Linear Model Selection Based on Extended Robust Least Angle Regression

by

Hongyang Zhang

B.Sc., Zhejiang University, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate Studies

(Statistics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2012

Abstract

In variable selection problems, when the number of candidate covariates is relatively large, the "two-step" model building strategy, which consists of two consecutive steps *sequencing* and *segmentation*, is often used. Sequencing aims to first sequence all the candidate covariates to form a list of candidate variables in which more "important" ones are likely to appear at the beginning. Then, in the segmentation step, the subsets of the first m (chosen by the user) candidate covariates which are ranked at the top of the sequenced list will be carefully examined in order to select the final prediction model. This thesis mainly focuses on the sequencing step.

Least Angle Regression (LARS), proposed by Efron, Hastie, Johnstone and Tibshirani (2004), is a quite powerful step-by-step algorithm which can be used to sequence the candidate covariates in order of their importance. Khan, Van Aelst, and Zamar (2007) further proposed its robust version — Robust LARS. Robust LARS is robust against outliers and computationally efficiency. However, neither the original LARS nor the Robust LARS is available for carrying out the sequencing step when the candidate covariates contain both quantitative and nominal variables. In order to remedy this, we propose the Extended Robust LARS by proposing the generalized definitions of correlations which includes the correlations between nominal variables and quantitative variables. Simulations and real examples are used to show that the Extended Robust LARS gives superior performance to two of its competitors, the classical Forward Selection and Group Lasso.

Table of Contents

List of Tables

List of Figures

Acknowledgements

This thesis would not have been possible without the continuous encouragement, support and guidance from my supervisor Professor Ruben Zamar. Also, I would like to thank my second reader Professor Matas Salibián-Barrera for his help in my thesis. Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of this thesis.

Chapter 1

Introduction

Generally speaking, there are two different strategies for linear model selection: "one-step model building" and "two-step model building". The one-step model building procedure aims to build a final prediction (or explanatory) model in one step by using step-by-step algorithms such as Forward Selection (FS) and Stepwise (SW). Unlike the one-step procedure, the two-step model building strategy contains two consecutive procedures: *sequencing* and *segmentation*. To be more specific, sequencing is a step that aims to first sequence all the candidate covariates and thus form a list of candidate variables in which more "important" ones are likely to appear at the beginning. Then, as the continuation of the sequencing step, the segmentation step will focus on the first m candidate covariates that are ranked at the top of the sequenced list (Note that m should be decided by people who are building the model). In this step, the subsets of the m chosen covariates will be carefully examined in order to select the final prediction model.

The two-step model building strategy is often used when the number of the candidate variables is large, because in such a case, rather than building a prediction model with a large amount of predictor variables, it is more realistic to first screen out the less important variables (in the sequencing step), then try to build a prediction model based on only the chosen important ones (in the segmentation step).

From the above introduction, it can be easily seen that the capability of the sequencing step to keep the important candidate variables while screening out the unimportant ones is quite crucial for the performance of the whole two-step model procedure, in other words, choosing the real important candidate covariates is an essential step in order to further producing a good prediction or explanation model from the segmentation step (it is almost impossible for the segmentation step to come up with a good prediction or explanation model based on only the unimportant variables). Thus, in this thesis, we will focus on the sequencing step (the first step) of the two-step model building procedure.

In the sequencing step, several different algorithms can be used. For example, the step-by-step algorithms: Forward Selection (FS), Forward Stagewise (Stagewise) [3] and Least Angle Regression (LARS) [2] all can help us sequence the candidate covariates. LARS is a quite powerful step-by-step algorithm and it has been shown to be favorable in several aspects compared to other step-by-step algorithms such as FS and Stagewise. In [2], Efron et al. showed that by slightly modifying the steps in LARS algorithm, the modified LARS yield the same solution path with another popular model selection algorithm Lasso [6].

However, in [5], Khan et al. showed that in the sequencing step, the sequences generated by LARS is not robust against outliers. In other words, the sequence can vary a lot if the data are contaminated. So they robustified LARS against outliers and thus proposed the Robust LARS. Yet, even the Robust LARS is not "robust" enough in the sense that it cannot be used to sequence the candidate covariates which contain nominal (i.e. categorical) variables. As we all know, it is common that there exists several nominal variables among the candidate covariates and it is also possible that some of these nominal variables are "important" predictor variables (e.g. the prediction accuracy will be improved by including those nominal variables in the regression model). In such cases, neither the original LARS nor the Robust LARS is available for carrying out the sequencing step. In order to remedy this, we are motivated to further robustifying LARS so it can be applicable for sequencing both quantitative and nominal variables.

In [5], Khan et al. also illustrated that, if LARS is used to sequence the covariates, the algorithm will only depend on sample means, variances and the pairwise correlations (among the candidate covariates) rather then the data themselves, and also, LARS algorithm can be expressed in the form of the sample correlation matrix. However, as long as we know, there are no proper definitions of sample correlations between one quantitative variable and one nominal variable (that are suitable for LARS), so the LARS algorithm cannot be applied to datasets that contain nominal variables. Thus, in this thesis, we first propose our generalized definitions of correlations which includes the correlations between nominal variables and quantitative variables, and then try to incorporate these definitions into LARS algorithm. Thus, we further propose the Extended Robust LARS which can be used for sequencing both quantitative and nominal variables while remaining the robust properties against outliers. In order to check the performance of the Extended Robust LARS, we run some simulation studies to compare it with two competitor methods, Forward Selection and the Group Lasso [7], which can also be used for sequencing quantitative and nominal variables.

The rest of this thesis is organized as follows. In Chapter 2, we review LARS briefly and express the LARS procedure in terms of the correlation matrix of the data (see [5] for details). In Chapter 3, we propose the definitions of the correlation between two nominal variables, and also, between one nominal variable and one quantitative variable. In Chapter 4, we incorporate the "extended" correlations defined in Chapter 3 in LARS in order to robustify LARS against nominal variables, and thus propose our Extended Robust LARS. Further, the Extended Robust LARS will be speed up and its performance will be checked through several simulation experiments. In Chapter 5, some numerical examples based on several real data are shown.

Chapter 2

Brief Review of LARS

In this chapter, we will briefly review the Least Angle Regression (LARS) algorithm.

Note that in this section, we will first assume all the covariates are quantitative variables, the cases when the covariates contain both quantitative and categorical variables will be discussed in later chapters.

In order to first get some insight of LARS, we can begin with a closely related algorithm called Forward Stagewise [3] by which LARS is motivated. In the following section, the Forward Stagewise algorithm will be reviewed and we will briefly discuss its advantages and disadvantages.

2.1 Forward stagewise procedure

Suppose we have the response variable Y and candidate covariates X_1, X_2, $\cdots$, X_d. Without loss of generality, we can assume that the covariates are all standardized (i.e. all the covariates have mean 0 and variance 1), and the response variable has mean 0. Denote ϵ as a small positive constant (typically smaller than the absolute value of the regression coefficients in an ordinary linear regression). Then according to [5], the Stagewise algorithm can be described as follows:

1. Set the prediction vector: $\hat{\boldsymbol{\mu}} = \mathbf{0}$.

2. Calculate $\hat{c}_j = X_j'(Y - \hat{\boldsymbol{\mu}}), j = 1, \cdots, d$,

   where $\hat{c}_j$ is proportional to the correlation between X_j and the current residual.

