Least Angle Regression

STA 232B Final Project

Zhaoyang Shi, Doudou Zhou, Yi Han, Yidong Zhou, Wancheng Cai

Department of Statistics, UC Davis

March 18, 2020

1 Summary

1.1 Introduction

Least Angle Regression was first proposed by Efron, *et al*[1] which is a method that is closely related with Lasso and Forward Stagewise linear regression, but is much computationally simpler. It is also correlated with the Forward Stepwise regression which is a method of model selection.

Lasso is a constrained version of ordinary least squares. Let X be the standardized and centralized design matrix with dimension $n \times m$. y is the centralized response vector with dimension $n \times 1$. A candidate vector of regression coefficients $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_m)'$. Prediction value is

$$\hat{\mu} = \sum_{j=1}^{m} x_j \hat{\beta}_j = X\hat{\beta}, \tag{1}$$

total square error is $S(\hat{\beta}) = ||y - \hat{\mu}||$. So the Lasso choose $\hat{\beta}$ by minimizing $S(\hat{\beta})$ subject to $T(\hat{\beta}) = \sum_{j=1}^{m} |\hat{\beta}_j| \leq t$.

Forward Stagewise Linear Regression is an iterative technique that begin with $\hat{\mu} = 0$ and builds up the regression function in successive small steps. If $\hat{\mu}$ is the current Stagewise estimate,

$$\hat{c} = c(\hat{\mu}) = X'(y - \hat{\mu}), \tag{2}$$

is the vector of current correlations. The second step of Stagewise is taken the taken the direction of the greatest current correlation where

$$\hat{j} = argmax|\hat{c}_j| \quad and \quad \hat{\mu} \to \hat{\mu} + \epsilon \cdot sign(\hat{c}_{\hat{j}})x_{\hat{j}}, \tag{3}$$

with ϵ be small constant. If ϵ is larger enough to be $|\hat{c}_{\hat{j}}|$, then it becomes Stepwise Algorithm.

Efron's paper focus more on the relationship between LARS, Lasso and stagewise methods because forward selection is an agressive fitting technique that can be overly greedy.

1.2 The LARS Algorithm

Least Angle Regression (LARS) is a modified Stagewise procedure that only m steps are needed for the full set of solutions. Assume that the covariate vectors $x_1, x_2, \ldots, x_m$ are linearly independent. $\mathcal{A}$ is a subset of the indices $1, 2, \ldots, m$. Define the matrix $X_{\mathcal{A}} = (\ldots s_j x_j \ldots)_{j \in \mathcal{A}}$, where the signs $s_j = \pm 1$. Let

$$\mathcal{G}_{\mathcal{A}} = X'_{\mathcal{A}} X_{\mathcal{A}} \quad and \quad A_{\mathcal{A}} = (1'_{\mathcal{A}} \mathcal{G}_{\mathcal{A}} 1_{\mathcal{A}})^{-\frac{1}{2}}, \tag{4}$$

where $1_{\mathcal{A}}$ is a vector of 1's pf length equaling $\mathcal{A}$.

Then LARS is to begin at $\hat{\mu}_0 = 0$ and build $\hat{\mu}$ in the following steps. Suppose that $\hat{\mu}_{\mathcal{A}} = X'_{\mathcal{A}}\hat{\beta}_{\mathcal{A}}$ is the current LARS estimate and that

$$\hat{c} = X'(y - \hat{\mu}_{\mathcal{A}}), \tag{5}$$

is the vector of current correlations. Outside the active set, find the covariate x_j that is most correlated with the current residuals which means find the largest $\hat{c}_j$.Then add j to the active set. Now all the covariates in $\mathcal{A}$ have the same absolute correlations with the residuals $y - \hat{\mu}_{\mathcal{A}}$.

Then move $\hat{\mu}_{\mathcal{A}} = X'_{\mathcal{A}}\hat{\beta}_{\mathcal{A}}$ in the direction of the equiangular vector: $u_{\mathcal{A}} = X_{\mathcal{A}}\omega_{\mathcal{A}}$ where $\omega_{\mathcal{A}} = A_{\mathcal{A}}\mathcal{G}_{\mathcal{A}}^{-1}1_{\mathcal{A}}$.(The prove of why this is the euqiangular vector is proved later) until some covariate $x_l, l \in \mathcal{A}^c$ has as much correlation as the current residual. Then

$$\mu(\gamma) = \hat{\mu}_{\mathcal{A}} + \gamma u_{\mathcal{A}}. \tag{6}$$

Then continuing in this way until all the m covariates have been introduced in the model, which means at the last step we will get the OLS solution. Lars is computational thrifty because it only takes m steps.

1.3 Relationship between LARS, Lasso and Stagewise

LARS algorithm can produce Lasso or Stagewise estimates after simple modifications[**?**]. The author put forward two theorems which state that under the two specific modifications, the LARS algorithm yields all Lasso solutions and Stagewise solutions, respectively. All three algorithms can be viewed as moderately greedy forward stepwise procedures whose forward progress is determined by compromise among the currently most correlated covariates, where LARS is the most computational efficient one. Basically, their main difference lies on the restriction of the sign of beta and current correlation. Stagewise requires successive differences of $\hat{\beta}_j$ agreeing in sign with the current correlation $\hat{c}_j = \boldsymbol{x'_j(y - \hat{\mu})}$, and Lasso needs $\hat{\beta}_j$ to agree in sign with $\hat{c}_j$, while LARS puts no sign restrictions.

1.4 Degrees of Freedom and C_p Estimates

LARS can provide all possible estimates of the vector of β, but people often want just a single $\hat{\beta}$. In the section, the author present a C_p-type selection criterion:

$$C_p(\hat{\mu}) = \frac{||y - \hat{\mu}||^2}{\sigma^2} - n + 2df_{\mu,\sigma^2}, \tag{7}$$

which can be applied to LASR, Lasso and Stagewise. df_{μ,σ^2} is more difficult to estimate in $C_p(\hat{\mu})$. A bootstrap estimate of df_{μ,σ^2} is discussed. The author also prove that under the positive cone condition, $df(\hat{\mu}_k) = k$ for LARS. So the risk of a $k-$step LARS estimator $\hat{\mu}_k$ can be estimated by

$$C_p(\hat{\mu}) = \frac{||y - \hat{\mu}||^2}{\sigma^2} - n + 2k. \tag{8}$$

But the formula $df(\hat{\mu}_k) = k$ cannot hold for the Lasso, since the degrees of freedom is m for the full model but the total number of steps taken can exceed m. However, it is found empirically that an intuitively plausible result holds: the degree of freedom is well approximated by the number of non-zero predictors in the model.

1.5 Properties of LARS, Lasso and Stagewise

With the modification of LARS-LASSO, we need to put some constraints on the active set of LARS since it should be consistent with the properties of LASSO.

The first three lemmas show some properties about LARS and will be used to derive the properties of LARS-LASSO.

Lemma (7) shows within $\mathcal{T}$, the increment of β_A will be on the line of $w_{\mathcal{A}}$ and recall the equiangular vector $u_{\mathcal{A}} = X_{\mathcal{A}} w_{\mathcal{A}}$ and it will be shown that $\mathcal{A}$ will give the same direction as that in the equiangular vector.

Lemma (8) gives a constraint on the active set that the sign of correlation in the active set should be consistent with the sign of $\beta_{\mathcal{A}}$: $\hat{c_j} = \hat{C} \cdot sign(\beta_j)$.

Lemma (9) and Lemma (10) explore the property of LASSO curve (T, S) since it should always decline at the fastest rate when it wants to reach the optimal. Therefore, the two lemmas show two properties that on the active set, the gradient of LASSO curve is bounded by $2\hat{C}$ and it self should have some maximal properties.

1.6 Constraints on LARS-LASSO's Choice of $\mathcal{A}$

According to the last section, the choice of the active set $\mathcal{A}$ should satisfy the following constraints. Define $\mathcal{A}_1 := \{j : \ \hat{\beta}_j \neq 0\}$, $\mathcal{A}_0 := \{j : \ \hat{\beta}_j = 0, \ |c_j| = \hat{C}\}$, $\mathcal{A}_{10} = \mathcal{A}_0 \bigcup \mathcal{A}_1$ and $\mathcal{A}_2 = \mathcal{A}_{10}^c$.

Constraint I: $\mathcal{A}_1 \subset \mathcal{A}$.

Constraint II: $\mathcal{A} \subset \mathcal{A}_{10}$

Constraint III: $w_{\mathcal{A}} = A_{\mathcal{A}} \mathcal{G}_{\mathcal{A}}^{-1} 1_{\mathcal{A}}$ can not have different sign of c_j for each coordinate.

Constraint VI: $\mathcal{A} = \arg\min\{A_{\mathcal{A}} : \ L(d) \leq A_{\mathcal{A}}\}$.

Similarly, the next part of the part derives some properties of LARS-Stagewise and put three constraints on the active set. Define $v_{\mathcal{A}} := X_{\mathcal{A}} P_{\mathcal{A}}$.

Constraint I: $v \in S_{\mathcal{A}}^+ := \{v : \ v = \sum_{j \in \mathcal{A}} x_j P_j, \ P_j \geq 0, \ \sum_{j \in \mathcal{A}} P_j = 1\}$, which is a simplex.

Constraint II: v must be proportional to $u_{\mathcal{B}}$.

Constraint III: $v = v_{\mathcal{B}}$ must statisfy $x_j^{'} v_{\mathcal{B}} \geq A_{\mathcal{B}}^2$ for $j \in \mathcal{A} - \mathcal{B}$.

