y axis represents estimated coefficients.

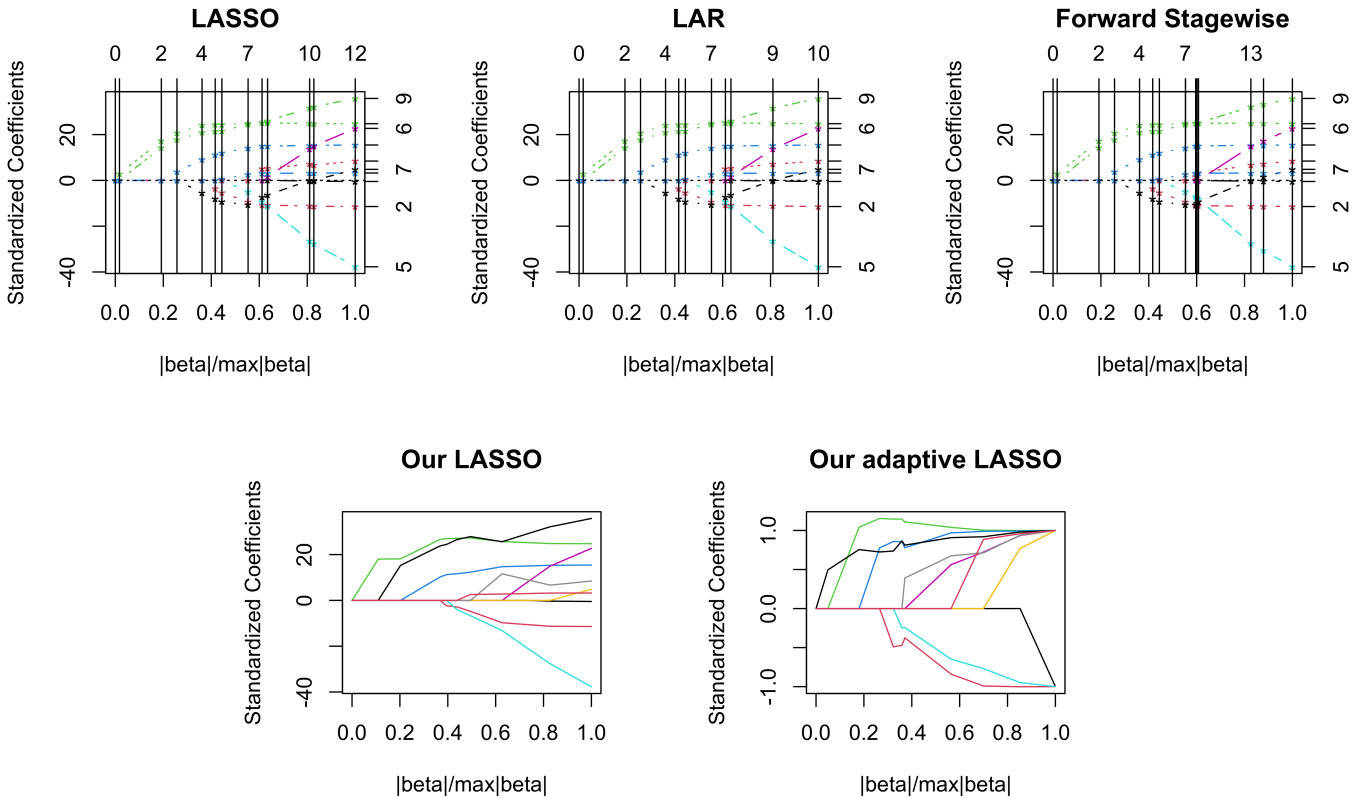

Figure 2: Coefficients Analysis

4.2 Variable Selection

In this part, we apply 3 different criteria (AIC, BIC, C_P) to each algorithm to implement variable selection. In figures, we start from a large value of lambda and then decreases it to see the change of the active variables. The x-axis indicates each step that there is a change of active variables. In the last step, lambda will become zero, all variables are active, and the coefficients will become the same as that from OLS.