3. Let $m = \texttt{argmax}_j |\hat{c}_j|$. Then the current prediction vector will be updated as follows:

   $$\hat{\boldsymbol{\mu}} \leftarrow \hat{\boldsymbol{\mu}} + \epsilon \ \texttt{sign}(\hat{c}_m) X_m,$$

   where ϵ is a (small) positive constant.

4. Repeat steps 2 and 3.

Once we repeat the above steps, the algorithm updates the prediction and at the same time, records the sequence of covariates as they enter the model.

Advantages of forward stagewise (stagewise) over forward selection (FS)

In order to better understand the advantages of the Forward Stagewise (Stagewise) procedure, we will compare the Stagewise with the classical algorithm Forward Selection (FS) as they are highly related. In FS, the candidate variable that has the largest absolute correlation with response variable Y will be selected as the first predictor (X_1, say). Once X_1 is selected, all the other predictors will be regressed after being adjusted by X_1, and the next predictor that enters the model will be decided according to the current residual vectors (i.e. the residual vectors after updating X_1). Then again, the other predictors will be regressed after being adjusted by the first two selected variables and the updated residual vectors will be used to decide the third predictor that enters the model, and so on. The procedure introduced above may cause a problem: some important predictors which are "accidentally" highly correlated with the selected variables (such as X_1) are not likely to be chosen in the following "competition" since the residual vectors are already adjusted by the selected variables. So we usually consider FS as an **aggressive** model-building algorithm.

As we can see from the above section, the Forward Stagewise procedure is quite similar to FS but **less aggressive**. Unlike FS, the Stagewise procedure will take many tiny steps to approach to a final model instead of taking a relatively "big" step within each selected variable (i.e. adjust the residual vectors by the selected variables and then select other variables after the adjustment). In Step 1 of Stagewise, we set the zero vector as the initial prediction of the response variable. Then if X_1 is selected as the first predictor to enter the model according to Step 2, the prediction will "move" a tiny step along the direction of X_1. Then we can get the new residual vector and continue the process above repeatedly until we obtain the required number of predictors in the model. From the above procedure, we can see that even if some other predictors are highly correlated with the selected variable (such as X_1), they will still have chances to enter the "competition" after the algorithm goes along the direction of the selected variable for only several tiny steps. Note that the aim of the Stagewise procedure here is to obtain the first m variables that have entered the model.

Although FS is an relatively aggressive algorithm, someone still can argue that the model chosen by FS might yield higher R^2 (at least at some certain stages of the selection procedure), and FS can guarantee the orthogonalization of the following selected covariates with respect to the active ones (the selected ones), which is a property that cannot be guaranteed by Stagewise. However, it should be noticed that we are now only focusing on the sequencing step in the two-step model building procedure. This means in each selection stage in either FS or Stagewise algorithm, our goal is not actually "fitting" the model, rather, we are just "selecting" or "screening" the covariates. Certainly, as argued in [5], when we are at the second selection stage (i.e. the stage where the second variable enters the model), the minimizer of the Stagewise loss cannot beat the minimizer of the FS loss. This is because FS already considers the residual sum of squares of the final fit at this stage. For example, if FS chooses $\{X_1, X_3\}$ and Stagewise selects $\{X_1, X_4\}$, then FS will have a smaller loss (because FS already considers fitting the final model by using $\{X_1, X_3\}$), which will further result in a greater value of R^2. However, for the next selection stage if, for example, FS selects $\{X_1, X_3, X_5\}$, and Stagewise selects $\{X_1, X_4, X_6\}$, it is not necessarily true that FS will still yield a small loss. This is because FS just considers the possible paths after adjusted for the path $\{X_1, X_3\}$. The

Stagewise combination (e.g. $\{X_1, X_4, X_6\}$) has not been considered by FS at all as FS has already taken a different path along $\{X_1, X_4\}$ from the second stage. From this example, we can see that FS cannot always guarantee greater R^2 for a particular subset size in all cases. Therefore, orthogonalization of the following selected covariates with respect to the active ones does not usually make much sense. This is another reason why researchers often prefer Stagewise to FS.

Problems of forward stagewise

Although Stagewise is favorable compared to FS, there are still some problems with it. One serious problem of the Stagewise procedure is how to choose a suitable ϵ. Actually, in [5], Khan et al. mentioned that the performance (even the convergency of the Forward Stagewise procedure) depends on the choice of ϵ because of the following reasons:

- If ϵ is chosen to be very "small", then according to the algorithm steps mentioned above earlier in Section 2.1, too many tiny Stagewise steps are needed in order to obtain a final model. This will definitely make the algorithm computational burdensome and ineffective.
- If ϵ is chosen to be quite "large", then according to [5], the following two problems may occur:

    1. Aggressiveness of the algorithm: If $\epsilon \to |\hat{c}_m|$ (refer to the Steps of Stagewise algorithm earlier in this section) is "large", then similar to the FS algorithm, Stagewise will tend to eliminate the covariates that are correlated with the active ones from the following

competitions, which makes Stagewise algorithm aggressive.

2. Non-convergence of the algorithm: In some certain cases, when the mth covariate, X_m say, has just been selected (i.e. Stagewise has already selected m predictors), the remaining inactive predictors (i.e. predictors that are not selected yet) may already have very small absolute correlations with the current residual vector. Suppose the correlation between X_m and the current residuals is positive. If ϵ is large, once the prediction is updated with an "$\epsilon - step$" along the direction of "$+X_m$", the correlation between X_m and the newly updated residuals may become negative and has a larger absolute value than the correlations between other inactive covariates and the updated residual. In such a case, Stagewise has to update the prediction with an "$\epsilon - step$" along the direction of "$-X_m$". In such a case, the Stagewise steps tend to just move back-and-forth in a close loop, and the algorithm procedure might be endless.

The problem of choosing an appropriate ϵ makes the Stagewise algorithm not computationally stable. However, this problem of Stagewise provides a motivation for LARS. LARS is able to overcome this problem by taking a mathematical approach.

2.2 The LARS algorithm

As we mentioned, LARS is motivated by the Stagewise algorithm. Instead of taking many tiny steps to modify the prediction, LARS updates the prediction and records the order of the variables based on mathematical approaches. We will just review the general algorithm procedure, the detail mathematical derivations can be found in [5].

Suppose X_1 is the first selected variable in Stagewise procedure (i.e. X_1 has the largest absolute correlation with Y). If ϵ is chosen to be "small", then Stagewise will modify the prediction by moving it along the direction of X_1 for several tiny steps until it reaches a certain point where the second selected predictor, X_2 say, enters the model. At this particular point, X_1 and X_2 should have equal absolute correlation with the current residual. Based on this important property, LARS derives a formula to determine this point mathematically. Thus, LARS can update the prediction by moving directly to this point in one single step instead of several tiny steps in Stagewise procedure.

As we mentioned, X_2 is the second predictor X_2 that enters the model. Stagewise will modify the prediction by moving along the direction of X_2 for some tiny steps. However, it is highly possible that after updating the prediction for several steps in the direction of X_2, the absolute correlation between X_1 and the newly updated residual becomes larger, then Stagewise will move back in the direction of X_1. Thus, by alternating between these two directions (i.e. X_1 and X_2), Stagewise actually updates the prediction by moving along a direction "in between", along which the absolute correlations of X_1 and X_2 with the updated residual are approximately the same (until a third selected predictor enters the model). LARS just simplifies the Stagewise procedure by mathematically finding this direction which guarantees that the correlations of X_1 and X_2 with the residual are equal, then in a single step, the prediction can be moved along this direction to a point where a third predictor, X_3 say, has equal absolute correlation with the updated residual vector, and so on.