2 Explanation for Unclear Places

2.1 Why $\hat{\gamma_k} < \bar{\gamma_k}$ (except the last stage)

Proof: From the definition, we know that $\hat{\gamma_k}$ can be expressed as $\hat{\gamma_k} = min_{j \in A^C}^+ \{\frac{\hat{C_k} - \hat{c_j}}{A_k - a_j}, \frac{\hat{C_k} + \hat{c_j}}{A_k + a_j}\}$, and the $\bar{\gamma}_k$ can be expressed as $\frac{\hat{C_k}}{A_k}$. We observe that from the process of LARS method, $\hat{C_k} = max\{|\hat{c_kj}|\}$, so we know that $|\hat{c_j}| < \hat{C_k}$, which means $\hat{C_k} - \hat{c_j}$ and $\hat{C_k} + \hat{c_j}$ all larger than 0. And $A_k = (1_k'(X_k' X_k)^{-1} 1_k)^{-1/2}$. So $A_k > 0$. Then we can compare them in different situations:

(1) If $A_k > 0 > -a_j$, then $\hat{\gamma_k} = min_{j \in A^C}^+ \{\frac{\hat{C_k} - \hat{c_j}}{A_k - a_j}, \frac{\hat{C_k} + \hat{c_j}}{A_k + a_j}\} = \frac{\hat{C_k} - \hat{c_j}}{A_k - a_j} < \frac{\hat{C_k} + \hat{C_k}}{A_k + A_k} = \frac{\hat{C_k}}{A_k} = \bar{\gamma_k}$

(2) If $A_k > a_j > 0$, then $\hat{\gamma_k} = min_{j \in A^C}^+ \{\frac{\hat{C_k} - \hat{c_j}}{A_k - a_j}, \frac{\hat{C_k} + \hat{c_j}}{A_k + a_j}\} = \frac{\hat{C_k} - \hat{c_j}}{A_k + a_j} < \frac{\hat{C_k} + \hat{C_k}}{A_k + A_k} = \frac{\hat{C_k}}{A_k} = \bar{\gamma_k}$

(3) If $|a_j| < A_k$, then $(\frac{\hat{C_k} - \hat{c_j}}{A_k - a_j} - \frac{\hat{C_k}}{A_k})(\frac{\hat{C_k} + \hat{c_j}}{A_k + a_j} - \frac{\hat{C_k}}{A_k}) = -\frac{[\hat{C_k} a_j - \hat{c_j} A_k]^2}{A_k^2 (A_k^2 - a_j^2)} < 0$, which means $\hat{\gamma_k} < \bar{\gamma_k}$.

2.2 Why $|c_j(\gamma)| = \hat{C} - \gamma A_{\mathcal{A}}$

Proof: First, according to the definition of $\hat{C} = max_j |\hat{c_j}|$, we know $|c_j| = \hat{C}$. And since $X_j = (\dots s_j x_j \dots)$, so $c_j = s_j \hat{C}$ where s_j is the sign of X_j. Also, $X' u_{\mathcal{A}} 1_{\mathcal{A}}$ and $s_j x_j' u_{\mathcal{A}} = A_{\mathcal{A}}$, we get $a_j = x_j' u_{\mathcal{A}} = s_j A_{\mathcal{A}}$. So we can get $|c_j(\gamma)| = |\hat{c_j} - \gamma a_j| = |s_j||\hat{c} - \gamma A_{\mathcal{A}}| = |\hat{c} - \gamma A_{\mathcal{A}}| = \hat{c} - \gamma A_{\mathcal{A}}$. The last equality holds because we prove $\hat{\gamma} < \bar{\gamma}_k$.

2.3 Proof of 3.20

Here we want to prove (3.20) of the paper:

$$R^2(\bar{\beta}_k) - R^2(\hat{\beta}_k) = \frac{1-\rho_k^2}{\rho_k(2-\rho_k)}[R^2(\hat{\beta}_k) - R^2(\hat{\beta}_{k-1})]. \tag{9}$$

In fact, we find that (3.20) is not correct, and it may be an typo of the author. In fact, the correct one is

$$R^2(\bar{\beta}_k) - R^2(\hat{\beta}_k) = \frac{(1-\rho_k)^2}{\rho_k(2-\rho_k)}[R^2(\hat{\beta}_k) - R^2(\hat{\beta}_{k-1})]. \tag{10}$$

Proof: Recall that $R^2(\beta) = 1 - \frac{||y-X\beta||^2}{||y||^2}$ when y is centered. So

$$R^2(\bar{\beta}_k) - R^2(\hat{\beta}_k) = \frac{||y-\hat{\mu}_k||^2 - ||y-\bar{y}_k||^2}{||y||^2}, \tag{11}$$

and

$$R^2(\hat{\beta}_k) - R^2(\hat{\beta}_{k-1}) = \frac{||y-\hat{\mu}_{k-1}||^2 - ||y-\hat{\mu}_k||^2}{||y||^2}. \tag{12}$$

By (2.22) of the paper,we get $\bar{y}_k = \frac{1}{\rho_k}\hat{\mu}_k + (1-\frac{1}{\rho_k})\hat{\mu}_{k-1}$ and $\hat{\mu}_k = \hat{\mu}_{k-1} + \hat{\gamma}_k u_k$.

Hence

$$\bar{y}_k = \frac{\hat{\gamma}_k}{\rho_k}u_k + \hat{u}_{k-1}. \tag{13}$$

By definition we know $X_k'(y-\hat{\mu}_{k-1}) = \hat{C}_k 1_k$ and $u_k = X_k w_k = X_k(X_k'X_k)^{-1}1_k A_k$. So

$$u_k'(y-\hat{\mu}_{k-1}) = w_k'X_k'(y-\hat{\mu}_{k-1}) = \hat{C}_k A_k 1_k'(X_k'X_k)^{-1}1_k = \frac{\hat{C}_k}{A_k} = \bar{\gamma}_k, \tag{14}$$

because $A_k = (1_k'(X_k'X_k)^{-1}1_k)^{-\frac{1}{2}}$. Then we have

$$||y-\bar{y_k}||^2 = ||y-\hat{\mu}_{k-1} - \frac{\hat{\gamma}_k}{\rho_k}u_k|| = ||y-\hat{\mu}_{k-1}||^2 - 2\frac{\hat{\gamma}_k\bar{\gamma}_k}{\rho_k} + \frac{\hat{\gamma}_k^2}{\rho_k^2}, \tag{15}$$

$$||y-\hat{\mu}_k||^2 = ||y-\hat{\mu}_{k-1} - \hat{\gamma}_k u_k|| = ||y-\hat{\mu}_{k-1}||^2 - 2\hat{\gamma}_k\bar{\gamma}_k + \hat{\gamma}_k^2, \tag{16}$$

because $u_k'(y-\hat{\mu}_{k-1}) = \bar{\gamma}_k$ and $||u_k|| = 1$. Then we get

$$||y-\hat{\mu}_k||^2 - ||y-\bar{y_k}||^2 = 2\hat{\gamma}_k\bar{\gamma}_k(\frac{1}{\rho_k}-1) - \hat{\gamma}_k^2(\frac{1}{\rho_k^2}-1) = \hat{\gamma}_k\bar{\gamma}_k\frac{(1-\rho_k)^2}{\rho_k}, \tag{17}$$

because $\rho_k = \frac{\hat{\gamma}_k}{\bar{\gamma}_k}$.

$$||y-\hat{\mu}_{k-1}||^2 - ||y-\hat{\mu}_k||^2 = 2\hat{\gamma}_k\bar{\gamma}_k - \hat{\gamma}_k^2 = \hat{\gamma}_k\bar{\gamma}_k(2-\rho_k), \tag{18}$$

then after some simple calculation we can get (10).

2.4 How to Find Equiangular Vector

The objective is to find an unit equiangular vector u_A lying on the space of X_A. So u_A can be expressed as $u_A = X_A w_A$ which satisfies $X_A'u_A = A_A 1_A$ for some $A_A > 0$ and $||u_A||^2 = 1$.

Because

$$X_A'u_A = X_A'X_A w_A = A_A 1_A, \tag{19}$$

we can get $w_A = A_A(X_A'X_A)^{-1}1_A$. Combined with

$$||u_A||^2 = w_A'X_A'X_A w_A = (A_A)^2 1_A'(X_A'X_A)^{-1}1_A = 1, \tag{20}$$

we have $A_A = (1_A'(X_A'X_A)^{-1}1_A)^{\frac{1}{2}}$, which finishes the proof.

2.5 Optimization over a Simplex

The optimization program (5.18) and (5.19) is to minimize a strictly convex objective function over a convex set. We will show why this is equivalent to (5.20), which is to minimize it without the second condition:

$$sign(\hat{\beta}_j) = s_j \ for \ j \in \mathcal{A}, \tag{21}$$

We note that the original optimization program of LASSO:

$$\min \|y - X_{\mathcal{A}} S_{\mathcal{A}} \beta_{\mathcal{A}}\|^2, \tag{22}$$

subject to

$$\sum_{\mathcal{A}} |\beta_j| \leq t. \tag{23}$$

Introducing a Lagrangian function

$$L(\lambda, \beta_{\mathcal{A}}) := \|y - X_{\mathcal{A}} S_{\mathcal{A}} \beta_{\mathcal{A}}\|^2 + \lambda \|\beta_{\mathcal{A}}\|_1. \tag{24}$$

According to the convex optimization with the subdifferential techniques[2], take derivative with respect to β_A on (24) and note that $\partial|\beta_j| = sign(\beta_j) = s_j$ for $\beta_j \neq 0$ and by LASSO's property:

$$-S_{\mathcal{A}} X_{\mathcal{A}}^{'} (y - X_{\mathcal{A}} S_{\mathcal{A}} \beta_{\mathcal{A}}) + \lambda S_{\mathcal{A}} 1_{\mathcal{A}} = 0, \tag{25}$$

which yields the same equation as we add the second condition (21). In fact, the second condition is to confine the sign of β_j while we are also requiring this condition along the subdifferential of $\beta_{\mathcal{A}}$ as long as it does not reach the boundary 0, in which case the second condition (21) is still satisfied because this is when there is not positive γ for LARS to move forward, which implies $0 = sign(\beta_j) = s_j$.

3 Data Analysis

3.1 Data Description

We studies the example data diabetes in the paper LEAST ANGLE REGRESSION[3]. Table 1 shows a small part of the original data. Ten baseline variables, age, sex, body mass index, average blood pressure and six blood serum measurements, were obtained for each of n = 442 diabetes patients. Only sex is categorical variable. The response of interest is a quantitative measure of disease progression one year after baseline. Our goal is to construct a model that predicted response y from covariates $x_1, x_2, ..., x_{10}$. Moreover, we hope that the model would produce accurate baseline predictions of response for future patients and that the form of the model would suggest covariates were important factors in diseases progression. In practice, we have standardized these covariates to have mean 0 and unit length, and the response has mean 0.