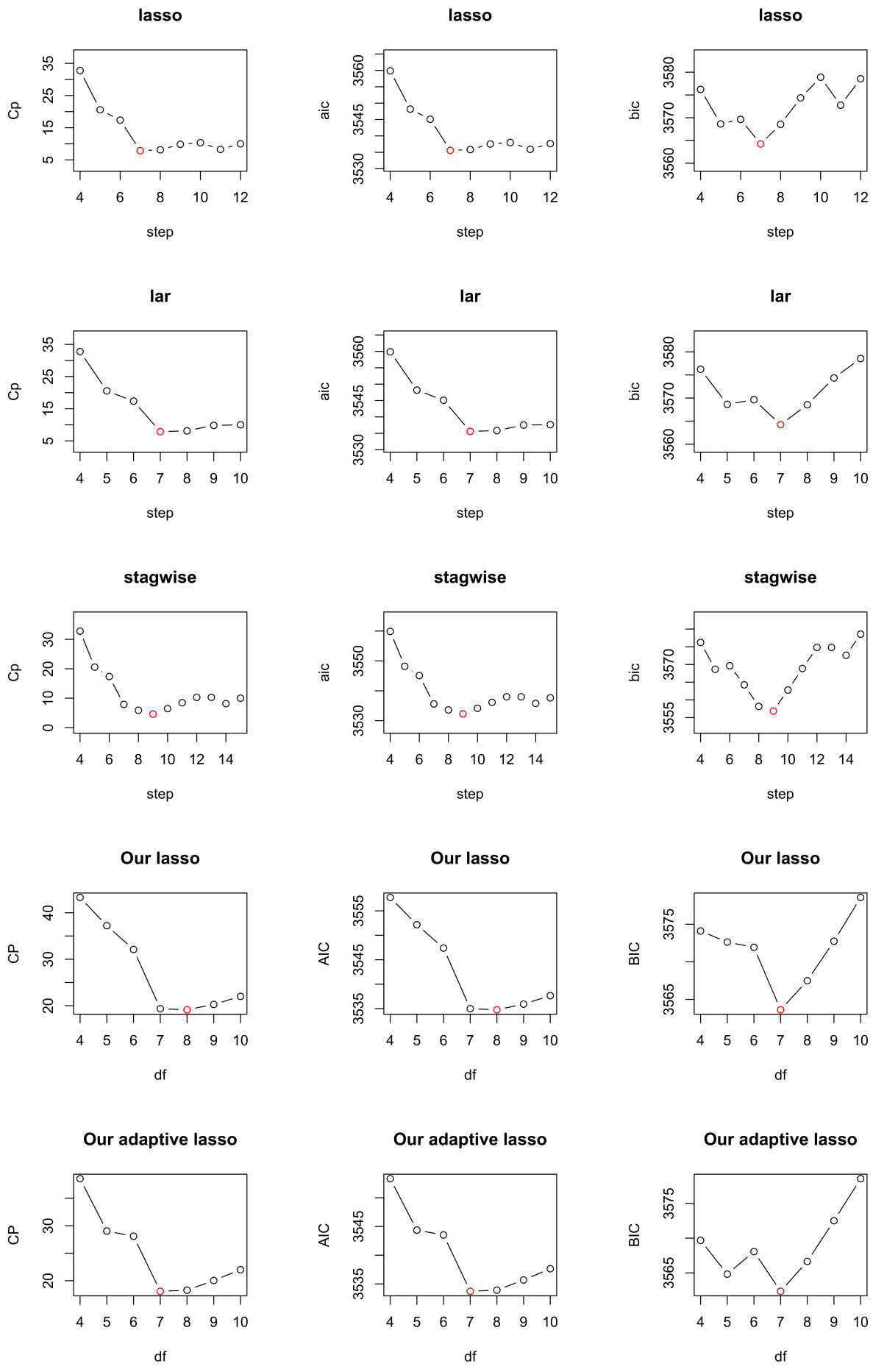

Figure 3: Variable Selection Analysis

4.3 Find the Best Model Based on MSE

In each model, we can compare the performance of our criteria based on MSE. MSE is $\frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2$, where y_i is an observed data and $\hat{y}_i$ is a predicted value from a combination of the model and the variable selection criteria.

Table 1: Model and its MSE Result

Model	Lasso	LARS	Stagewise	Our Lasso	Our Adaptive Lasso
Best Criteria	AIC, BIC, C_p	AIC,BIC, C_p	AIC, BIC, C_p	AIC, C_p	AIC, BIC, C_p
MSE	2961.39	2961.39	2995.752	2866.939	2873.114

In lars-Lasso, lars-LARS, lars-Stagewise and our adaptive lasso algorithms, selected variables from three criteria (C_p, AIC, BIC) are exactly same. But selected model from our Lasso algorithm based on AIC criteria has the minimum MSE.

Therefore, our final model from our LASSO algorithm is

$$Y = -10.59 \times \text{Sex} + 25.22 \times \text{BMI} + 15.00 \times \text{BP} - 20.88 \times \text{S1} + 7.94 \times \text{S2} + 9.00 \times \text{S4} + 29.06 \times \text{S5} + 2.98 \times \text{S6} \quad (20)$$

5 Conclusion

In this paper, we introduce famous shrinkage methods and various model selection criteria. We analyze the diabetes data by using these algorithms. We showed that the model from Lasso regression along the path by AIC has the lowest MSE value.

A Technical Details of the [Tib96] and [EHJT04]

A.1 Lasso in Orthonormal Design Case and the Influence of Multicollinearity on Lasso

When the predictors are orthonormal $\mathbf{X}^T\mathbf{X} = \mathbf{I}$ and the ordinary least squared estimate of β is given by

$$\hat{\beta} = \mathbf{X}^T\mathbf{y} \tag{21}$$

Since the columns of $\mathbf{X}$ are orthonormal we can construct a basis for $\mathbb{R}^N$ by using the first p columns of $\mathbf{X}$ and then extending these with $N-p$ linearly independent additional orthonormal vectors. The Gram-Schmidt procedure guarantees that we can do this. Thus in this extended basis we can write $\mathbf{y}$ as

$$\mathbf{y} = \sum_{j=1}^{p} \hat{\beta}_j \mathbf{x}_j + \sum_{j=p+1}^{N} \gamma_j \tilde{\mathbf{x}}_j \tag{22}$$

Where $\hat{\beta}_j$ equal the components of $\hat{\beta}$ in Equation (21), $\tilde{\mathbf{x}}_j$ are the extended basis vectors required to span $\mathbb{R}^N$, and γ_j are the coefficients of y with respect to these extended basis vectors.

For the lasso regression procedure we pick the values of β_j to minimize

$$F^{Lasso}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta) + \lambda \sum_{j=1}^{p} |\beta_j|$$

Expanding $\hat{\mathbf{y}}$ as $\hat{\mathbf{y}} = \sum_{j=1}^{p} \beta_j \mathbf{x}_j$ and with $\mathbf{y}$ expressed again as in Equation (22) we have that $RSS(\beta)$ in this case becomes

$$\begin{aligned} F^{Lasso}(\beta) &= ||\sum_{j=1}^{p}(\hat{\beta}_j - \beta_j)\mathbf{x}_j + \sum_{j=p+1}^{N} \gamma_j \tilde{\mathbf{x}}_j||^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\ &= \sum_{j=1}^{p}(\hat{\beta}_j - \beta_j)^2 + \sum_{j=p+1}^{N} \gamma_j^2 + \lambda \sum_{j=1}^{p} |\beta_j| \\ &= \sum_{j=1}^{p} \Big\{ (\hat{\beta}_j - \beta_j)^2 + \lambda|\beta_j| \Big\} + \sum_{j=p+1}^{N} \gamma_j^2 \end{aligned} \tag{23}$$

We can minimize this expression for each value of β_j for $1 \leqslant j \leqslant p$ independently. Thus our vector problem becomes that of solving p scalar minimization problems all of which look like

$$\beta^* = \underset{\beta}{argmin} \Big\{ (\hat{\beta} - \beta)^2 + \lambda|\beta| \Big\} \tag{24}$$

In this expression $\hat{\beta}$ and λ are assumed fixed. This expression can be represented as the sum of two terms $(\hat{\beta} - \beta)^2$ and $\lambda|\beta|$. The first expression $(\hat{\beta} - \beta)^2$ is symmetric about the least squares estimate $\hat{\beta}$ while the second expression is symmetric about $\beta = 0$.