The original LARS algorithm is designed to obtain the updated predictions at each step, as well as the sequence of the covariates when they enter the model. In [5], Khan et al. showed that, if we are only interested in the sequence of the covariates as they enter the model, the LARS algorithm can be expressed in terms of the correlation matrix of the data (not the observations themselves).

2.3 LARS algorithm expressed in terms of correlation

As mentioned, it is proved in [5] that the sequence of covariates obtained by LARS can be derived from the correlation matrix of the data (without using the observations themselves). In this section, we will review how to express the LARS algorithm in terms of correlation matrix. Note that in this section, we will still assume all the covariates are quantitative variables.

Let Y, $X_1, \cdots, X_d$ be the variables. Y is the response variable and $X_1, \cdots, X_d$ are the covariates. Let's assume, without loss of generality, the variables are standardized using their means and standard deviations. Denote r_{jY} as the correlation between X_j and Y, and let R_x be the correlation matrix of the covariates $X_1, \cdots, X_d$.

Suppose that X_m has the maximum absolute correlation r with Y and denote $s_m = \mathtt{sign}(r_{mY})$. Then, X_m becomes the first *active variable* and the current prediction $\hat{\boldsymbol{\mu}} \leftarrow \mathbf{0}$ should be modified by moving along the direction of $s_m X_m$ to a certain distance γ until the second selected covariate (second active variable) enters the model. Note that γ can be expressed in terms of correlations between the variables (see [5] for details). And the second active variable is identified simultaneously as LARS determines γ.

Once we have more than one active variable, LARS will modify the current prediction by moving it along the *equiangular direction*, which is the direction (i.e. a linear combination of the active covariates) that has equal angle (correlation) with all current active covariates. LARS modifies the prediction by moving along this direction because it can make sure that the current correlation between each active covariate and the residual decreases at an equal speed. Let A be the set of subscripts corresponding to the active variables. The standard equiangular vector B_A can be derived mathematically (see [5] for details). However, the selection of the next active variable is not determined by B_A directly, what really matters are the correlations between all the covariates (both active and inactive) with B_A, which can be expressed in terms of the correlation matrix of all the covariates. LARS will modify the current prediction by moving along the direction of B_A by a distance γ_A until the next active variable enters the model. This distance can also be expressed in terms of the correlation matrix of the variables.

As introduced above, we can see that the covariates sequenced by the LARS algorithm are actually a function of the correlation matrix of the standardized data. In [5], the LARS algorithm are expressed in terms of correlation r_{jY} between X_j and Y, and the correlation matrix R_x of the covariates:

1. Set the active set, $A = \phi$, and the sign vector $\boldsymbol{s_A} = \phi$.

2. Determine $m = \texttt{argmax}|r_{jY}|$, and $s_m = \texttt{sign}\{r_{mY}\}$. Let $r = s_m r_{mY}$.

3. Put $A \leftarrow A \cup \{m\}$, and $\boldsymbol{s_A} \leftarrow \boldsymbol{s_A} \cup \{s_m\}$.

4. Calculate $a = [\boldsymbol{1'_A}(D_A R_A D_A)^{-1}\boldsymbol{1_A}]^{-1/2}$, where $\boldsymbol{1_A}$ is a vector of $1's$, $D_A = diag(\boldsymbol{s_A})$, and R_A is the submatrix of R_X corresponding to the active variables. Calculate $\boldsymbol{w_A} = a(D_A R_A D_A)^{-1}\boldsymbol{1_A}$, and $a_j = (D_A \boldsymbol{r_{jA}})'\boldsymbol{w_A}$, for $j \in A^c$, where $\boldsymbol{r_{jA}}$ is the vector of correlations between X_j and the active variables. (Note that, when there is only one active covariate X_m, the above quantities simplify to $a = 1,\ w = 1$, and $a_j = r_{jm}$.)

5. For $j \in A^c$, calculate $\gamma_j^+ = (r - r_{jY})/(a - a_j)$, and $\gamma_j^- = (r + r_{jY})/(a + a_j)$, and let $\gamma_j = \texttt{min}(\gamma_j^+, \gamma_j^-)$. Determine $\gamma = \texttt{min}\{\gamma_j, j \in A^c\}$, and m, the index corresponding to the minimum $\gamma = \gamma_m$. If $\gamma_m = \gamma_m^+$, set $s_m = +1$. Otherwise, set $s_m = -1$. Modify $r \leftarrow r - \gamma a$, and $r_{jY} \leftarrow r_{jY} - \gamma a_j$, for $j \in A^c$.

6. Repeat steps 3, 4 and 5.

According to the above algorithm steps, the chosen candidate covariates are stored in vector A in the order of importance, and the corresponding signs of these covariates are kept in the sign vector s_A.

Chapter 3

Sample Correlations Between Any Combinations of Quantitative and Nominal Variables

As mentioned in Section 2.3, LARS can be expressed in terms of the correlation matrix of the data (rather than the data themselves). In this chapter, we will introduce the (sample) correlation coefficient between each combination of quantitative and nominal variables in order to form a generalized correlation matrix of the covariates that contain both quantitative and nominal variables. Note that we will focus on the approaches to calculate the pairwise sample correlation instead of the correlation at the population level.

In Section 3.1, we will first review the bivariate Winsorization method [5] which can be used to calculate the pairwise sample correlations between quantitative variables. Then the sample correlation between a quantitative variable and a nominal variable, as well as the correlation between two nominal variables will be defined in the following sections.

3.1 Robust sample correlation between two quantitative variables: bivariate winsorization

In order to obtain the sample correlation between two quantitative variables, the classical Pearson correlation is commonly used. However, the classical Pearson correlation coefficient is not robust against outliers.

A more robust way to calculate correlation is proposed in [5]. This paper proposed to first apply a bivariate Winsorization (which will be introduced later) to the two quantitative variables, and then the robustified correlation is defined as *the classical correlation coefficient of the bivariate Winsorized data.*

Suppose we want to apply bivariate Winsorization to two quantitative variables with the same size n (i.e. paired bivariate data), then the bivariate transformation $\boldsymbol{u} = \texttt{min}(\sqrt{C/D(\boldsymbol{x})}, 1)\boldsymbol{x}$ ($\boldsymbol{x} = (x_1, x_2)^t$) should be applied to each pair of the two quantitative variables, where $C = 5.99$ is the tuning constant (5.99 is the 95% quantile of the χ_2^2 distribution), $D(\boldsymbol{x})$ is the Mahalanobis distance based on an initial bivariate correlation matrix R_0. Then the classical correlation coefficient of $\boldsymbol{u}$ is defined as the robustified correlation of $\boldsymbol{x} = (x_1, x_2)^t$.

Figure 3.1 shows the bivariate Winsorizations for a sample data set including several obvious outliers. The ellipse for this contaminated data is shown in the figure (the ellipse for the data without outliers is only slightly smaller than that for the contaminated data, since the two ellipses almost coincide, we just show the one for the contaminated data). Recall the bivariate transformation equation mentioned above, we can see that by using bivariate Windsorization, the outliers are shrunken to the boundary of the ellipsoid.