Two models were proposed by the author to achieve the goals, one was just based on the 10 predictors $x_1, x_2, \ldots, x_10$. Another one was to also consider the all the quadratic forms like $x_1 x_2, x_1^2, \ldots, x_{10}^2$ except x_2^2 which represent age of the patients.

$$\sum_{i=1}^{n} y_i = 0, \quad \sum_{i=1}^{n} x_{ij} = 0, \quad \sum_{i=1}^{n} x_{ij}^2 = 1 \quad \text{for } j = 1, 2, \ldots, m \tag{26}$$

Table 1: Part of diabetes data

	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

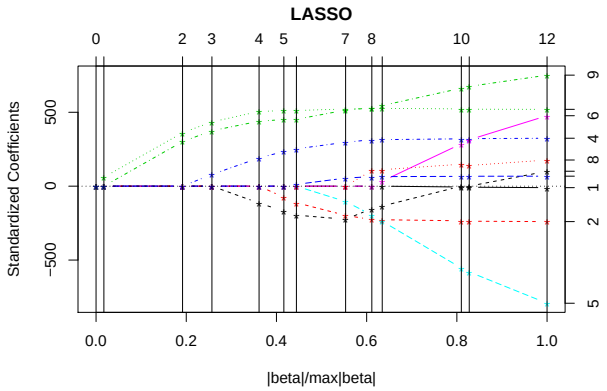
3.1 Results of the data analysis (without interaction)

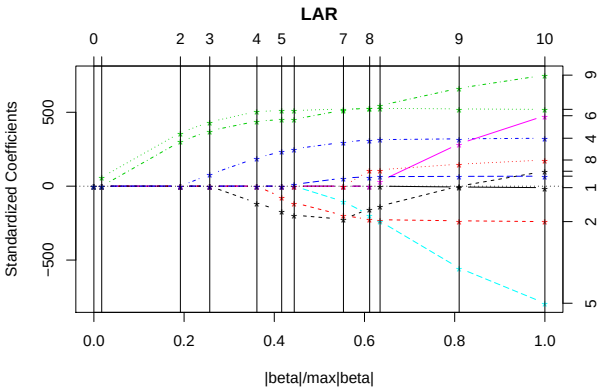
3.1.1 Lasso, Stagewise and LARS

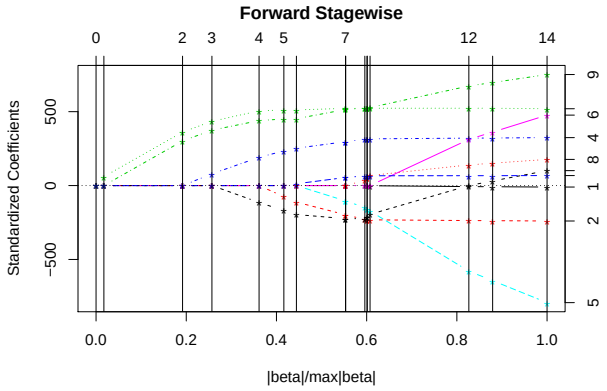
At first, we only consider the main effects $(x_1, x_2, ..., x_{10})$ in the linear model. These three methods are used in the following data analysis, lasso, lar and stagewise. Table 2 is the summary of these three methods and Figure 1 shows the corresponding coefficient plot. The steps of lar is smallest (m=10) compared with lasso (12) and stagewise(14). Although the definition of these three methods are quite different, the results are nearly but not exactly identical. Here are some difference. As for lar, it add one variable in each step. As for lasso, in the step 9 it selects all main effects except for age x_1. And in the step 10 it add the variable age x_1 and delete the variable hdl x_7. The chosen variables are unchanged in the step 11. As for stagewise, the chosen variables remains the same in the step 9, 10, 12 and 14. In the most of steps of lasso and stagewise they add one variable in each step like lar. In the diabetes data, only lasso can delete one variable in a step. And they all ends up with the same results.

Table 2: Results summary

lasso				lar				stagewise			
Df	Rss	Cp	R2	Df	Rss	Cp	R2	Df	Rss	Cp	R2
0	2621009	453.80	0.00	0	2621009	453.80	0.00	0	2621009	453.80	0.00
1	2510465	418.02	0.04	1	2510465	418.02	0.04	1	2510465	418.02	0.04
2	1700369	143.15	0.35	2	1700369	143.15	0.35	2	1700369	143.15	0.35
3	1527165	85.95	0.42	3	1527165	85.95	0.42	3	1527165	85.95	0.42
4	1365734	32.78	0.48	4	1365734	32.78	0.48	4	1365734	32.78	0.48
5	1324118	20.55	0.49	5	1324118	20.55	0.49	5	1324118	20.55	0.49
6	1308932	17.36	0.50	6	1308932	17.36	0.50	6	1308932	17.36	0.50
7	1275355	7.89	0.51	7	1275355	7.89	0.51	7	1275355	7.89	0.51
8	1270233	8.14	0.52	8	1270233	8.14	0.52	6	1275355	5.89	0.51
9	1269390	9.85	0.52	9	1269390	9.85	0.52	6	1271599	4.60	0.51
10	1264977	10.34	0.52	10	1263983	10.00	0.52	7	1271154	6.45	0.52
9	1264765	8.27	0.52					8	1271150	8.45	0.52
10	1263983	10.00	0.52					9	1270685	10.29	0.52
								9	1264371	8.13	0.52
								10	1263983	10.00	0.52





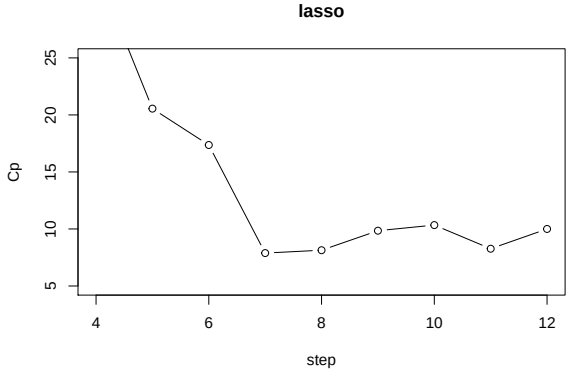


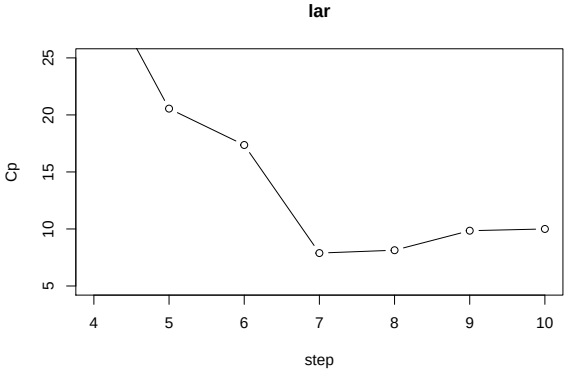
3.1.2 C_p-type selection criterion

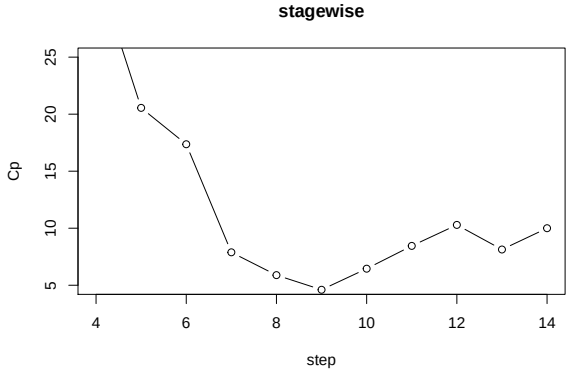
In the paper, it provides the C_p-type selection criterion. We choose the model with the minimum C_p from Table 2. From Table 3, lasso and lar choose the same model. But stagewise add one more variable (tch).

Table 3: C_p-type selection criterion

	step	age	sex	bmi	map	tc	ldl	hdl	tch	ltg	glu
lasso	7	0	-197.76	522.26	297.16	-103.95	0	-223.93	0	514.75	54.77
lar	7	0	-197.76	522.26	297.16	-103.95	0	-223.93	0	514.75	54.77
stagewise	9	0	-229.78	522.26	313.41	-148.46	0	-223.93	34.92	524.22	65.12





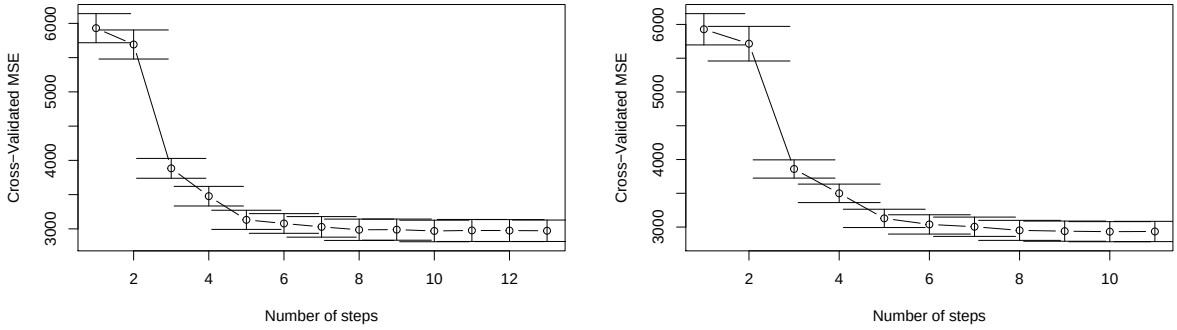


3.1.3 K-fold cross-validated MSPE

In order to choose the final model, we also try the 10-fold cross-validated mean squared prediction error for lars and lasso. Table 4 shows that the results of lar and lasso are same. Moreover, in practice lasso may choose another model by minimum cross-validated MSPE because of the randomness. It could be a problem. Compared with the results in the section C_p-type selection criterion, this time the chosen models are both more complicated than themselves before. Both of them add the variable tch which is only chosen in the type stageiwse by C_p.