Then the objective function $F(\beta)$ in Equation (24) we want to minimize is

$$F(\beta) = \begin{cases} (\beta - \hat{\beta})^2 - \lambda\beta & \beta < 0 \\ (\beta - \hat{\beta})^2 + \lambda\beta & \beta > 0 \end{cases} \tag{25}$$

To find the minimum of this function take the derivative with respect to β and set the result equal to zero and solve for β. We find the derivative of $F(\beta)$ given by

$$F'(\beta) = \begin{cases} 2(\beta - \hat{\beta}) - \lambda & \beta < 0 \\ 2(\beta - \hat{\beta}) + \lambda & \beta > 0 \end{cases} \tag{26}$$

When we set $F'(\beta)$ equal to zero we get two possible solutions for β given by

$$\begin{aligned} \beta &= +\frac{\lambda}{2} + \hat{\beta} \qquad \beta < 0 \\ \beta &= -\frac{\lambda}{2} + \hat{\beta} \qquad \beta > 0 \end{aligned} \tag{27}$$

If we use the notation $\hat{\beta}_j^{lasso}$ and $\hat{\beta}_j^{OLS}$, we have

$$\begin{aligned} \hat{\beta}_j^{lasso} &= +\frac{\lambda}{2} + \hat{\beta}_j^{OLS} \qquad \hat{\beta}_j^{lasso} < 0 \\ \hat{\beta}_j^{lasso} &= -\frac{\lambda}{2} + \hat{\beta}_j^{OLS} \qquad \hat{\beta}_j^{lasso} > 0 \end{aligned} \tag{28}$$

By using the notation sign that denotes the signature, x_+ denotes the positive part of x and letting $\gamma = \frac{\lambda}{2}$, we will find (28) can be written as

$$sign(\hat{\beta}_j^{OLS})(|\hat{\beta}_j^{OLS}| - \gamma)_+ \tag{29}$$

by comparing the graph of these two functions.

Finally, we consider the influence of collinearity on the regression coefficients on ridge regression and lasso.Suppose for a given t in (1), the fitted lasso coefficient for variable X_j is $\hat{\beta}_j = a$. Suppose we augment our set of variables with an identical copy

$$X_j^* = X_j$$

Now we characterize the effect of this exact collinearity by describing the set of solutions for $\hat{\beta}_j$ and $\hat{\beta}_j^*$, using the same value of t.

Let X_j be the feature that we dupilicate and let X_{-j} denote all other features except X_j. Let β_j denote the coefficient of X_j in the original Lasso problem, and let β_{-j} denote all the other coefficients. Then the **original Lasso problem** be written as the following optimization problem:

$$\begin{aligned} \hat{\beta} = &\text{argmin}_\beta \|Y - X_{-j}\beta_{-j} - X_j\beta_j\|_2^2 \\ &\text{s.t. } \|\beta_{-j}\|_1 + |\beta_j| \le t \end{aligned}$$

Let X_j^* denote the duplicated feature and let $\tilde{\beta}_j$ and β_j^* denote the coefficients of the original feature X_j and the duplicated feature X_j^* in the new Lasso problem. Let $\tilde{\beta}_{-j}$ denote the coefficients of other feature vectors in the new Lasso problem. Then the **updated Lasso problem** can be written as:

$$\begin{aligned} \hat{\beta} = &\text{argmin}_\beta \left\|Y - X_{-j}\tilde{\beta}_{-j} - X_j\tilde{\beta}_j - X_j^*\beta_j^*\right\|_2^2 \\ &\text{s.t. } \left\|\tilde{\beta}_{-j}\right\|_1 + \left|\tilde{\beta}_j\right| + |\beta_j^*| \le t \end{aligned}$$

Now say for a particular solution to the updated Lasso problem, our coefficients are: $\tilde{\beta}_{-j}$, $\tilde{\beta}_j$ and β_j^*. Now if we choose $\beta_{-j} = \tilde{\beta}_{-j}$, and $\beta_j = \tilde{\beta}_j + \beta_j^*$, then I claim this set of β_{-j} and β_j is also a solution to the original Lasso problem. Using Triangle Inequality (i.e. $|a+b| \le |a| + |b|$) we get:

$$\begin{aligned} &\left\|\tilde{\beta}_{-j}\right\|_1 + \left|\tilde{\beta}_j\right| + |\beta_j^*| \le t \\ \Longrightarrow &\left\|\tilde{\beta}_{-j}\right\|_1 + \left|\tilde{\beta}_j + \beta_j^*\right| \le t \end{aligned}$$

But we already know that for the given value of t, the optimal coefficient of X_j for the original Lasso problem is $\beta_j = a$. Therefore, this new coefficient $\tilde{\beta}_j + \beta_j^*$ also has to equal a. Further, we also know that the absolute value of each individual coefficient can never exceed t. Therefore we conclude the solution set for coefficients of X_j and X_j^* is characterized by the following line segment:

$$\begin{aligned} &\tilde{\beta}_j + \beta_j^* = a \\ \text{subject to } &\left|\tilde{\beta}_j\right| \le t, |\beta_j^*| \le t \end{aligned}$$

A.2 Details and Technical Lemmas for LAR

Formally, suppose $\mathcal{A}_k$ is the active set of variables at the beginning of the k-th step and $\beta_{\mathcal{A}_k}$ be the coefficient vector for these variables at this step, then the current residual is given by

$$\mathbf{r}_k = \mathbf{y} - \mathbf{X}_{\mathcal{A}_k}\beta_{\mathcal{A}_k}$$

and the direction for this step is

$$\delta_k = (\mathbf{X}_{\mathcal{A}_k}^T\mathbf{X}_{\mathcal{A}_k})^{-1}\mathbf{X}_{\mathcal{A}_k}^T\mathbf{r}_k \tag{30}$$

The coefficient profile then evolves as

$$\beta_{\mathcal{A}_k}(\alpha) = \beta_{\mathcal{A}_k} + \alpha \cdot \delta_k$$

the directions chosen in this fashion keep the correlations tied and decreasing. If the fit vector at the beginning of this step is $\hat{\mathbf{f}}_k$, then it evolves as

$$\hat{\mathbf{f}}_k(\alpha) = \hat{\mathbf{f}}_k + \alpha \cdot \mathbf{u}_k$$

where $\mathbf{u}_k = \mathbf{X}_{\mathcal{A}_k}\delta_k$ is the new fit direction.