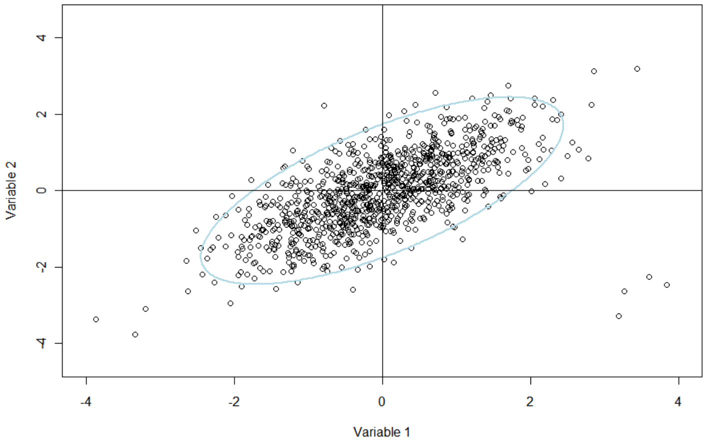


Figure 3.1: Bivariate winsorization for the contaminated data

Choosing an appropriate initial correlation matrix R_0 is crucial for bivariate Winsorization. In order to obtain R_0, the adjusted Winsorization method [5] can be used for calculating the pairwise correlations in R_0. Adjusted Winsorization is an extension of Huber's (1981) one-dimensional Winsorization [4]. In order to make the correlation coefficients more robust against outliers, Huber suggested that the correlations can be calculated by the classical Pearson correlation coefficients of the transformed (one-dimensional Winsorized) data. To be more specific, for n univariate data $X = (x_1, x_2, \cdots, x_n)^T$, the transformed data can be obtained by

$$u_i = \psi_c((x_i - med(X))/mad(X)), \quad i = 1, 2, \cdots, n,$$

where the Huber score function $\psi_c(x)$ is defined as $\psi_c(x) = \texttt{min}\{\texttt{max}\{-c, x\}, c\}$. Notice that c is a tuning constant chosen by the user, e.g., $c = 1.345$ or $c = 2$. Unlike one-dimensional Winsorization, it is suggested in [5] to transform the data by adjusted Winsorization method, which needs two tuning constants, say c_1 and c_2. c_1 and c_2 can be determined as follows: First standardize the two quantitative variables (with mean 0 and variance 1), and plot one against another in four quadrants. Then c_1 is the tuning constant for the two quadrants which contain the majority of the standardized data and c_2 is a smaller tuning constant for the other two quadrants. Similar to the tuning constant in Huber's one-dimensional Winsorization, the tuning constant c_1 should be chosen by the user. Suppose c_1 is chosen to be 1.345 or 2 (In this thesis, we will choose $c_1 = 1.345$), then c_2 can be determined by

$$c_2 = hc_1, \quad (h = n_2/n_1),$$

where n_1 is the number of (paired) data in the quadrants that contain the majority of the data and $n_2 = n - n_1$. *Finally, the initial correlation matrix R_0 can be obtained by calculating the classical correlation matrix of the adjusted Winsorized data.*

Figure 3.2 shows how the contaminated data are transformed by the adjusted Windsorization. The bivariate outliers are now shrunken to the boundary of the squares.

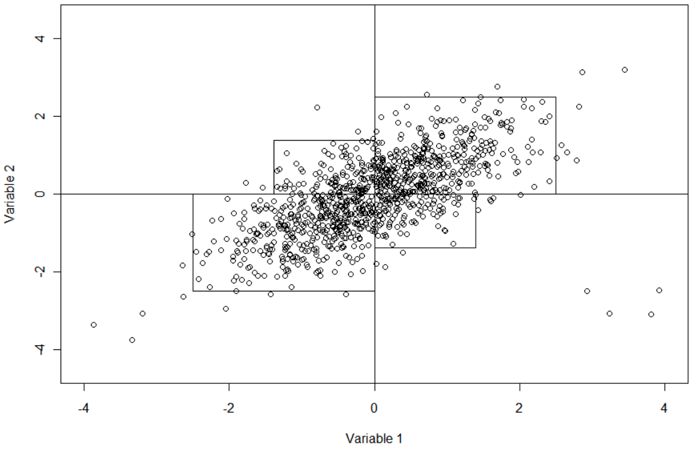


Figure 3.2: Adjusted windsorization (for initial estimate R_0) with c = 2.5.

3.2 Sample correlation between quantitative and nominal variables

As we mentioned above, LARS is based on the correlation matrix of the data rather the data itself. So in order to extend LARS to make it applicable for sequencing both quantitative and nominal variables, we proposed to construct a generalized correlation matrix that incorporate both the quantitative variables and the nominal variables. However, there are no standard techniques of calculating the correlation coefficients between one quantitative and a nominal variable. So, in Section 3.2.1, we propose a definition of calculating the correlation between a quantitative and a nominal variable.

3.2.1 Sample correlation between quantitative variables and dichotomous variables

Dichotomous (or dummy) variables, such as gender, are commonly seen in the regression problem. For calculating the correlation between a quantitative variable and a dichotomous variable, the classical point biserial correlation coefficient is often used. Note that the point-biserial correlation is shown to be mathematically equivalent to the Pearson (product moment) correlation when calculating the correlation coefficient between a quantitative variable and a dichotomous variable (in which the two categories are coded as 0 and 1 respectively).

3.2.2 Sample correlation between quantitative covariates and nominal covariates which contain more than two categories

As mentioned in Section 3.2.1, the sample correlation between a quantitative covariates (say, X) and a dichotomous covariate (say, Y) can be calculated by the point-biserial correlation, which is equivalent to the Pearson correlation coefficient. However, neither the point-biserial correlation nor the classical Pearson correlation coefficient can be applied when a nominal variable contains more than two categories. This gives us the motivation to propose the definition of the sample correlation coefficient between a quantitative variable and a nominal variable which has more than two categories.

Definition 1 *Suppose X is a quantitative variable, Y is a nominal variable with K categories. Denote vector $\boldsymbol{p_i} = (p_{i1}, p_{i2}, \cdots, p_{iK})$ as the i^{th} permutation among all the possible permutations of set $\{1, 2, \cdots, K\}$, $i = 1, 2, \cdots, K!$. Denote Y^i as the variable that labels K categories of Y as $(p_{i1}, p_{i2}, \cdots, p_{iK})$ respectively, $i = 1, 2, \cdots, K!$. Then the sample correlation between X and Y is defined as:*

$$\max_{i=1,\cdots,K!} |r_{XY^i}|$$

where r_{XY^i} is the Pearson (product moment) correlation between X and Y^i.

The intuition behind this definition is that the categories of the nominal variable cannot be ranked in order, so by assuming that the "distance" between each pair of the categories is the same, we can possibly label these K categories by all the possible permutations of set $\{1, 2, \cdots, K\}$, and then find the particular permutation that can reflect the largest correlation between this nominal variable and a quantitative variable.

In order to better understand the above definition, we have the following toy example:

Suppose X is a quantitative variable with size 16:

$$X = (1, 2, 3, \cdots, 16)$$

and Y is a nominal variable with 4 categories, which is paired with X:

$$Y = (B, B, B, B, D, D, D, D, A, A, A, A, C, C, C, C).$$

From the way that X and Y are generated, it is clear that X and Y are actually (highly) correlated, because the categories of Y have some "impact" on the corresponding values of X. To be more specific, we can see that if the category of Y is "B", then the corresponding values of X tend to be small (i.e. 1, 2, 3 or 4), and if the category of Y is "C", the corresponding values of X will be large (i.e. 13, 14, 15 or 16). Since the corresponding values of X will also change a lot when the category of Y changes, we can roughly predict the values of X based on the categories of Y. This is a clear sign of correlation.