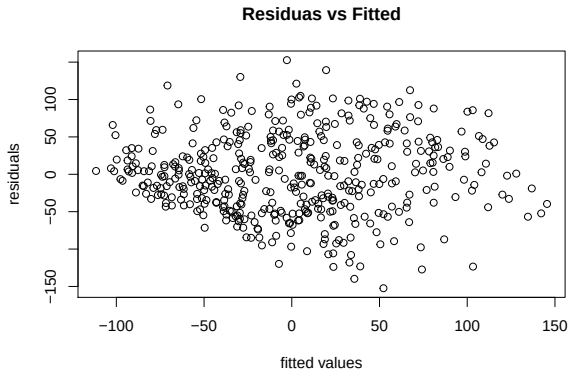
Table 4: K-fold cross-validated MSPE

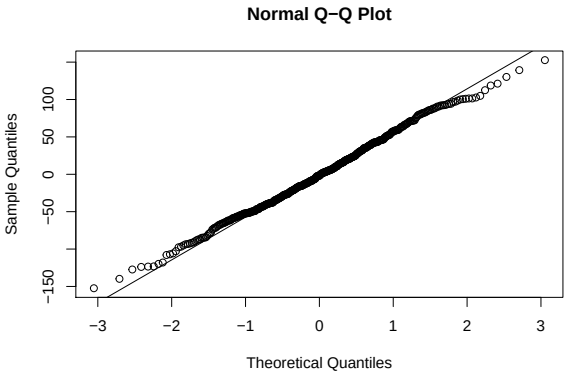
	step	CV	age	sex	bmi	map	tc	ldl	hdl	tch	ltg	glu
lasso	9	2980.24	0	-226.13	526.89	314.39	-195.11	0	-152.48	106.34	529.91	64.49
lar	8	2932.52	0	-226.13	526.89	314.39	-195.11	0	-152.48	106.34	529.92	64.49



3.1.4 Diagnosis of the residuals

We choose the lar result in the 3 according to C_p-type selection criterion as the final model. The following plots show the diagnosis of the final model without interaction. It is clear that there is no relationship between the fitted values and the residuals. From the QQ-plot, the normality of the residuals is reasonable.





3.2 Results of the data analysis (quadratic model)

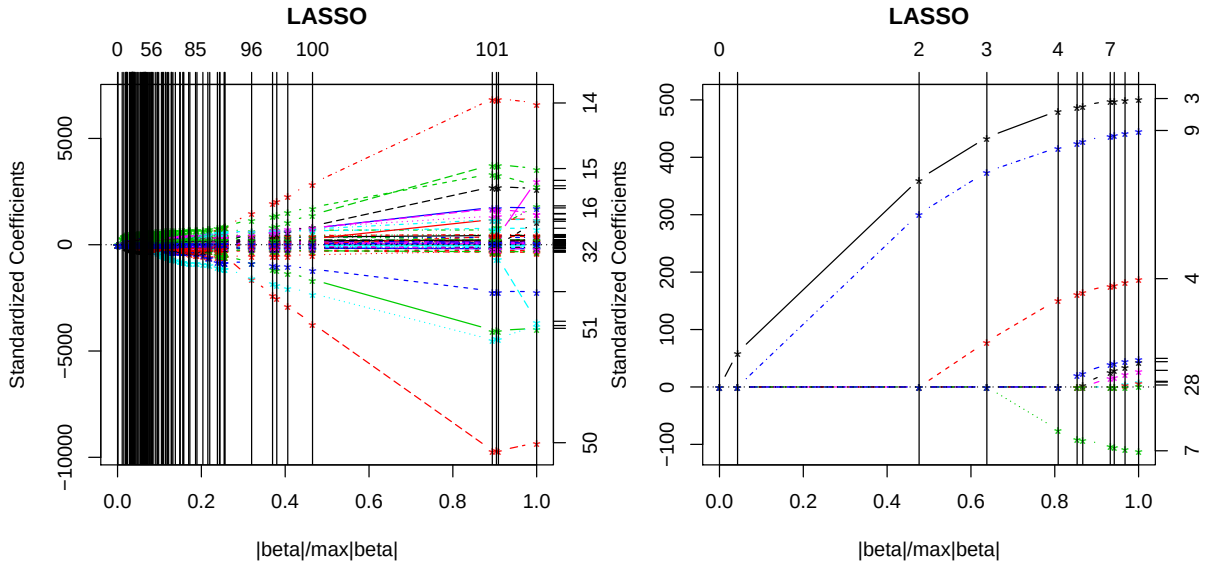
Consider an expansion of the variables by including the quadratic terms of the continuous variables, which has $m = 64$ predictors, including interactions and squares of the 10 original covariates:

$$\text{Quadratic Model} \qquad 10 \text{ main effects}, \ 45 \text{ interactions}, \ 9 \text{ squares},$$

the last being the squares of each x_j except the dichotomous variable x_2.

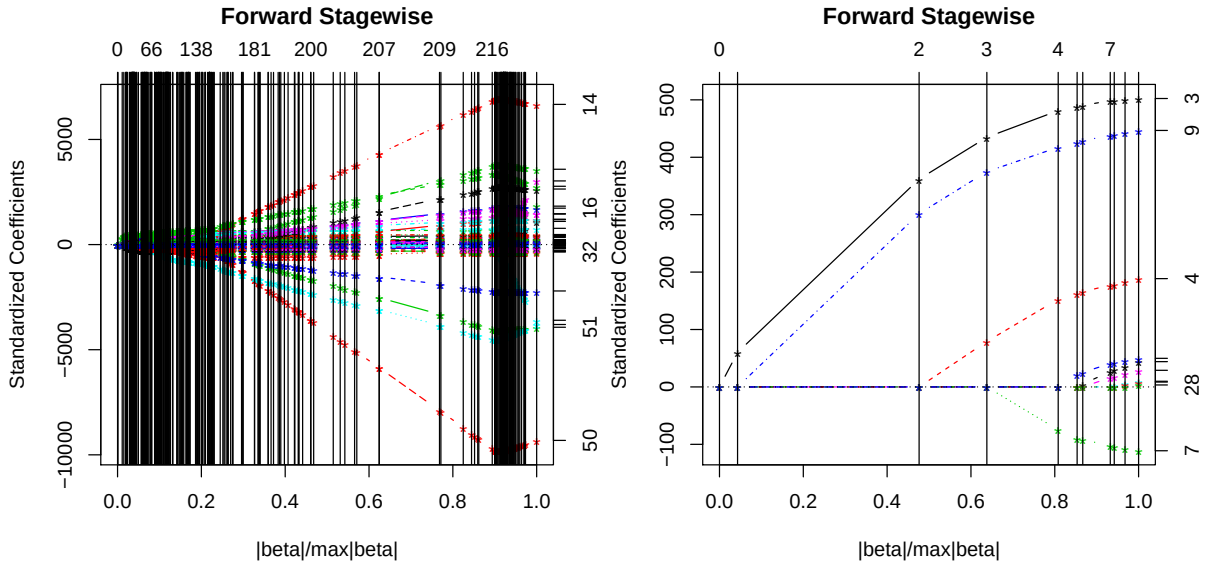
3.2.1 Lasso, Stagewise and LARS

Setting `type` as `'lasso'` yields the Lasso solutions, which are visualized in the following plots. The left panel shows all Lasso solutions $\hat{\beta}(t)$ for the quadratic model against the L1 norm of the coefficient vector, as a fraction of the maximal L1 norm. We see that the Lasso tends to shrink the OLS coefficients toward 0, more so for small values of L1 norm of the coefficient vector. All Lasso solutions shown in one plot is so messy due to the large number of predictors in quadratic model. It's more clear to look at the right panel, which only shows the first 10 steps of Lasso algorithm. The covariates enter the regression equation sequentially in order $j = 3, 9, 4, 7, 37, 20, \cdots$.

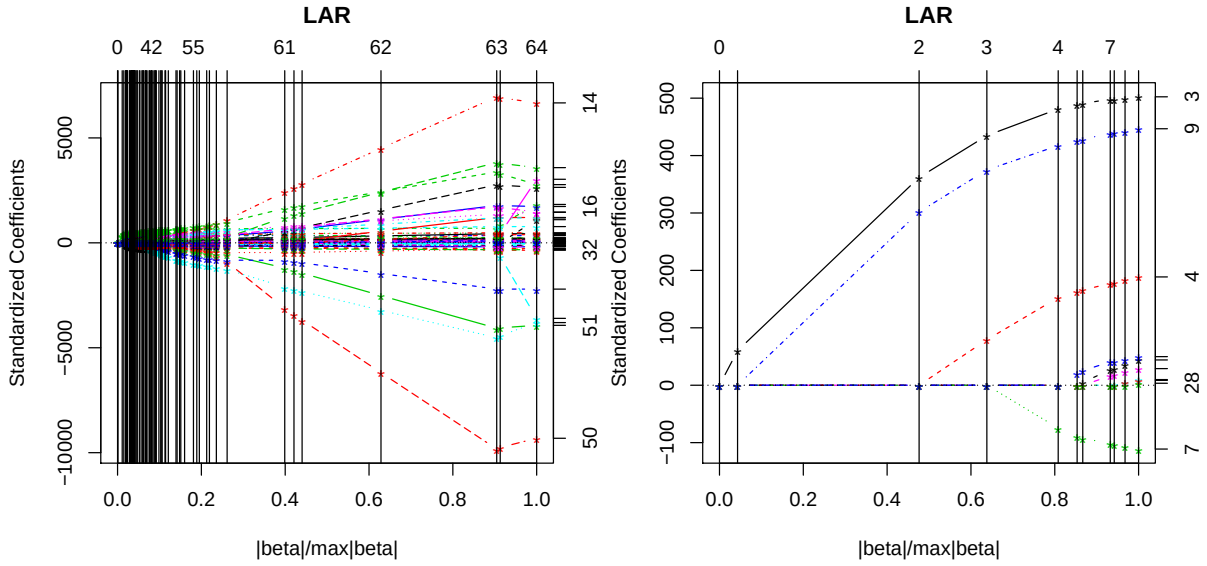


Setting `type` as `'forward.stagewise'` yields the Stagewise solutions, which are visualized in the following plots. The left panel shows the coefficients plot for Stagewise applied to quadratic model. Similarly,

the right panel shows the first 10 steps of Stepwise algorithm. The covariates enter the regression equation sequentially in order $j = 3, 9, 4, 7, 37, 20, \cdots$. The striking fact is the similarity between the Lasso and Stagewise estimates. Although their definitions look completely different, the results are nearly, but not exactly, identical.



Setting `type` as `'lar'` yields the LARS solutions, which are visualized in the following plots. The left panel concerns the LARS analysis of the quadratic model. Similarly, the right panel shows the first 10 steps of LARS algorithm. The complete algorithm required only $m = 64$ steps with the variables joining in the same order as for the Lasso: $j = 3, 9, 4, 7, 37, 20, \cdots$. Tracks of the regression coefficients are nearly but not exactly the same as either the Lasso or Stagewise tracks.



To sum up, Lasso and Stagewise give almost identical results as LARS. However, LARS requires the least steps compared to the other two. The following analysis focuses on LARS algorithm.