Lemma A.1 (LAR directions)**.** Using the notation around equation (30), the LAR direction makes an equal angle with each of the predictors in $\mathcal{A}_k$.

Proof. From the definition of the LAR direction vector $\mathbf{u}_k$, we see that

$$\begin{aligned}\mathbf{X}_{\mathcal{A}_k}^T\mathbf{u}_k &= \mathbf{X}_{\mathcal{A}_k}^T\mathbf{X}_{\mathcal{A}_k}\delta_k \\ &= \mathbf{X}_{\mathcal{A}_k}^T\mathbf{X}_{\mathcal{A}_k}(\mathbf{X}_{\mathcal{A}_k}^T\mathbf{X}_{\mathcal{A}_k})^{-1}\mathbf{X}_{\mathcal{A}_k}^T\mathbf{r}_k \\ &= \mathbf{X}_{\mathcal{A}_k}^T\mathbf{r}_k\end{aligned} \tag{31}$$

Since the cosign of the angle of $\mathbf{u}_k$ with each predictor $\mathbf{x}_j$ in $\mathcal{A}_k$ is given by

$$\frac{\mathbf{x}_j^T\mathbf{u}_k}{||\mathbf{x}_j|| \cdot ||\mathbf{u}_k||} = \frac{\mathbf{x}_j^T\mathbf{u}_k}{||\mathbf{u}_k||}$$

each element of the vector $\mathbf{X}_{\mathcal{A}_k}^T\mathbf{u}_k$ corresponds to a cosign of an angle between a predictor $\mathbf{x}_j$ and the vector $\mathbf{u}_k$. Since the procedure for LAR adds the predictor $\mathbf{x}_{j'}$ exactly when the absolute value of $\mathbf{x}_{j'}^T\mathbf{r}$ equals that of $\mathbf{x}_j^T\mathbf{r}$ for all predictors $\mathbf{x}_j$ in $\mathcal{A}_k$, the direction $\mathbf{u}_k$ makes an equal angle with all predictors in $\mathcal{A}_k$. □

To derive a better connection between LAR Algorithm (a few steps of Least Angle Regression) and the notation on the general LAR step k that is presented in this section that follows LAR algorithm. It is helpful to perform the first few steps of this algorithm by hand and explicitly writing out what each variable was. In this way we can move from the specific notation to the more general expression.

- Standardize all predictors to have a zero mean and unit variance. Begin with all regression coefficients at zero i.e. $\beta_1 = \beta_2 = \cdots = \beta_p = 0$. The first residual will be $\mathbf{r} = \mathbf{y} - \bar{\mathbf{y}}$, since with all $\beta_j = 0$ and standardized predictors the constant coefficient $\beta_0 = \bar{y}$
- Set $k = 1$ and begin start the k-th step. Since all values of β_j are zero, the first residual is $\mathbf{r}_1 = \mathbf{y} - \bar{\mathbf{y}}$. Find the predictor $\mathbf{x}_j$ that is most correlated with this residual $\mathbf{r}_1$. Then as we begin this $k = 1$ step we have the active step given by $\mathcal{A}_1 = \{\mathbf{x}_j\}$ and the active coefficients given by $\beta_{\mathcal{A}_1} = [0]$.
- Move β_j from its initial value of 0 and in the direction

$$\delta_1 = (\mathbf{X}_{\mathcal{A}_1}^T\mathbf{X}_{\mathcal{A}_1})^{-1}\mathbf{X}_{\mathcal{A}_1}^T\mathbf{r}_1 = \frac{\mathbf{x}_j^T\mathbf{r}_1}{\mathbf{x}_j^T\mathbf{x}_j} = \mathbf{x}_j^T\mathbf{r}_1$$

  Note that the term $\mathbf{x}_j^T\mathbf{x}_j$ in the denominator is not present since $\mathbf{x}_j^T\mathbf{x}_j = 1$ as all variables are normalized to have unit variance. The path taken by the elements in $\beta_{\mathcal{A}_1}$ can be parametrized by

$$\beta_{\mathcal{A}_1}(\alpha) = \beta_{\mathcal{A}_1} + \alpha\delta_1 = 0 + \alpha\mathbf{x}_j^T\mathbf{r}_1 = (\mathbf{x}_j^T\mathbf{r}_1)\alpha \quad \text{for} \quad 0 \leqslant \alpha \leqslant 1$$

This path of the coefficients $\beta_{\mathcal{A}_1}(\alpha)$ will produce a path of fitted values given by

$$\hat{\mathbf{f}}_1(\alpha) = \mathbf{X}_{\mathcal{A}_1}\beta_{\mathcal{A}_1}(\alpha) = (\mathbf{x}_j^T\mathbf{r}_1)\alpha\mathbf{x}_j$$

and a residual of

$$\mathbf{r}(\alpha) = \mathbf{y} - \bar{\mathbf{y}} - \alpha(\mathbf{x}_j^T\mathbf{r}_1)\mathbf{x}_j = \mathbf{r}_1 - \alpha(\mathbf{x}_j^T\mathbf{r}_1)\mathbf{x}_j$$