Now we follow Definition 1 to calculate the sample correlation between X and Y, and thus check whether the underlying correlation between X and Y can be captured. We should first relabel the four categories of Y by using every possible permutation of the set {1,2,3,4}, and then calculate the maximum absolute correlation coefficient between X and each of these relabeled Ys. In this example, the maximum absolute correlation coefficient is 0.97, so according to Definition 1, the correlation between X and Y is 0.97. Such a large correlation coefficient actually reflects the underlying high correlation between X and Y.

This toy example, though naive, gives us some insights of the reasoning behind Definition 1. In this definition, the reason why we need to find the maximum absolute correlation between X and each of the relabeled Ys is because not every relabeled Y is suitable for calculating this correlation, and we are trying to select the most "favorable" label for Y so that the underlying correlation between X and Y can be captured at the best chance. Again, let's look at this toy example, if we relabel Y's categories (A, B, C, D) with (1, 2, 3, 4) respectively, the absolute Pearson correlation calculated from X and this relabeled Y will be 0, which indicates no correlation between X and Y. This can also be shown in Figure 3.3(a), it is not easy to detect any correlation between X and Y. On the other hand, if we relabel the categories (A, B, C, D) by (3, 1, 4, 2), then the absolute Pearson correlation calculated is 0.97; see Figure 3.3(b), we can detect a clear pattern which indicates X and the "new" relabeled Y is highly correlated. Clearly, the label (3, 1, 4, 2) is more favorable because it help to capture the actual underlying correlation.

According to Definition 1, when the number of categories for the nominal variable is 2 (i.e. $n = 2$), the correlation coefficient is equivalent to the absolute value of the point-biserial correlation coefficient introduced in Section 3.2.1. So in the following section, the correlation between a quantitative variable and a dichotomous variable will also be calculated based on Definition 1.

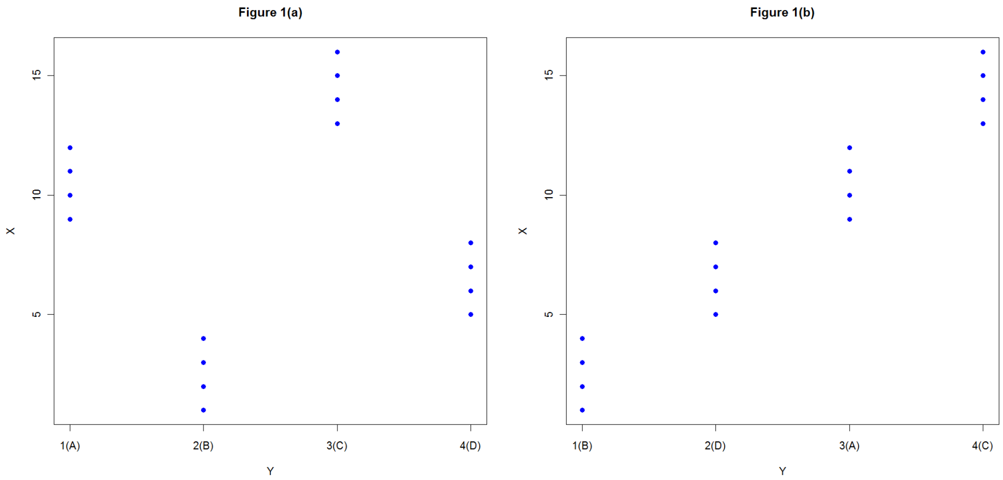


Figure 3.3: A toy example of definition 1

3.3 Sample correlation between two nominal variables

For calculating the sample correlation between two nominal variables, there are some existing methods, such as Phi and Cramer's V [1]. Essentially, these methods are all based on the contingency table and the corresponding χ^2 statistics which will be introduced later.

Cross tabulating the data in a contingency table is a usual approach of accessing the relationship between two nominal variables. A contingency table is a two-dimensional (rows and columns) table created by "crossclassifying" observations or events on two nominal variables. The rows of the contingency table are defined by the categories of one variable, and the columns are defined by the categories of the other one. The intersection of each row and column forms a cell, which contains the count (frequency) of observations (cases) that correspond to the applicable categories of both variables. Based on the contingency table, the χ^2 test can be used to assess the relationship between the two categorical (nominal) variables. The null hypothesis in such a χ^2 test is that the rows and the columns of a contingency table are independent (i.e. two nominal variables are independent). Note that under this null hypothesis, the expected values for each cell (i.e., the number of cases we would expect in each cell based on the marginal distributions of the rows and columns in the contingency table) can be easily calculated. Note that the greater the difference between the observed (O) and expected (E) cell counts, the less possible that the null hypothesis of independence is true, in other words, the stronger the evidence that the two nominal variables are related. To be more specific, suppose there are r rows and c columns in the contingency table, the expected count or frequency in the cell corresponds to the ith row and jth column, given the hypothesis of independence, is as following:

$$E_{i,j} = \frac{(\sum_{n_c=1}^{c} O_{i,n_c}) \cdot (\sum_{n_r=1}^{r} O_{n_r,j})}{N}$$

where N is the total sample size (the sum of all cells in the table), and O_{i,n_c} is the observed frequency in the cell formed by ith row and n_cth column (similar to $O_{n_r,j}$). The value of the test-statistic is

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}.$$

Note that the number of degrees of freedom is $(r-1)(c-1)$.

Based on the above description, we can see that the results of the χ^2 tests can tell us whether the two nominal variables are related or not. However, the correlation between the two variables are not provided (directly) from the χ^2 tests. In order to obtain a measure of correlation, In [1], Cramer proposed the Cramer's V as a way of calculating correlation between two nominal variables which have more than 2 categories (i.e. the contingency table contains more than 2 rows and 2 columns). Cramer's V can be used as post-test to determine strengths of association after χ^2 test has determined significance. Based on the χ^2 test-statistic introduced above, Cramer's V can be calculated by

$$V = \sqrt{(}\chi^2/(n(k-1)))$$

where χ^2 is the χ^2 test-statistic, and k is the lesser of the numbers of rows and columns in the contingency table. It has been shown that Cramer's V can take values between 0 and 1. A value of V which is close to 0 indicates little association or correlation between the two variables, on the other hand, a value close to 1 indicates a strong correlation (Cramer's V can reach 1 only when the two variables are equal to each other).

Note that if both nominal variables are dichotomous (the contingency table is 2 × 2), then the Phi correlation can be used. However, in this case, Cramer's V is equivalent to the Phi correlation, so we won't introduce the Phi correlation in details.

As we mentioned, Cramer's V (or Phi correlation) is based on the χ^2 test-statistic in the χ^2 test. However, it is known that the resulting χ^2 statistic may not be accurate if the minimum expected count for any cell in a contingency table is less than 5. Consequently, the corresponding Cramer's V may also be inaccurate. Unfortunately, it is quite common that the expected counts for some cells in a contingency table are less than 5, so in order to make the way of calculating the correlation between two nominal variables more "stable" and consistent with Definition 1, we decide to provide our own definition of the correlation between two nominal variables by following the idea of Definition 1.