3.2.2 Absolute current correlations

The following plot shows the absolute current correlations

$$|\hat{c}_{kj}| = \mathbf{x}_j'(\mathbf{y} - \hat{\boldsymbol{\mu}}_{k-1})|$$

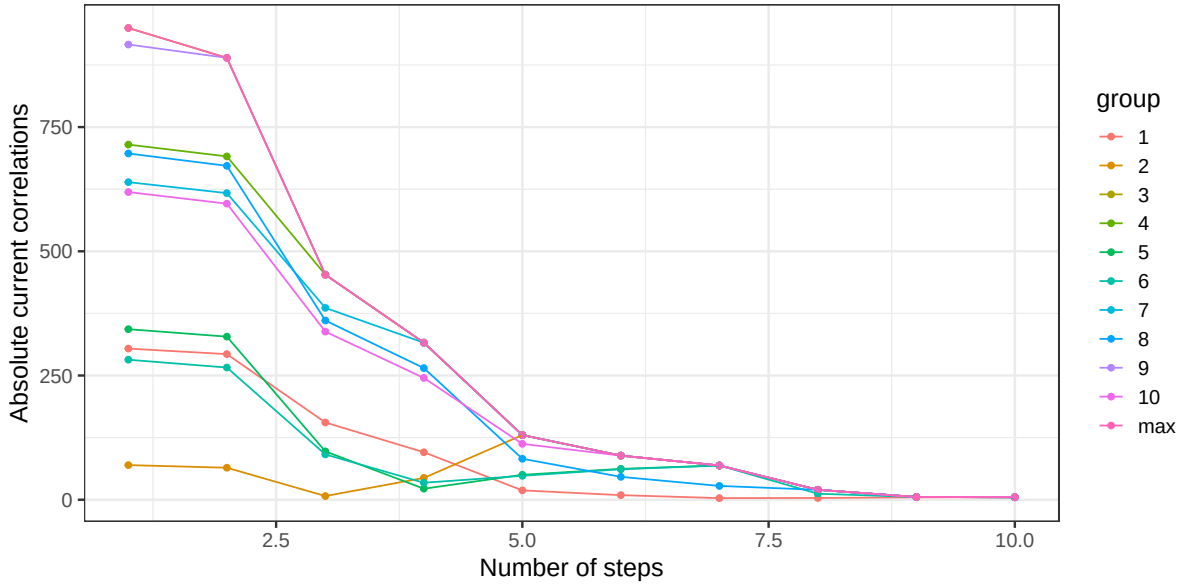
for variables $j = 1, 2, \cdots, 64$ in the quadratic model, as a function of the LARS step k. Variables enter active set in order $3, 9, 4, 7, 37, 20, \cdots, 54, 6$. The maximum correlation

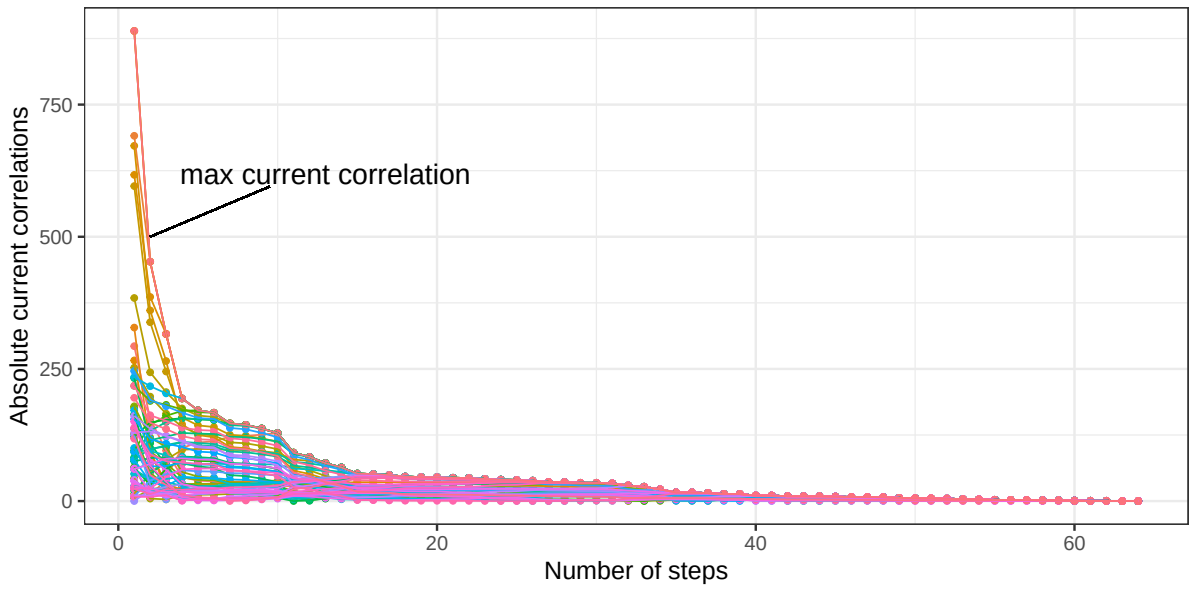
$$\hat{C}_k = \max\{|\hat{c}_{kj}|\} = \hat{C}_{k-1} - \hat{\gamma}_{k-1} A_{k-1}$$

declines with step k, as it must. At each step a new variable j joins the active set, henceforth having $\widehat{C_k} = |\widehat{c}_{kj}|$. Here we compares the plots between two kinds of models. The upper plot is corresponding to the model without interaction and the lower plot is corresponding to the quadratic model.

Table 5: Sequence of the active set for the model without interaction

step	1	2	3	4	5	6	7	8	9	10
name	bmi	ltg	map	hdl	sex	glu	tc	tch	ldl	age
column	3	9	4	7	2	10	5	8	6	1





3.2.3 Simulation study

A small simulation study was carried out comparing the LARS, Lasso, and Stagewise algorithms. The true mean vector $\boldsymbol{\mu}$ for the simulation was $\boldsymbol{\mu} = X\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ was obtained by running LARS for 10 steps on the original $(X, \mathbf{y})$ diabetes data (agreeing in this case with the 10 step Lasso or Stagewise analysis.) Subtracting $\boldsymbol{\mu}$ from a centered version of the original $\mathbf{y}$ vector gave a vector $\boldsymbol{\epsilon} = \mathbf{y} - \boldsymbol{\mu}$ of $n = 442$ residuals. The "true R^2" for this model, $\|\boldsymbol{\mu}\|^2/(\|\boldsymbol{\mu}\|^2 + \|\boldsymbol{\epsilon}\|^2)$, equaled 0.416.

100 simulated response vectors $\mathbf{y}^\star$ were generated from the model

$$\mathbf{y}^\star = \boldsymbol{\mu} + \boldsymbol{\epsilon}^\star,$$

with $\boldsymbol{\epsilon}^\star = (\epsilon_1^\star, \epsilon_2^\star, \cdots, \epsilon_n^\star)$ a random sample, with replacement, from the components of $\boldsymbol{\epsilon}$. The LARS algorithm with $K = 40$ steps was run for each simulated data set $(X, \mathbf{y}\star)$, yielding a sequence of estimates $\hat{\boldsymbol{\mu}}^{(k)\star}, k = 1, 2, \cdots, 40$, and likewise using the Lasso and Stagewise algorithms.

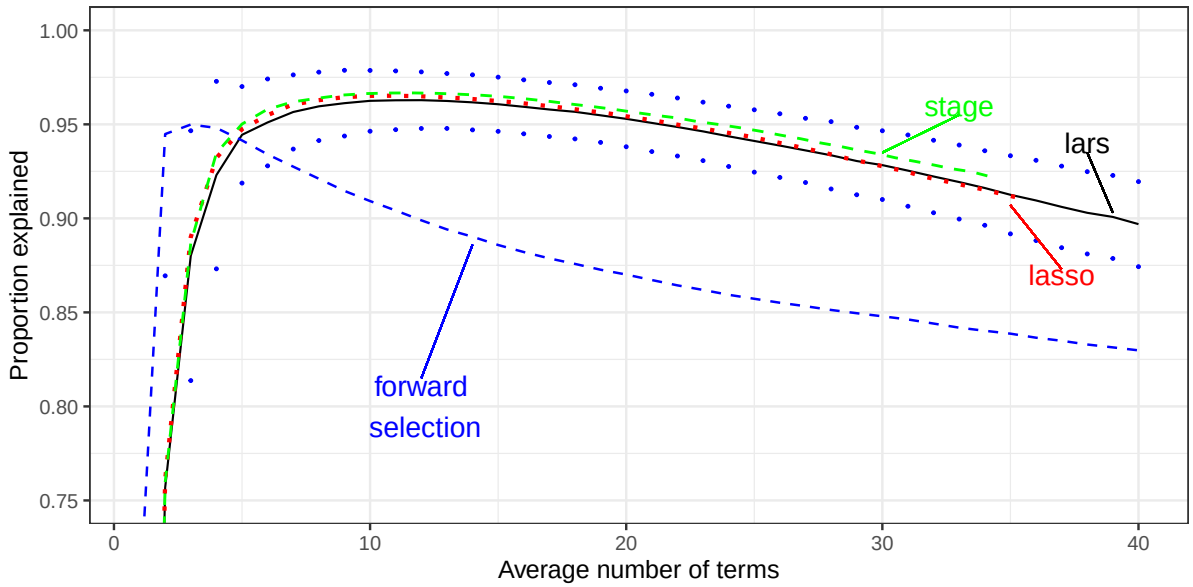
The following plot compares the LARS, Lasso, and Stagewise estimates. For a given estimate $\hat{\boldsymbol{\mu}}$ define the proportion explained $pe(\hat{\boldsymbol{\mu}})$ to be

$$pe(\hat{\boldsymbol{\mu}}) = 1 - \|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2/\|\hat{\boldsymbol{\mu}}\|^2$$

so $pe(\mathbf{0}) = 0$ and $pe(\boldsymbol{\mu}) = 1$. The solid curve graphs the average of $pe(\hat{\boldsymbol{\mu}}^{(k)\star})$ over the 100 simulations, versus step number k for LARS, $k = 1, 2, \cdots, 40$. The corresponding curves are graphed for Lasso and Stagewise, except that the horizontal axis is now the average number of non-zero $\beta_j^\star$ terms composing $\hat{\boldsymbol{\mu}}^{(k)\star}$.