Now at this point $\mathbf{x}_j$ itself has a correlation with this residual as α varies given by

$$\mathbf{x}_j^T(\mathbf{r}_1 - \alpha(\mathbf{x}_j^T\mathbf{r}_1)\mathbf{x}_j) = \mathbf{x}_j^T\mathbf{r}_1 - \alpha(\mathbf{x}_j^T\mathbf{r}_1) = (1-\alpha)\mathbf{x}_j^T\mathbf{r}_1$$

When $\alpha = 0$ this is the maximum value of $\mathbf{x}_j^T\mathbf{r}_1$ and when $\alpha = 1$ this is the value 0. All other features (like $\mathbf{x}_k$) have a correlation with this residual given by

$$\mathbf{x}_k^T(\mathbf{r}_1 - \alpha(\mathbf{x}_j^T\mathbf{r}_1)\mathbf{x}_j) = \mathbf{x}_k^T\mathbf{r}_1 - \alpha(\mathbf{x}_j^T\mathbf{r}_1)\mathbf{x}_k^T\mathbf{x}_j$$

Starting at the beginning of the k-th step of the LAR algorithm, derive expressions to identify the next variable to enter the active set at step $k+1$, and the value of α at which this occurs

Lemma A.2. Consider a regression problem with all variables and response having mean zero and standard deviation one. Suppose also that each variable has identical absolute correlation with the response:

$$\frac{1}{N}|\langle\mathbf{x}_j, \mathbf{y}\rangle| = \lambda \quad j = 1, \ldots, p$$

Let $\hat{\beta}$ be the least-squares coefficient of $\mathbf{y}$ on $\mathbf{X}$, and let $\mathbf{u}(\alpha) = \alpha\mathbf{X}\hat{\beta}$ for $\alpha \in [0, 1]$ be the vector that moves a fraction α toward the least squares fit $\mathbf{u}$. Let RSS be the residual sum-of-squares from the full least squares fit

- **(a)**: we have
$$\frac{1}{N}|\langle\mathbf{x}_j, \mathbf{y} - \mathbf{u}(\alpha)\rangle| = (1-\alpha)\lambda, \quad j = 1, \ldots, p$$
and hence the correlations of each $\mathbf{x}_j$ with the residuals remain equal in magnitude as we progress toward $\mathbf{u}$.
- **(b)**: These correlations are all equal to
$$\lambda(\alpha) = \frac{(1-\alpha)}{\sqrt{(1-\alpha)^2 + \frac{\alpha(2-\alpha)}{N} \cdot RSS}}$$
and hence they decrease monotonically to zero.

Proof. Now in the expression $\frac{1}{N}|\langle\mathbf{x}_j, \mathbf{y} - \mathbf{u}(\alpha)\rangle|$, the function $\mathbf{u}(\alpha)$ is a "scaled least squares solution" and takes the form $\mathbf{u}(\alpha) = \alpha\mathbf{X}\hat{\beta}$ where $\hat{\beta}$ is given by the least squares solution.

- **(a)**: Because of this $\frac{1}{N}|\langle\mathbf{x}_j, \mathbf{y} - \mathbf{u}(\alpha)\rangle|$ is the absolute value of the j-th component of
$$\begin{aligned}\frac{1}{N}\mathbf{X}^T(\mathbf{y} - \mathbf{u}(\alpha)) &= \frac{1}{N}\mathbf{X}^T(\mathbf{y} - \alpha\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}) \\ &= \frac{1}{N}(\mathbf{X}^T\mathbf{y} - \alpha\mathbf{X}^T\mathbf{y}) \\ &= \frac{1}{N}(1-\alpha)\mathbf{X}^T\mathbf{y}\end{aligned} \tag{32}$$
Since in this problem we are told that the absolute value of each element of $\mathbf{X}^T\mathbf{y}$ is equal to $N\lambda$ we have from the above that $\frac{1}{N}|\mathbf{X}^T(\mathbf{y} - \mathbf{u}(\alpha))| = (1-\alpha)\lambda$, or looking at the j-th row and taking absolute values of this expression we conclude that
$$\frac{1}{N}|\langle\mathbf{x}_j, \mathbf{y} - \mathbf{u}(\alpha)\rangle| = (1-\alpha)\lambda \tag{33}$$
for $j = 1, 2, \ldots, p$ as we were to show. In words, the magnitude of the projections of $\mathbf{x}_j$ onto the residual $\mathbf{y} - \mathbf{u}(\alpha) = \mathbf{y} - \alpha\mathbf{X}\hat{\beta}$ is the same for every value of j.

- **(b)**: The correlations (not covariances) would be given by

$$\frac{\frac{\langle \mathbf{x}_j, \mathbf{y}-\mathbf{u}(\alpha)\rangle}{N}}{\left(\frac{\langle \mathbf{x}_j,\mathbf{x}_j\rangle}{N}\right)^{\frac{1}{2}}\left(\frac{\langle \mathbf{y}-\mathbf{u}(\alpha),\mathbf{y}-\mathbf{u}(\alpha)\rangle}{N}\right)^{\frac{1}{2}}} = \frac{(1-\alpha)\lambda}{\left(\frac{\langle \mathbf{y}-\mathbf{u}(\alpha),\mathbf{y}-\mathbf{u}(\alpha)\rangle}{N}\right)^{\frac{1}{2}}}$$

using the result from **(a)**.
We next need to evaluate the expression $\langle \mathbf{y}-\mathbf{u}(\alpha), \mathbf{y}-\mathbf{u}(\alpha)\rangle$ in the denominator above. As a first step we have

$$\langle \mathbf{y}-\alpha\mathbf{X}\hat{\beta}, \mathbf{y}-\alpha\mathbf{X}\hat{\beta}\rangle = \mathbf{y}^T\mathbf{y} - \alpha\mathbf{y}^T\mathbf{X}\hat{\beta} - \alpha\hat{\beta}^T\mathbf{X}^T\mathbf{y} + \alpha^2\hat{\beta}^T(\mathbf{X}^T\mathbf{X}\hat{\beta})$$