Definition 2 *Suppose X is a nominal variable with K_1 categories, Y is a nominal variable with K_2 categories. Denote vector $\boldsymbol{p_i} = (p_{i1}, p_{i2}, \cdots, p_{iK_1})$ as the i^{th} permutation among all the possible permutations of set $\{1, 2, \cdots, K_1\}$, $i = 1, 2, \cdots, K_1!$. Denote X^i as the variable that labels the K_1 categories of X as $(p_{i1}, p_{i2}, \cdots, p_{iK_1})$ respectively, $i = 1, 2, \cdots, K_1!$. Similarly, denote vector $\boldsymbol{q_j} = (q_{j1}, q_{j2}, \cdots, q_{jK_2})$ as the j^{th} permutation among all the possible permutations of set $\{1, 2, \cdots, K_2\}$, $j = 1, 2, \cdots, K_2!$. Denote Y^j as the variable that labels the K_2 categories of Y as $(q_{j1}, q_{j2}, \cdots, q_{jK_2})$ respectively, $j = 1, 2, \cdots, K_2!$. Then the sample correlation between X and Y is defined* *as:*

$$\max_{i,j} |r_{X^i Y^j}|$$

where $r_{X^i Y^j}$ *is the Pearson (product moment) correlation between* X^i *and* Y^j*.*

It is clear that calculating the sample correlation between two nominal variables according to Definition 2 is very computational burdensome, especially when the number of categories in either of the nominal variables is large (say, larger than 7). We will introduce a speed-up version of Definition 2 in the next chapter, however, if in a dataset there exists one or more nominal variables that contain more that 7 categories, we tend to use Cramer's V as a quick-and-dirty approach to calculate the correlation between nominal variables. How to come up with other better approaches to finally overcome this computational issue is one of our future works.

Chapter 4

Extended Robust LARS: Robust LARS for Sequencing Both Quantitative and Nominal Variables

4.1 Extended Robust LARS

In Chapter 3, the definitions of sample correlations between different combinations of quantitative and nominal variables are introduced. Based on these definitions, we propose to form a "generalized correlation matrix" of the covariates which contains the pairwise sample correlations between each pair of the covariates (note that the pairwise correlations of two quantitative variables are calculated according to the Bivariate Windsorization method). As we introduced in Section 2.3, the sequence of covariates obtained by LARS can be derived from the correlation matrix of the data rather than using the observations themselves. Therefore, once we have the correlation matrix for both quantitative and nominal covariates, we can apply the LARS algorithm based on this generalized correlation matrix to sequence the candidate covariates. Thus, by following the LARS algorithm steps introduced in Section 2.3, we can sequence the covariates with both quantitative and nominal covariates.

As we can see, the Extended Robust LARS approach introduced above is an extension of the Robust LARS for sequencing both quantitative and nominal variables. It incorporates the sequencing that includes nominal variables by creating the "generalized correlation matrix", and also, since it inherits the Bivariate Windsorization method when calculating the correlation between quantitative variables, the Extended Robust LARS should also be robust against outliers (see the simulation results in Chapter 5 for details).

However, although our idea of the "generalized correlation matrix" enables us to extend the application of LARS to a wider scope, it produces some other problems:

- The "generalized correlation matrix" is not guaranteed to be positive definite, which does not coincide with the general property of the ordinary correlation matrix. Suppose the "generalized correlation matrix" is not positive definite, then in Step 4 of the LARS algorithm (Section 2.3), when we calculate $a = [\mathbf{1}'_{\boldsymbol{A}}(D_A R_A D_A)^{-1}\mathbf{1}_{\boldsymbol{A}}]^{-1/2}$ (note that now R_A is a subset of the "generalized correlation matrix" corresponding to the active variables), there is a possibility that we are calculating the square root of a negative number, which will cause the interruption of the algorithm. Note that because only the first few (top ranking) variables in the sequenced list are of interest, so if the algorithm stops at the point where enough active variables have been selected, we can still continue to the segmentation step based on these selected active variables. Although the above problem seldom occurs in the simulation study and the real example carried out in the later chapters, it still potentially harms the robustness of the whole algorithm. How to make the "generalized correlation matrix" positive definite is one of our future works.

- The calculation of the "generalized correlation matrix" is too computational intensive. In order to get this "generalized correlation matrix", we need to calculate the correlations between quantitative variables and nominal variables as well as the correlations between two nominal variables. According to Definition 1, in order to calculate the correlation between a quantitative variable X and a nominal variable Y which has n categories, the n categories of Y should be relabeled by every different permutation of the set $\{1, 2, \cdots, n\}$, and then the classical correlation coefficient between X and each relabeled Y should be calculated. This means we have to calculate $n!$ classical correlation coefficients (which can be very computational intensive if n is large). Similarly, in Definition 2, if we want to calculate the correlation between two nominal variables with m and k categories respectively, we need to calculate $m! \times k!$ different classical correlation coefficients, which is so computational burdensome. In the next section, we will propose some approaches to speed up the calculation of the "generalized correlation matrix".

4.2 Approaches to speed up the robustified LARS

As mentioned in the previous section, the Extended Robust LARS can be a quite computational intensive algorithm due to the calculation of the "generalized correlation matrix". In this section, we introduce some approaches to speed up the calculation of the pairwise correlations between "a quantitative variable and a nominal variable" and "two nominal variables" respectively.

4.2.1 Speed-up sample correlation between a quantitative variable and a nominal variable

In order to speed up the correlation calculation in Definition 1, we propose the following approach:

Suppose X is a quantitative variable, Y is a nominal variable with n categories. Denote the medians of X values corresponding to different Y categories as $\texttt{Med}_1, \cdots, \texttt{Med}_n$. Then relabel the category of Y that corresponds to the smallest $\texttt{Med}$ among $\texttt{Med}_1, \cdots, \texttt{Med}_n$ as 1 (numeric); relabel the category of Y that corresponds to the second smallest $\texttt{Med}$ among $\texttt{Med}_1, \cdots, \texttt{Med}_n$ as 2, and so on. Denote such a relabeled Y as Y_0, then the speed-up sample correlation between X and Y can be calculated as r_{XY_0}, where r_{XY_0} is the

Pearson (product moment) correlation between X and Y_0.

By using this speed-up correlation between a quantitative variable and a nominal variable instead of the correlation in Definition 1, we can make our Extended Robust LARS less computational intensive than before.

4.2.2 Speed-up sample correlation between two nominal variables

As mentioned, the correlation calculation defined in Definition 2 is too computational burdensome and consequentially, time-consuming. In order to speed up the calculation, we propose the following approach to reduce some unnecessary permutations when relabeling the categorical variable in Definition 2.

Suppose we want to calculate the sample correlation between nominal variable X with m categories and nominal variable Y with n categories. Instead of calculate the classical correlation coefficients between every possible combination of relabeled Xs and relabeled Ys, we will use the following steps:

1. Relabel one nominal variable (say, X) with a random permutation from the set $\{1, 2, \cdots, m\}$, and denote this relabeled X as X_0.

2. Fix X_0, then relabel the n categories of Y by each of different permutations from the set $\{1, 2, \cdots, n\}$. Calculate the absolute Pearson correlation coefficient between X_0 and each of the relabeled Ys, and

then find out the relabeled Y which maximizes the Pearson correlation with X_0. Denote this relabeled Y as Y_0 and record the maximum correlation coefficient as r_{max}.

(Note: Suppose $\boldsymbol{p} = (p_1, p_2, \cdots, p_n)$ is one permutation among al-l the possible permutations of set $\{1, 2, \cdots, n\}$. We will consider $\boldsymbol{p^R} = (p_n, p_{(n-1)}, \cdots, p_1)$ (i.e. the reverse of $\boldsymbol{p}$) as a replicate of $\boldsymbol{p}$ in the context of relabeling the categories of a nominal variable. This is because no matter which one of $\boldsymbol{p}$ and $\boldsymbol{p^R}$ is used to relabel the categories of Y, we will get the same absolute correlation coefficient when calculating the Pearson correlation between X and the relabeled Y. Thus, by "different permutations", we mean all the possible permutations with the replicate ones excluded.)