This plot's most striking message is that the three algorithms performed almost identically, and rather well. The average proportion explained rises quickly, reaching a maximum of 0.963 at $k = 10$, and then declines slowly as k grows to 40. The light dots display the small standard deviation of $pe(\hat{\boldsymbol{\mu}}^{(k)\star})$ over the 100 simulations, roughly ± 0.02. Stopping at any point between $k = 5$ and 25 typically gave a $\hat{\boldsymbol{\mu}}^{(k)\star}$ with true predictive R^2 about 0.40, compared to the ideal value 0.416 for $\boldsymbol{\mu}$.

The dashed curve in this plot tracks the average proportion explained by classic Forward Selection. It rises very quickly, to a maximum of 0.950 after $k = 3$ steps, and then falls back more abruptly than the LARS/Lasso/Stagewise curves. This behavior agrees with the characterization of Forward Selection as a dangerously greedy algorithm.



3.2.4 Estimate of degrees of freedom (df)

In the numerical results below, the usual OLS estimates $\bar{\boldsymbol{\mu}}$ and $\bar{\sigma}^2$ from the full OLS model were used to calculate bootstrap estimates of $df_{\boldsymbol{\mu},\sigma^2}$; bootstrap samples $\mathbf{y}^\star$ and replications $\hat{\boldsymbol{\mu}}^\star$ were then generated

according to

$$\mathbf{y}^\star \sim N(\bar{\boldsymbol{\mu}}, \bar{\sigma}^2\mathbf{I}) \text{ and } \hat{\boldsymbol{\mu}}^\star = g(\mathbf{y}^\star).$$

Independently repeat the above procedure say B times gives straightforward estimates for the covariances,

$$\widehat{\text{cov}}_i = \frac{\sum_{b=1}^{B} \hat{\boldsymbol{\mu}}_i^\star(b)[\mathbf{y}_i^\star(b) - \mathbf{y}_i^\star(\cdot)]}{B-1}, \text{ where } \mathbf{y}_i^\star(\cdot) = \frac{\sum_{b=1}^{B} \mathbf{y}_i^\star(b)}{B},$$

and then

$$\widehat{df} = \sum_{i=1}^{n} \widehat{\text{cov}}_i/\bar{\sigma}^2.$$

Normality is not crucial here. Nearly the same results were obtained using $\mathbf{y}^\star = \bar{\boldsymbol{\mu}} + \boldsymbol{\epsilon}^\star$, where the components of $\boldsymbol{\epsilon}^\star$ were resampled from $\boldsymbol{\epsilon}^\star = \mathbf{y} - \bar{\boldsymbol{\mu}}$.

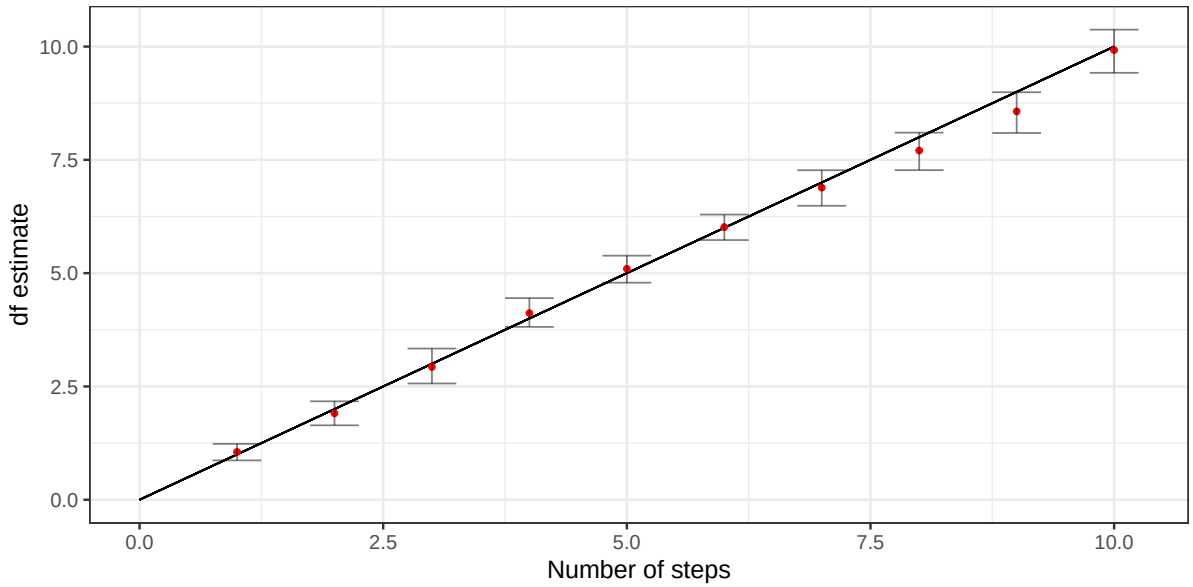
The following plot shows $\widehat{df}_k$ for the diabetes data (the quadratic model) LARS estimates $\hat{\boldsymbol{\mu}}_k, k = 1, 2, \cdots, 64$. It portrays a startlingly simple situation that we will call the "simple approximation",

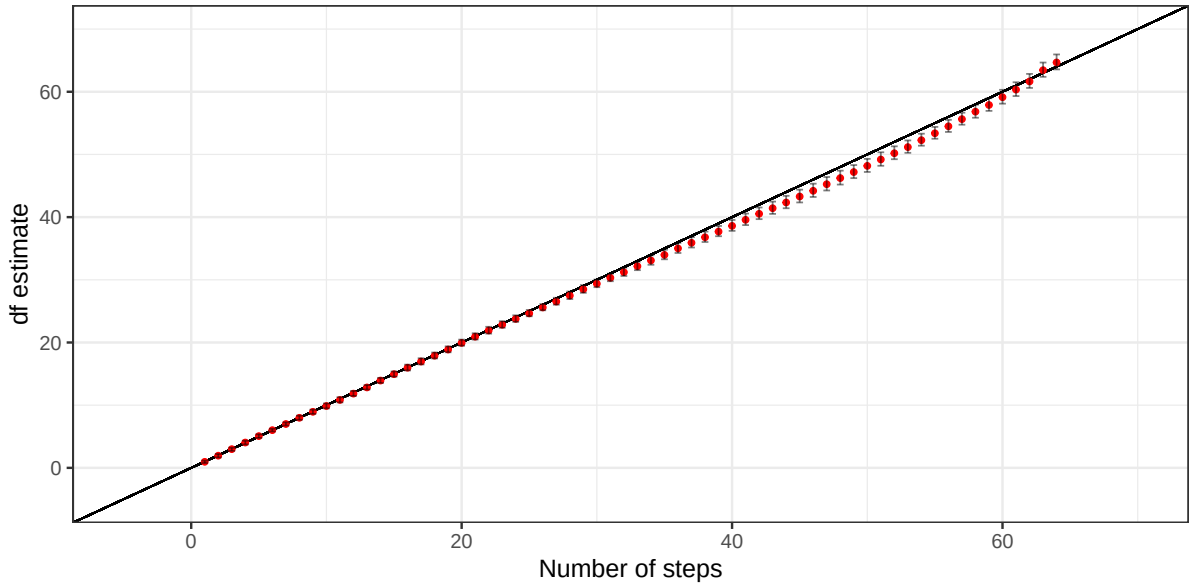
$$df(\hat{\boldsymbol{\mu}}_k) \approx k.$$

Here $B = 500$ replications were divided into 10 groups of 50 each in order to calculate student-t confidence intervals.

$$\widehat{df}_k \pm \frac{t_{n-1}^{1-\alpha/2} s_k}{\sqrt{n}}, \text{ where } n = 10, \alpha = 0.05, k = 1, 2, \cdots, 64$$

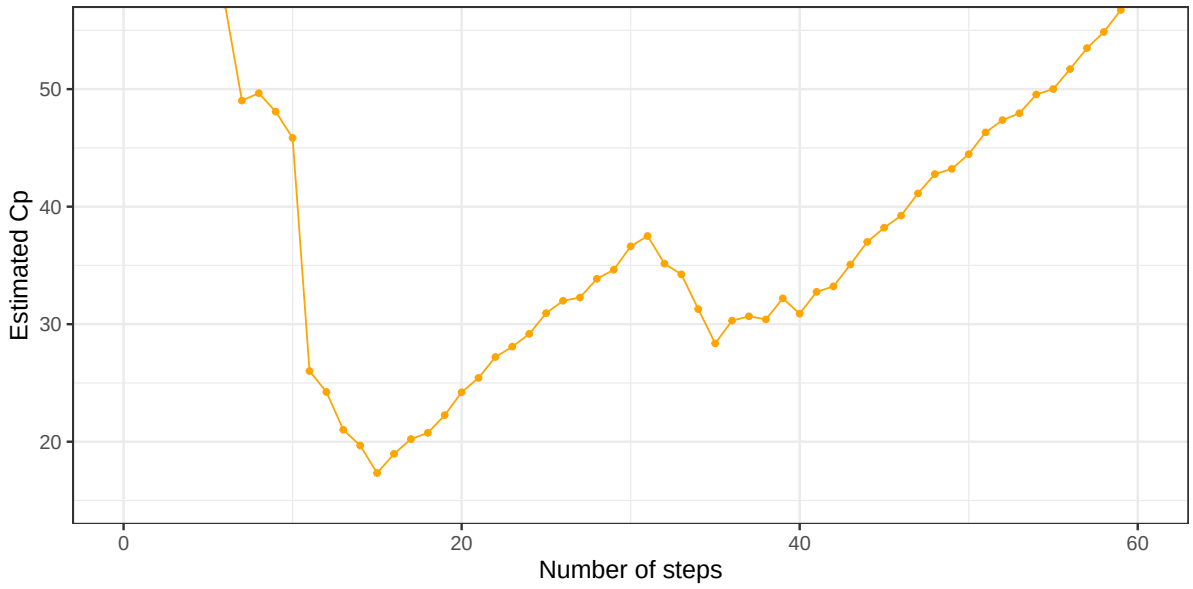
are lower bounds and upper bounds of the 95% confidence intervals for the bootstrap estimates, respectively. Like the absolute current correlations, the upper plot is corresponding to the model without interaction and the lower plot is corresponding to the quadratic model.





3.2.5 C_p-type selection criterion

The following plot displays $C_p(\hat{\boldsymbol{\mu}}_k)$ as a function of k for the diabetes data (the quadratic model) LARS estimates $\hat{\boldsymbol{\mu}}_k, k = 1, 2, \cdots, m = 64$. Minimum C_p was achieved at steps $k = 15$. The minimum C_p model looks sensible.