Now recall the normal equations for linear regression

$$\mathbf{X}^T(\mathbf{y}-\mathbf{X}\hat{\beta}) = 0 \quad \text{or} \quad \mathbf{X}^T\mathbf{X}\hat{\beta} = \mathbf{X}^T\mathbf{y}$$

Using this we can write

$$\begin{aligned}\langle \mathbf{y}-\alpha\mathbf{X}\hat{\beta}, \mathbf{y}-\alpha\mathbf{X}\hat{\beta}\rangle &= \mathbf{y}^T\mathbf{y} - 2\alpha\mathbf{y}^T\mathbf{X}\hat{\beta} + \alpha^2\mathbf{y}^T\mathbf{X}\hat{\beta} \\ &= \mathbf{y}^T\mathbf{y} + \alpha(\alpha-2)\mathbf{y}^T\mathbf{X}\hat{\beta}\end{aligned} \tag{34}$$

If $\alpha = 1$ the left-hand-side is the RSS. This means that

$$RSS = \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{X}\hat{\beta}$$

So

$$\mathbf{y}^T\mathbf{X}\hat{\beta} = \mathbf{y}^T\mathbf{y} - RSS$$

Using this we have that

$$\begin{aligned}\langle \mathbf{y}-\alpha\mathbf{X}\hat{\beta}, \mathbf{y}-\alpha\mathbf{X}\hat{\beta}\rangle &= \mathbf{y}^T\mathbf{y} + \alpha(\alpha-2)\mathbf{y}^T\mathbf{X}\hat{\beta} \\ &= \mathbf{y}^T\mathbf{y} + \alpha(\alpha-2)(\mathbf{y}^T\mathbf{y} - RSS) \\ &= (1-\alpha)^2\mathbf{y}^T\mathbf{y} + \alpha(2-\alpha)RSS\end{aligned} \tag{35}$$

As $\mathbf{y}$ has a mean zero and a standard deviation of one means that $\frac{1}{N}\mathbf{y}^T\mathbf{y} = 1$ so the above becomes

$$\frac{1}{N}\langle \mathbf{y}-\alpha\mathbf{X}\hat{\beta}, \mathbf{y}-\alpha\mathbf{X}\hat{\beta}\rangle = (1-\alpha)^2 + \frac{\alpha(\alpha-2)}{N}RSS$$

Putting this expression into the above gives the desired expression.

□

References

[Boz87] Hamparsum Bozdogan, *Model selection and akaike's information criterion (aic): The general theory and its analytical extensions*, Psychometrika **52** (1987), no. 3, 345–370.

[EHJT04] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani, *Least angle regression*, The Annals of statistics **32** (2004), no. 2, 407–499.

[LH95] Charles L Lawson and Richard J Hanson, *Solving least squares problems*, SIAM, 1995.

[Sch78] Gideon Schwarz, *Estimating the dimension of a model*, The annals of statistics (1978), 461–464.

[Ste81] Charles M Stein, *Estimation of the mean of a multivariate normal distribution*, The annals of Statistics (1981), 1135–1151.

[Tib96] Robert Tibshirani, *Regression shrinkage and selection via the lasso*, Journal of the Royal Statistical Society: Series B (Methodological) **58** (1996), no. 1, 267–288.

[ZH05] Hui Zou and Trevor Hastie, *Regularization and variable selection via the elastic net*, Journal of the royal statistical society: series B (statistical methodology) **67** (2005), no. 2, 301–320.

[ZHT07] Hui Zou, Trevor Hastie, and Robert Tibshirani, *On the "degrees of freedom" of the lasso*, The Annals of Statistics **35** (2007), no. 5, 2173–2192.