3. Fix Y_0, then relabel the m categories of X by each of different permutations from set $\{1, 2, \cdots, m\}$. Calculate the absolute Pearson correlation coefficient between Y_0 and each of the relabeled Xs, and then find out the relabeled X which maximizes the Pearson correlation with Y_0. Denote this relabeled X as X_0 and record the maximum correlation coefficient as $r^{'}_{max}$.

4. Repeat Step 2 and 3 until the difference between r_{max} and $r^{'}_{max}$ is less than a small value chosen by the user (say, 0.0001). Then the speed-up sample correlation between two nominal variables X and Y can be defined as $\max\{r_{max}, r^{'}_{max}\}$.

As we can see, such a speed-up method introduced above does not guarantee that $\max\{r_{max}, r^{'}_{max}\}$ is the global maximum among all possible combinations of $m! \times n!$ permutations. It highly depends on the start point we choose (the initial labels used in the first step). So in the following applications in this thesis (the simulation study and real example), we will start the above steps with two different initial points to increase the chance of reaching the global maximum correlation coefficient.

Although the above speed-up approach enables us to avoid many unnecessary permutations when relabeling both nominal variables and thus speed up the correlation calculation, it still will be too time consuming when the number of categories in either of the nominal variables is large. As we mentioned in Section 3.3, we still prefer to use the Cramer's V as a quick-and-dirty approach to approximate the correlation between two nominal variables when the nominal variables contain large number of categories.

Chapter 5

Simulation Study

To check the performance of the Extended Robust LARS which uses the speed-up correlations, a simulation study similar to [5] is carried out.

In this simulation study, the total number of the candidate covariates is 50, in which 9 are "important covariates" or "target variables" (i.e. the covariates that are actually related to the response variable). Three different cases according to the different correlation structures among the target covariates will be considered, and in each case, the performances of the Extended Robust LARS and its competitor methods: Forward Selection (FS) and Group Lasso (GrpLasso) will be compared.

Case 1: independent target variables (i.e. the true correlation between each pair of target covariates is 0)

In this case, the simulation is carried out by following the steps listed below:

1. Generate the candidate variables x_i $(i = 1, 2, \cdots, 50)$ (with size $n = 150$) independently from a standard normal distribution $N(0, 1)$.

2. Generate the response variable y using the following linear model:

$$y = 7(x_1 + x_2 + x_3) + 6(x_4 + x_5 + x_6) + 5(x_7 + x_8 + x_9) + \epsilon$$

   Therefore, $x_1, x_2, \cdots, x_9$ are considered to be the target covariates. The variance of the error term $Var(\epsilon)$ is chosen such that the signal-to-noise ratio is equal to 2.

3. Convert 5 quantitative covariates into nominal variables.

   The nominal covariates are converted from the quantitative covariates. We convert x_3 and x_4, which are target covariates, into nominal covariates with three categories. The largest 50 values of the covariate (e.g. x_3 or x_4) are stratified as the first category, say "A", and the smallest 50 values are in the third category, say "C". Then rest 50 values (50 moderate values) are grouped as the second category, say "B". Similarly, we convert x_{28}, x_{29} and x_{30}, which are "irrelevant" covariates (i.e. covariates that are actually independent from the response y), into nominal covariates with 2, 3 and 4 categories respectively.

4. Certain levels of asymmetric, shifted normal contaminations are applied to the response variable y by using the noise term ϵ with large positive means. We consider three different levels of contamination: 0%, 5% (i.e. 5% of the y values are simulated using the error terms with large positive means) and 10%.

In order to compare the performances of our robustified LARS and F-S, we will repeat the above simulation steps 1000 times. For each of the three contamination levels in a simulation run, the candidate variables will be sequenced by our robustified LARS (for both quantitative and nominal variables) and the classical sequencing algorithm Forward Selection (FS), and thus, we can get the sequenced lists of the covariates from both procedures. We also compare the sequences from our robustified LARS with those generated by the Group Lasso. The Group Lasso is an extension of the Lasso, which can be used for selecting the grouped variables. Once we have nominal variables as candidate predictors, the Group Lasso will be used instead of the regular Lasso. This is because in methods such as Lasso and FS, a collection of indicator (dummy) variables are used for representing the levels of a categorical (nominal) variable, which means these dummy variables are actually grouped together. Thus, they should be chosen as a group in the variable selection procedure. However, the regular Lasso only works well for the variables which can be treated individually. When the variables are grouped, the lasso does not work well. So in our case, the Group Lasso (instead of the regular Lasso) will be used as a competitor with our robustified LARS for sequencing both the quantitative and nominal candidate variables because it treats and selects the dummy variables, which corresponds to the same nominal variable, as a group. It is also worth noticing that when each of the factors only corresponds to one measured variable, then the Group Lasso reduces to the regular Lasso.

Then the performances of the robustified LARS (RobLARS), the Forward Selection (FS) and the Group Lasso (GrpLasso) can be compared based on the number of target (or important) variables (i.e. t_m) included in the first m sequenced variables. Larger number of the target variables that appear at the top (i.e. first m) of the sequenced list indicates better performance of the corresponding sequencing algorithm. Figure 5.1 shows the average (over the 1000 simulation runs) of t_m for each of the sequencing methods (i.e. RobLARS, FS and GrpLasso) and contamination levels (i.e. 0%, 5%, 10%).

From Figure 5.1 (a), we can see that if the covariates are uncorrelated and uncontaminated, the performances of the FS and GrpLasso are quite similar to that of our RobLARS when the number of variables m is small (say, smaller than 10). When m gets larger, the FS and GrpLasso have slightly better performances than the RobLARS. However, in general, all the three procedures perform reasonably well. Figure 5.1 (b)-(c) show that, the performances of the FS and GrpLasso considerably deteriorates under contamination, while our RobLARS procedure is less affected by contamination.

Case 2: the correlations between the important covariates are large

In this case, we generate the data sets by following the steps listed below:

1. Generate the latent variables L_i $(i = 1, 2, 3)$ which are independently standard normal distributed ($L_i \sim N(0, 1)$) with size $n = 150$.

2. Generate the response variable y based on the latent variables:

$$y = 5L_1 + 4L_2 + 3L_3 + \epsilon = \texttt{Signal} + \epsilon$$

   where ϵ is a normal error and is independent from the latent variables. We want to ensure that the signal-to-noise ratio is equal to 2, so the variance of ϵ is chosen to be $\texttt{Var}(\epsilon) = 50/4$.

3. Generate the target (or important) covariates x_j, $(j = 1, 2, \cdots, 9)$ according to the latent variables:

   - For $j = 1, 2, 3$, $x_j = L_1 + \delta_j$
   - For $j = 4, 5, 6$, $x_j = L_2 + \delta_j$
   - For $j = 7, 8, 9$, $x_j = L_3 + \delta_j$

   where $\delta_j \sim N(0, \sigma_j)$ $(j = 1, 2, \cdots, 9)$. It is easy to see that the response variable y and the target variables $x_1, x_2, \cdots, x_9$ are linked through the latent variables L_1, L_2 and L_3. The correlation structure of the target variables is determined by the values of σ_j. In this case, we choose the values of σ_j such that the true correlation between each two covariates generated with the same latent variable is 0.9.

4. Generate each of the "irrelevant" covariates x_j $(j = 10, 11, \cdots, 30)$ independently from a standard normal distribution $N(0, 1)$ (size $n = 150$).