Based on C_p-type selection criterion, we prefer the estimate given by step $k = 15$. Selected predictors and corresponding coefficients are

sex	bmi	map	hdl	ltg	glu	age^2	bmi^2
-133.93	500.60	264.23	-201.88	470.16	26.23	18.655	43.46

glu^2	age:sex	age:map	age:ltg	age:glu	sex:map	bmi:map
77.29	116.33	30.694	13.144	9.12	8.62	91.07

Therefore, the selected model is

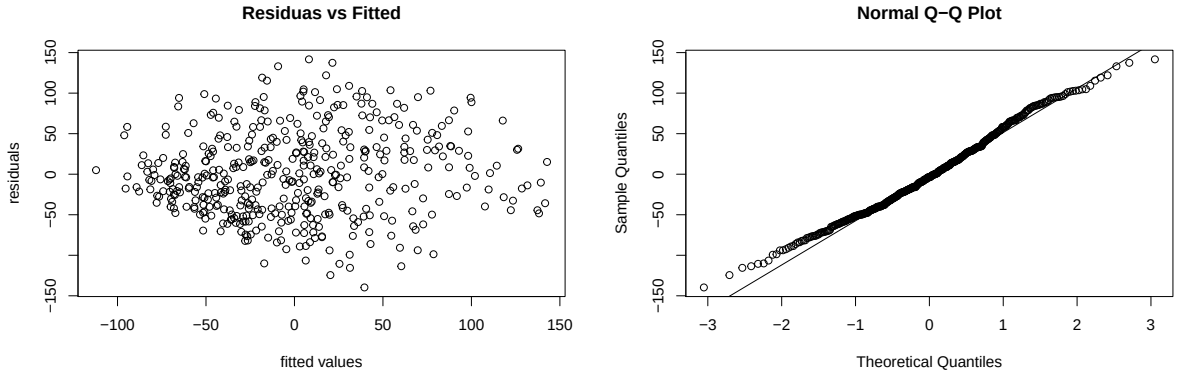
$$\mathbf{y} = -133.93\text{sex} + 500.60\text{bmi} + 264.23\text{map} - 201.88\text{hdl} + 470.16\text{ltg} +$$
$$26.23\text{glu} + 18.655\text{age}^2 + 43.46\text{bim}^2 + 77.29\text{glu}^2 + 116.33\text{age:sex} +$$
$$30.694\text{age:map} + 13.14\text{age:ltg} + 9.12\text{age:glu} + 8.62\text{sex:map} + 91.07\text{bmi:map}$$

Compared to the model without quadratic terms

$$\mathbf{y} = -197.7565\text{sex} + 522.265\text{bmi} + 297.16\text{map} - 103.946\text{tc} - 223.93\text{hdl} + 514.74948\text{ltg} + 54.7677\text{glu},$$

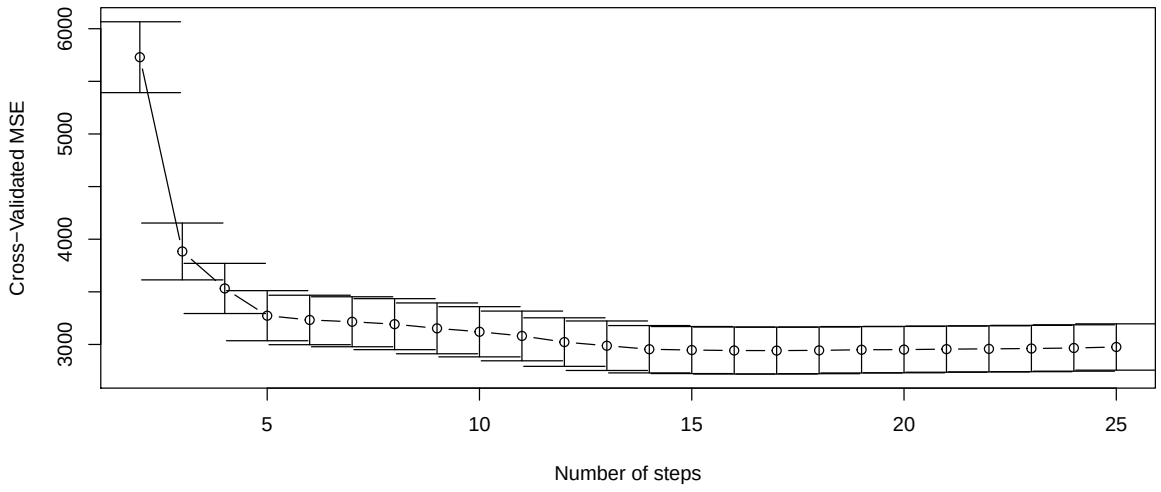
The model with quadratic terms selected the same once term except for `tc`. We can conclude that the following covariates: `sex`, `bmi`, `map`, `hdl`, `ltg`, `glu` are important factors in disease progression.

The following plots show the diagnosis of the final quadratic model. There is no relationship between the fitted values and the residuals. And the normality assumption is not violated.



3.2.6 K-fold cross-validated MSPE for LARS

To evaluate the performance of the selected model, we computed the 10-fold cross-validated mean squared prediction error for lars. As expected, the MSPE, 2965.124, given by step $k = 15$ is the smallest. Thus we can expect the selected model would perform well while predicting.



3.3 Discussion on the results

As authors pointed out in the paper, both Lasso and Stagewise are variants of the basic procedure LARS. These three model selection algorithms yield almost identical solutions, which LARS is computationally superior. Meanwhile, they all perform better than forward stepwise selection. We can select the preferred model along the path by C_p-type selection criterion.

References

[1] Efron B, Hastie T, Johnstone I, et al. *Least angle regression[J].* The Annals of statistics, 2004, 32(2): 407-499.

[2] Boyd, S., Boyd, S. P., Vandenberghe, L. *Convex optimization.* Cambridge university press, 2004.

[3] Hesterberg T, Choi N H, Meier L, et al. *Least angle and L_1 penalized regression: A review[J].* Statistics Surveys, 2008, 2: 61-93.