Supplementary – R code

```
library(lars)
library(ggplot2)
mydiag = function(a){
  if(length(a) == 1){
    return(matrix(a, 1, 1))
  }else{
    return(diag(a))
  }
}
aic_bic <- function(sum.fit, fit.name, n, plot_it = TRUE){
  p <- sum.fit$Df
  l <- nrow(sum.fit)
  aic <- n * log(sum.fit$Rss/n) + 2 * p
  bic <- n * log(sum.fit$Rss/n) + log(n) * p
  sum.fit <- cbind(sum.fit, aic , bic)
  best_idx <- t(as.matrix(apply(sum.fit[, 3:5], 2, which.min)))
  best_step <- t(as.matrix(sum.fit$Df[best_idx]))
  colnames(best_step) <- c("Cp", "aic", "bic")
  colnames(best_idx) <- c("Cp", "aic", "bic")
  if(plot_it){
    for (i in colnames(best_step)){
      png(file = paste(i, "_", fit.name, ".png", sep = ""),
          width = 3.25, height = 3.25, units = "in", res = 500)
      plot(4:(l-1), sum.fit[-(1:4), i], type = 'b', xlab = 'step', ylab = i,
           main = fit.name, ylim = c(min(sum.fit[-(1:4), i])-5, max(sum.fit[-(1:4), i]) + 5))
      points((best_idx[, i]-1), sum.fit[best_idx[, i], i], col = "red")
      dev.off()
    }
  }
```

```
  return(list("sum.fit" = sum.fit, "opt.model" = best_step, "idx" = best_idx))
}
my_LH_algo = function(X, Y, GE, t_now, rescale, gamma_OLS){
  # minimize ||Y - X\beta||^2 where G\beta \geq tnow
  converge = FALSE
  p = ncol(X)
  Eset = sum((GE == 1) * 2^(0:(p - 1)))
  GE_new = GE %*% rescale

  while(!converge){
    svd_GE = svd(GE_new)
    gamma_now = gamma_OLS +
      svd_GE$v %*% mydiag(1/svd_GE$d) %*% t(svd_GE$u) %*% (t_now - GE_new %*% gamma_OLS)
    beta_now = rescale %*% gamma_now

    if(sum(abs(beta_now)) > t_now * (1 + 1e-8)){
      delta_now = matrix(sign(beta_now), nrow = 1)
      index_i = sum((delta_now == 1) * 2^(0:(p - 1)))
      if(!(index_i %in% Eset)){
        Eset = c(Eset, index_i)
        GE_new = rbind(GE_new, delta_now %*% rescale)
        GE = rbind(GE, delta_now)
      }else{break}
    }else{
      converge = TRUE
    }

  }
  return(list(beta = beta_now, GE = GE, converge = converge))
}
aic_bic_mylasso_object = function(myfit, fit.name, main){
  p = max(myfit$RSS$Df)
  range = 5:(p + 1)
  AIC = myfit$n * log(myfit$RSS$Rss/myfit$n) + 2 * myfit$RSS$Df
  png(file = paste("aic_", fit.name, ".png", sep = ""),
      width = 3.25, height = 3.25, units = "in", res = 500)
  plot(myfit$RSS$Df[range], AIC[range], main = main, xlab = "df", ylab = "AIC", type = "b")
  points(myfit$RSS$Df[which.min(AIC)], AIC[which.min(AIC)], col = "red")
  dev.off()

  png(file = paste("bic_", fit.name, ".png", sep = ""),
      width = 3.25, height = 3.25, units = "in", res = 500)
  BIC = myfit$n * log(myfit$RSS$Rss/myfit$n) + log(myfit$n) * myfit$RSS$Df
  plot(myfit$RSS$Df[range], BIC[range], main = main, xlab = "df", ylab = "BIC", type = "b")
  points(myfit$RSS$Df[which.min(BIC)], BIC[which.min(BIC)], col = "red")
  dev.off()

  png(file = paste("Cp_", fit.name, ".png", sep = ""),
      width = 3.25, height = 3.25, units = "in", res = 500)
  CP = myfit$RSS$Rss/myfit$RSS$Rss[p + 1]*myfit$n - myfit$n + 2 * (myfit$RSS$Df + 1)
  plot(myfit$RSS$Df[range], CP[range], main = main, xlab = "df", ylab = "CP", type = "b")
  points(myfit$RSS$Df[which.min(CP)], CP[which.min(CP)], col = "red")
```

```
  dev.off()
}
mylasso = function(X, Y, num_t){
  n = nrow(X)
  p = ncol(X)
  beta_OLS = summary(lm(Y ~ -1 + X))$coef[, 1]
  beta_threshold = 1e-8
  L1norm = pracma::linspace(sum(abs(beta_OLS)), beta_threshold, num_t) * (1 + 1e-10)

  # solve for beta under each t
  eigen_Sigma = eigen(cov(X))
  rescale = eigen_Sigma$vectors %*% diag(1/sqrt(eigen_Sigma[[1]])) %*% t(eigen_Sigma$vectors)
  Xnew = X %*% rescale
  Xnew_XTX_inv = diag(rep(1/(n - 1), p)) #solve(crossprod(Xnew))
  gamma_OLS = summary(lm(Y ~ -1 + Xnew))$coef[, 1]
  beta_mat = matrix(0, p, num_t)
  sigma_sq = rep(0, num_t)
  GE = matrix(sign(beta_OLS), nrow = 1)

  for(i in 1:num_t){
    t_now = L1norm[i]

    # optimize for beta_hat
    LH_result = my_LH_algo(X, Y, GE, t_now, rescale, gamma_OLS)
    beta_mat[, i] = LH_result$beta * (abs(LH_result$beta) > beta_threshold)
    sigma_sq[i] = sum((Y - X%*%beta_mat[, i])^2)/n

    if(!LH_result$converge){
      beta_mat[, i] = 0
      break
    }
  }
  if(i < num_t){sigma_sq[i:num_t] = sigma_sq[i - 1]}

  pk = colSums(beta_mat!=0)
  RSS = data.frame(Df = 0:10, Rss = c(sum(Y^2), rep(NA, p)))
  beta_pick = data.frame(matrix(0, p + 1, p + 1))
  colnames(beta_pick)[1] = "Df"
  colnames(beta_pick)[2:(p + 1)] = colnames(X)
  beta_pick[, 1] = 0:10
  for(i in 1:p){
    RSS$Rss[i + 1] = min(sigma_sq[pk == i]) * n
    beta_pick[i + 1, 2:(p+1)] = beta_mat[, which(sigma_sq == min(sigma_sq[pk == i]))]
  }
  RSS$AIC = n * log(RSS$Rss/n) + 2 * RSS$Df
  RSS$BIC = n * log(RSS$Rss/n) + log(n) * RSS$Df
  RSS$CP = RSS$Rss/RSS$Rss[p + 1]*n - n + 2 * (RSS$Df + 1)

  best_model = list(AIC = NULL, BIC = NULL, CP = NULL)
  best_model$AIC = list(MSE = RSS$Rss[which.min(RSS$AIC)]/n,
                        model = beta_pick[which.min(RSS$AIC), 2:(p + 1)])
  best_model$BIC = list(MSE = RSS$Rss[which.min(RSS$BIC)]/n,
```

```
                        model = beta_pick[which.min(RSS$BIC), 2:(p + 1)])
  best_model$CP = list(MSE = RSS$Rss[which.min(RSS$CP)]/n,
                       model = beta_pick[which.min(RSS$CP), 2:(p + 1)])
  all_mse = matrix(c(best_model$AIC$MSE, best_model$BIC$MSE, best_model$CP$MSE), 1)
  colnames(all_mse) = c("aic", "bic", "cp")
  criteria = c("AIC", "BIC", "CP")[all_mse == min(all_mse)]
  model = best_model[[which(all_mse == min(all_mse))[1]]]$model
  mse = best_model[[which(all_mse == min(all_mse))[1]]]$MSE

  beta_raw = data.frame(cbind(L1norm, pk, t(beta_mat)))
  colnames(beta_raw) = c("L1norm", colnames(beta_pick))
  result = list(mse = mse, model = model, criteria = criteria, mse.all = all_mse,
                best_model = best_model, RSS = RSS, beta = beta_pick, beta_raw = beta_raw, n = n)
  return(result)
}
myada_lasso = function(X, Y, num_t){
  beta_OLS = summary(lm(Y ~ -1 + X))$coef[, 1]
  X_ada = scale(X, scale = 1/abs(beta_OLS))
  return(mylasso(X_ada, Y, num_t = num_t))
}
plot_mylasso_object = function(myfit, xvar = "beta", main){
  if(xvar == "L1norm"){
    ncol = ncol(myfit$beta_raw)
    plot(myfit$beta_raw$L1norm, myfit$beta_raw[, 3], type = 'l',
         ylim = c(min(myfit$beta_raw[, 3:ncol]), max(myfit$beta_raw[, 3:ncol])),
         col = 1, xlab = "L1norm", ylab = "Coefficients", main = main)
    for(j in 4:ncol){
      lines(myfit$beta_raw$L1norm, myfit$beta_raw[, j], col = j - 2)
    }
  }
  if(xvar == "beta"){
    ncol = ncol(myfit$beta)
    beta_pick = myfit$beta[, 2:ncol]
    x_axis = apply(beta_pick, 1, function(X) sum(abs(X)))
    x_axis = x_axis/max(x_axis)
    plot(x_axis, beta_pick[, 1], type = 'l', ylim = c(min(beta_pick), max(beta_pick)), col = 1,
         xlab = "|beta|/max|beta|", ylab = "Standardized Coefficients", main = main)
    for(j in 2:10){
      lines(x_axis, beta_pick[, j], col = j)
      # abline(v = x_axis[j])
    }
  }
}
min.mse <- function(fit, sum.fit){
  pred <- predict(fit, X, s=c(sum.fit$opt.model), type = 'fit')$fit
  mse <- t(as.matrix(colMeans((Y - pred)^2)))
  tag <- c("Cp", "aic", "bic")
  colnames(mse) <- tag
  criteria <- tag[which.min(mse)]
  return(list("pred" = pred, "mse" = min(mse), "model" =  coef(fit)[sum.fit$idx[, criteria],],
              "criteria" = criteria, "mse.all" = mse))
}
```

```r
final_project_models = function(X, Y, num_t = 2e3){
  # initialization
  n = length(Y)
  p = ncol(X)

  fit.lasso <- lars(X, Y, intercept = F, normalize = F)
  fit.lar <- lars(X, Y, type="lar", intercept = F, normalize = F)
  fit.for <- lars(X, Y, type="for", intercept = F, normalize = F)

  sum.fit.lasso <- aic_bic(summary(fit.lasso), "lasso", n = n, plot_it = FALSE)
  sum.fit.lar <- aic_bic(summary(fit.lar), "lar", n = n, plot_it = FALSE)
  sum.fit.for <- aic_bic(summary(fit.for), "stagwise", n = n, plot_it = FALSE)

  min.lasso <- min.mse(fit.lasso, sum.fit.lasso)
  min.lar <- min.mse(fit.lar, sum.fit.lar)
  min.for <- min.mse(fit.for, sum.fit.for)

  min.our_lasso = mylasso(X, Y, num_t = num_t)
  min.ada_lasso = myada_lasso(X, Y, num_t = num_t)

  png(file = "coeff_plot_LASSO.png", width = 3.25, height = 3.25, units = "in", res = 500)
  plot(fit.lasso)
  dev.off()
  png(file = "coeff_plot_lar.png", width = 3.25, height = 3.25, units = "in", res = 500)
  plot(fit.lar)
  dev.off()
  png(file = "coeff_plot_for.png", width = 3.25, height = 3.25, units = "in", res = 500)
  plot(fit.for)
  dev.off()
  png(file = "coeff_plot_our_LASSO.png", width = 3.25, height = 3.25, units = "in", res = 500)
  plot_mylasso_object(min.our_lasso, xvar = "beta", main = "Our LASSO")
  dev.off()
  png(file = "coeff_plot_ada_LASSO.png", width = 3.25, height = 3.25, units = "in", res = 500)
  plot_mylasso_object(min.ada_lasso, xvar = "beta", main = "Our adaptive LASSO")
  dev.off()

  sum.fit.lasso <- aic_bic(summary(fit.lasso), "lasso", n = n)
  sum.fit.lar <- aic_bic(summary(fit.lar), "lar", n = n)
  sum.fit.for <- aic_bic(summary(fit.for), "stagwise", n = n)
  aic_bic_mylasso_object(min.our_lasso, fit.name = "our_lasso", main = "Our lasso")
  aic_bic_mylasso_object(min.ada_lasso, fit.name = "our_ada_lasso", main = "Our adaptive lasso")

  result = list(min.our_lasso = min.our_lasso, min.ada_lasso = min.ada_lasso,
                min.lasso = min.lasso, min.lar = min.lar, min.for = min.for)
  return(result)
}

## main code starts here
diabetes = read.csv("diabetes_data.csv", header = TRUE)
p = ncol(diabetes) - 1
X = scale(diabetes[, 1:p])
Y = scale(diabetes[, p + 1], scale = FALSE)
```

```r
# setwd("/Users/eric/Desktop/UCD/STA232B/final_project")
result = final_project_models(X, Y)
print(paste0("Figures are saved in ", getwd(), sep = ""))

NOT_RUN = TRUE
if(NOT_RUN){
  ## model 1: lasso from lars
  result$min.lasso$mse
  result$min.lasso$model
  result$min.lasso$criteria
  result$min.lasso$mse.all

  ## model 2: lar from lars
  result$min.lar$mse
  result$min.lar$model
  result$min.lar$criteria
  result$min.lar$mse.all

  ## model 3: forward stagewise from lars
  result$min.for$mse
  result$min.for$model
  result$min.for$criteria
  result$min.for$mse.all

  ## model 4: our lasso
  result$min.our_lasso$mse
  result$min.our_lasso$model
  result$min.our_lasso$criteria
  result$min.our_lasso$mse.all

  ## model 5: our adaptive lasso
  result$min.ada_lasso$mse
  result$min.ada_lasso$model
  result$min.ada_lasso$criteria
  result$min.ada_lasso$mse.all
}
```