Code Appendix

```
knitr::opts_chunk$set(echo = TRUE)
library(lars)
library(ggplot2)
library(knitr)
data(diabetes)
attach(diabetes)
y.mean=mean(y)
y=y-y.mean
object <- lars(x,y,intercept = FALSE)
object2 <- lars(x,y,type="lar",intercept = FALSE)
object3 <- lars(x,y,type="for",intercept = FALSE)
plot(object)
plot(object2)
plot(object3)
plot(x=4:12,y=object$Cp[-(1:4)],type='b',col=1,xlab='step',ylab = 'Cp',main='lasso'
     ,ylim=c(5,25))
plot(x=4:10,object2$Cp[-(1:4)],type='b',col=1,xlab='step',ylab = 'Cp',main='lar'
     ,ylim=c(5,25))
plot(x=4:14,object3$Cp[-(1:4)],type='b',col=1,xlab='step',ylab = 'Cp',main='stagewise'
     ,ylim=c(5,25))
cv1=cv.lars(x,y,type='lasso',mode='step',intercept = FALSE)
cv2=cv.lars(x,y,type='lar',mode='step',intercept = FALSE)

#lar final model step 9
y.hat=x%*%as.matrix(object2$beta[9,],ncol=1)
residual=y-y.hat
plot(y.hat,residual,xlab='fitted values',ylab='residuals', main='Residuas vs Fitted')
qqnorm(residual)
qqline(residual)
data(diabetes)
attach(diabetes)
y <- y-mean(y)
object1 <- lars(x2, y, type = 'lasso', intercept = FALSE)
object1tc <- lars(x2, y, type = 'lasso', intercept = FALSE, max.steps = 10)
par(mfrow = c(1, 2))
plot(object1)
plot(object1tc)
par(mfrow = c(1, 1))
object2 <- lars(x2, y, type = 'forward.stagewise', intercept = FALSE)
object2tc <- lars(x2, y, type = 'forward.stagewise', intercept = FALSE, max.steps = 10)
par(mfrow = c(1, 2))
plot(object2)
plot(object2tc)
par(mfrow = c(1, 1))
object3 <- lars(x2, y, type = 'lar', intercept = FALSE)
object3tc <- lars(x2, y, type = 'lar', intercept = FALSE, max.steps = 10)
par(mfrow = c(1, 2))
plot(object3)
plot(object3tc)
par(mfrow = c(1, 1))
object2 <- lars(x,y,type="lar",intercept = FALSE)
c.kj=matrix(0,10,ncol(x))
for(i in 1:10){
  y.hat=x%*%object2$beta[i,]
  for(j in 1:ncol(x)){
```

```
    c.kj[i,j]=abs(sum(x[,j]*(y-y.hat)))
  }
}
c.kj=as.data.frame(c.kj)
C=apply(c.kj,1,max)
for(i in 1:ncol(x)){
  c.kj[i:10,object2$actions[[i]]]=C[i:10]
}

df=data.frame(step = (rep(1:10, 11)),
correlation = as.numeric(as.matrix(cbind(c.kj,C))),
group =gl(11,10,labels=c(1:10,'max')))

ggplot(df, aes(x=step, y=correlation))+
geom_point(aes(color=group))+
geom_line(aes(color=group))+
labs(x='Number of steps', y='Absolute current correlations')+
theme_bw(base_size=15)

mu <- x2%*%t(coef(object3)[2:(ncol(x2)+1), ])
crlt <- abs(sapply(1:ncol(mu), function(i) {
  sapply(1:ncol(x2), function(j) sum(x2[, j]*(y-mu[, i])))
  }))# one row corresponds to one variable
crlt <- rbind(crlt, apply(crlt, 2, max))
df <- data.frame(step = sort(rep(1:64, 65)),
                 chat = as.vector(crlt),
                 group = factor(rep(1:65, 64)))
p <- ggplot(df, aes(x=step, y=chat))+
  geom_point(aes(color=group))+
  geom_line(aes(color=group))+
  labs(x='Number of steps', y='Absolute current correlations')+
  annotate(geom="text", x=13, y=620, label='max current correlation', size=6)+
  geom_segment(aes(x = 9.5, y = 595, xend = 2, yend = 500), size=0.5)+
  theme_bw(base_size=15)+
  theme(legend.position = "none")
p
# lasso
object <- lars(x2, y, type="lasso", intercept=FALSE)
beta <- coef(object, s=11, mode = 'step')
mu <- x2%*%beta
eps <- y-mu
B <- 100
K <- 40
pe1 <- matrix(0, nrow = B, ncol = K)
nonzero1 <- rep(0, K)
for(b in 1:B){
  epsb <- sample(eps, replace = TRUE)
  yb <- mu+epsb
  objectb <- lars(x2, yb, type="lasso", intercept=FALSE, max.steps = K)
  nonzero1 <- nonzero1+apply(coef(objectb)[2:(K+1), ]!=0, 1, sum)/B
  mub <- x2%*%t(coef(objectb)[2:(K+1), ])# n*K
  pe1[b, ] <- apply(mub, 2, function(x) 1-sum((x-mu)^2)/sum(mu^2))
}

# forward.stagewise
object <- lars(x2, y, type="forward.stagewise", intercept=FALSE)
beta <- coef(object, s=11, mode = 'step')
```

```r
mu <- x2%*%beta
eps <- y-mu
B <- 100
K <- 40
pe2 <- matrix(0, nrow = B, ncol = K)
nonzero2 <- rep(0, K)
for(b in 1:B){
  epsb <- sample(eps, replace = TRUE)
  yb <- mu+epsb
  objectb <- lars(x2, yb, type="forward.stagewise", intercept=FALSE, max.steps = K)
  nonzero2 <- nonzero2+apply(coef(objectb)[2:(K+1), ]!=0, 1, sum)/B
  mub <- x2%*%t(coef(objectb)[2:(K+1), ])# n*K
  pe2[b, ] <- apply(mub, 2, function(x) 1-sum((x-mu)^2)/sum(mu^2))
}

# lar
object <- lars(x2, y, type="lar", intercept=FALSE)
beta <- coef(object, s=11, mode = 'step')
mu <- x2%*%beta
eps <- y-mu
# sum(mu^2)/(sum(mu^2)+sum(eps^2))
# 0.4156301
B <- 100
K <- 40
pe3 <- matrix(0, nrow = B, ncol = K)
for(b in 1:B){
  epsb <- sample(eps, replace = TRUE)
  yb <- mu+epsb
  objectb <- lars(x2, yb, type="lar", intercept=FALSE, max.steps = K)
  mub <- x2%*%t(coef(objectb)[2:(K+1), ])# n*K
  pe3[b, ] <- apply(mub, 2, function(x) 1-sum((x-mu)^2)/sum(mu^2))
}

# forward selection
object <- lars(x2, y, type="lar", intercept=FALSE)
beta <- coef(object, s=11, mode = 'step')
mu <- x2%*%beta
eps <- y-mu
B <- 100
K <- 40
pe4 <- matrix(0, nrow = B, ncol = K)
for(b in 1:B){
  epsb <- sample(eps, replace = TRUE)
  yb <- mu+epsb
  objectb <- lars(x2, yb, type="stepwise", intercept=FALSE, max.steps = K)
  mub <- x2%*%t(coef(objectb)[2:(K+1), ])# n*K
  pe4[b, ] <- apply(mub, 2, function(x) 1-sum((x-mu)^2)/sum(mu^2))
}

df <- data.frame(x=1:K, nonzero1, nonzero2,
                 pe1=apply(pe1, 2, mean),
                 pe2=apply(pe2, 2, mean),
                 pe3=apply(pe3, 2, mean),
                 pe4=apply(pe4, 2, mean),
                 sd=apply(pe3, 2, sd))
ggplot(df)+
  geom_line(aes(x=x, y=pe3), color='black', size=0.6)+
```

```
  geom_line(aes(x=nonzero1, y=pe1), linetype='dotted', color='red', size=1.2)+
  geom_line(aes(x=nonzero2, y=pe2), linetype='dashed', color='green', size=0.8)+
  geom_line(aes(x=x, y=pe4), linetype='dashed', color='blue', size=0.7)+
  geom_point(aes(x=x, y=pe3+sd), color='blue', size=0.7)+
  geom_point(aes(x=x, y=pe3-sd), color='blue', size=0.7)+
  annotate(geom="text", x=12, y=0.8, label="forward\n selection", size=6, color='blue')+
  geom_segment(aes(x = 12, y = 0.815, xend = 14, yend = 0.886), size=0.5, color='blue')+
  annotate(geom="text", x=37, y=0.87, label="lasso", size=6, color='red')+
  geom_segment(aes(x = 37, y = 0.873, xend = 35, yend = 0.907), size=0.5, color='red')+
  annotate(geom="text", x=38, y=0.94, label="lars", size=6)+
  geom_segment(aes(x = 38, y = 0.935, xend = 39, yend = 0.903), size=0.5)+
  annotate(geom="text", x=33, y=0.96, label="stage", size=6, color='green')+
  geom_segment(aes(x = 33, y = 0.955, xend = 30, yend = 0.935), size=0.5, color='green')+
  labs(x='Average number of terms', y='Proportion explained')+
  coord_cartesian(ylim=c(0.75, 1))+
  theme_bw(base_size=15)

B=500
steps=10
fit1=lm(y~x-1)
sigma1=sigma(fit1)
mu1=fit1$fitted.values
y.star=sapply(1:B, function(o){mu1+rnorm(nrow(x),sd=sigma1)})

#dim=[y,steps,B replication]
y.hat=array(0,dim=c(length(y),steps,B))
for(i in 1:B){
  fit2=lars(x,y.star[,i],type="lar",intercept = FALSE)
  y.hat[,,i]=x%*%t(fit2$beta[-1,])
}
ystar.mean=apply(y.star,1,mean)

cov.hat=matrix(0,steps,length(y))
for(i in 1:steps){
  for(j in 1:length(y)){
    cov.hat[i,j]=sum((y.hat[j,i,])*(y.star[j,]-ystar.mean[j]))/(B-1)
  }
}
df.hat=apply(cov.hat,1,sum)/sigma1^2

cov.hat=array(0,dim=c(steps,length(y),10))
for(i in 1:steps){
  for(j in 1:length(y)){
    for(k in  seq(1, 500, 50)){
      cov.hat[i,j,ceiling(k/50)]=sum((y.hat[j,i,k:(k+49)])*(y.star[j,k:(k+49)]
                                                            -mean(y.star[j,k:(k+49)])))/(50-1)
    }

  }
}

dfmat=matrix(0,steps,10)
for(i in 1:steps){
    dfmat[i,]=apply(cov.hat[i,,],2,sum)/sigma1^2
}

s <- apply(dfmat, 1, sd)
```

```r
alpha <- 0.05
df1 <- data.frame(x=1:ncol(x), df.hat,lcb=apply(dfmat, 1, mean)-qt(1-alpha/2, df=9)*s/sqrt(10)
                  ,ucb=apply(dfmat, 1, mean)+qt(1-alpha/2, df=9)*s/sqrt(10))


ols <- lm(y~x2-1)
beta <- coef(ols)
sigma <- sigma(ols)
mu <- x2%*%beta
B <- 500
mumat <- rep(list(matrix(0, ncol=B, nrow=nrow(x2))), ncol(x2))
ymat <- matrix(0, ncol=B, nrow=length(y))
for(b in 1:B){
  ymat[, b] <- sapply(mu, function(i) rnorm(1, mean=i, sd=sigma))
  objectb <- lars(x2, ymat[, b], type="lar", intercept=FALSE)
  mub <- x2%*%t(coef(objectb)[2:(ncol(x2)+1), ])# n*64
  mumat <- lapply(1:ncol(mub), function(i) {
    mumat[[i]][, b] = mub[, i]
    mumat[[i]]
    })
}

dfhat <- sapply(mumat, function(mui) {
  sum(sapply(1:nrow(ymat), function(i){
    sum(mui[i, ]*(ymat[i, ]-mean(ymat[i, ])))/(B-1)
  }))/sigma^2
})

dfmat <- sapply(seq(1, 500, 50), function(k) {
  sapply(mumat, function(mui) {
    sum(sapply(1:nrow(ymat), function(i){
      sum(mui[i, k:(k+49)]*(ymat[i, k:(k+49)]-mean(ymat[i, k:(k+49)])))/(50-1)
    }))/sigma^2
  })
})

s <- apply(dfmat, 1, sd)

alpha <- 0.05
lcb <- apply(dfmat, 1, mean)-qt(1-alpha/2, df=9)*s/sqrt(10)
ucb <- apply(dfmat, 1, mean)+qt(1-alpha/2, df=9)*s/sqrt(10)

df <- data.frame(x=1:ncol(x2), dfhat, lcb, ucb)

par(mfrow=c(1,2))
ggplot(df1)+
geom_segment(aes(x = 0, y = 0, xend = 10, yend = 10))+
geom_point(aes(x=x, y=df.hat), color='red', size=1.5)+
geom_errorbar(aes(x=x, ymin=lcb, ymax=ucb), alpha=0.5, width=0.5)+
labs(x='Number of steps', y='df estimate')+
theme_bw(base_size=15)

ggplot(df)+
  geom_segment(aes(x = -10, y = -10, xend = 75, yend = 75))+
  geom_point(aes(x=x, y=dfhat), color='red', size=1.5)+
  geom_errorbar(aes(x=x, ymin=lcb, ymax=ucb), alpha=0.5, width=0.5)+
  labs(x='Number of steps', y='df estimate')+
  coord_cartesian(xlim=c(-5, 70), ylim=c(-5, 70))+
```

```
  theme_bw(base_size=15)
df <- summary(object3)
ggplot(df, aes(x=Df, y=Cp)) +
  geom_line(color='orange')+
  geom_point(color='orange')+
  labs(x = 'Number of steps', y = 'Estimated Cp')+
  coord_cartesian(xlim=c(0, 60), ylim=c(15, 55))+
  theme_bw(base_size=15)
object3 <- lars(x2, y, type = 'lar', intercept = FALSE)
y.hat <- x2%*%coef(object3)[16, ]
residual=y-y.hat
plot(y.hat,residual,xlab='fitted values',ylab='residuals', main='Residuas vs Fitted')
qqnorm(residual)
qqline(residual)
k <- which(object3$Cp==min(object3$Cp))# smallest Cp
beta <- coef(object)[16, ]
kable(matrix(beta[beta!=0], ncol=5, byrow=TRUE), digits=4)
cv10 <- cv.lars(x2, y, K=10, index=2:25, type='lar', mode='step', max.steps=25)
detach(diabetes)
```