5. Similar to Step 3 in Case 1, convert several quantitative covariates (e.g. $x_3, x_4, x_{28}, x_{29}, x_{30}$) into nominal variables.

6. Similar to Step 4 in Case 1, add outliers to the response variable y. In this case, we also consider three different levels of contamination: 0%, 5% and 10%.

Similar to Case 1, we will repeat the above simulation steps 1000 times. For each of the three contamination levels in a simulation run, the candidate variables will be sequenced by our Extended Robust LARS, the Forward Selection and the Group Lasso. Again, the performances of the Extended Robust LARS (E-RobLARS), the Forward Selection (FS) and the Group Lasso (GrpLasso) can be compared based on the number of target variables (i.e. t_m) included in the first m sequenced variables. Figure 5.2 shows the average (over the 1000 simulation runs) of t_m for each of the sequencing methods (i.e. E-RobLARS, FS and GrpLasso) and contamination levels (i.e. 0%, 5%, 10%).

From Figure 5.2 (a)-(c), we can see that if the candidate covariates are highly correlated, the performances of the FS and GrpLasso are not comparable to that of our E-RobLARS. Also, the performances of the FS and GrpLasso deteriorate under contamination, while the E-RobLARS procedure is less affected by contamination.

Case 3: the correlations between the important covariates are moderate

In this case, the simulation steps are quite similar to those in Case 2. However, when generating the important covariates (Step 3), we choose the values of σ_j such that the true correlation between the covariates generated with the same latent variable is 0.5 (instead of 0.9 in Case 2).

After we run the simulation 1000 times, the performances of the E-RobLARS, the FS and the GrpLasso are shown in Figure 5.3.

From Figure 5.3 (a)-(c), we can see that if the candidate covariates are moderately correlated, the performances of the FS and GrpLasso are also not comparable to that of our E-RobLARS and the E-RobLARS procedure is less affected by contamination comparing to the FS and GrpLasso.

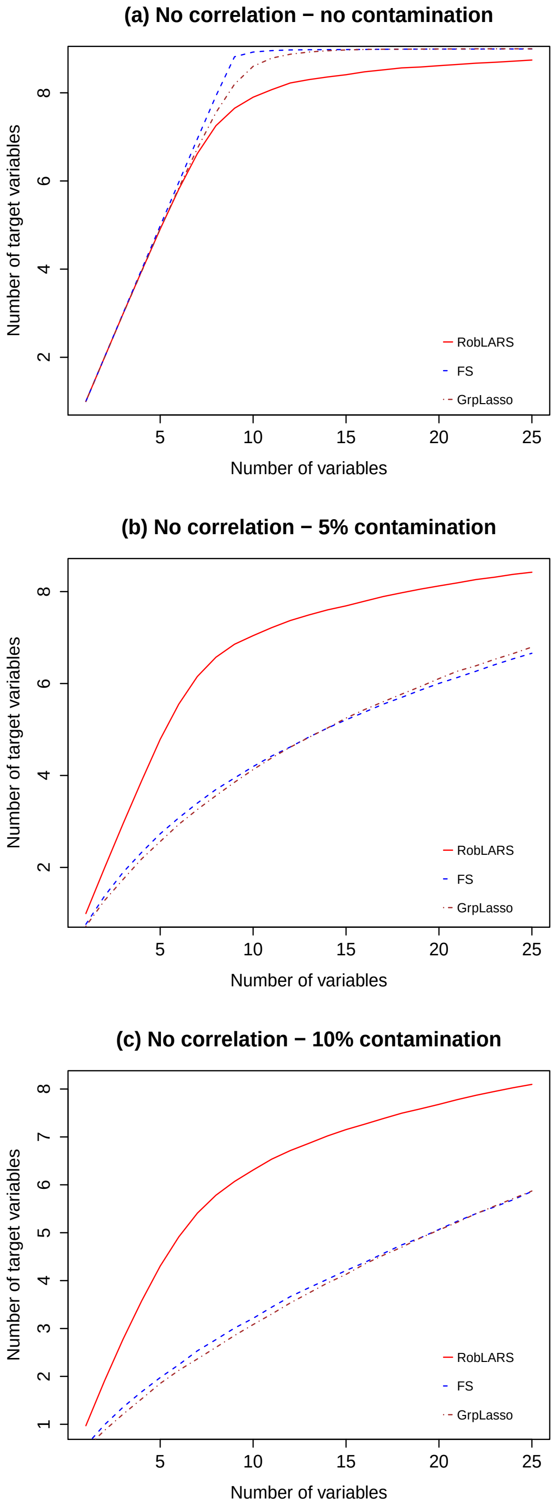


Figure 5.1: Case 1, the numbers of the important covariates that have the top (ten) rankings in the LARS lists and the FS lists

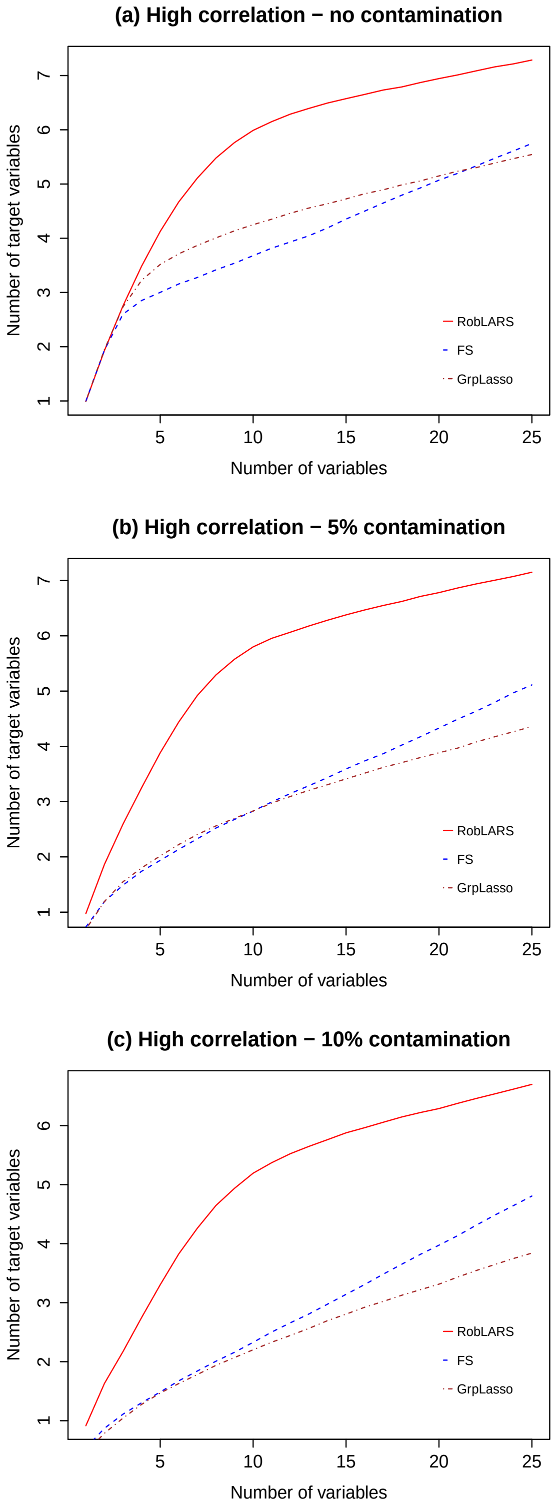


Figure 5.2: Case 2, the numbers of the important covariates that have the top (ten) rankings in the LARS lists and the FS lists

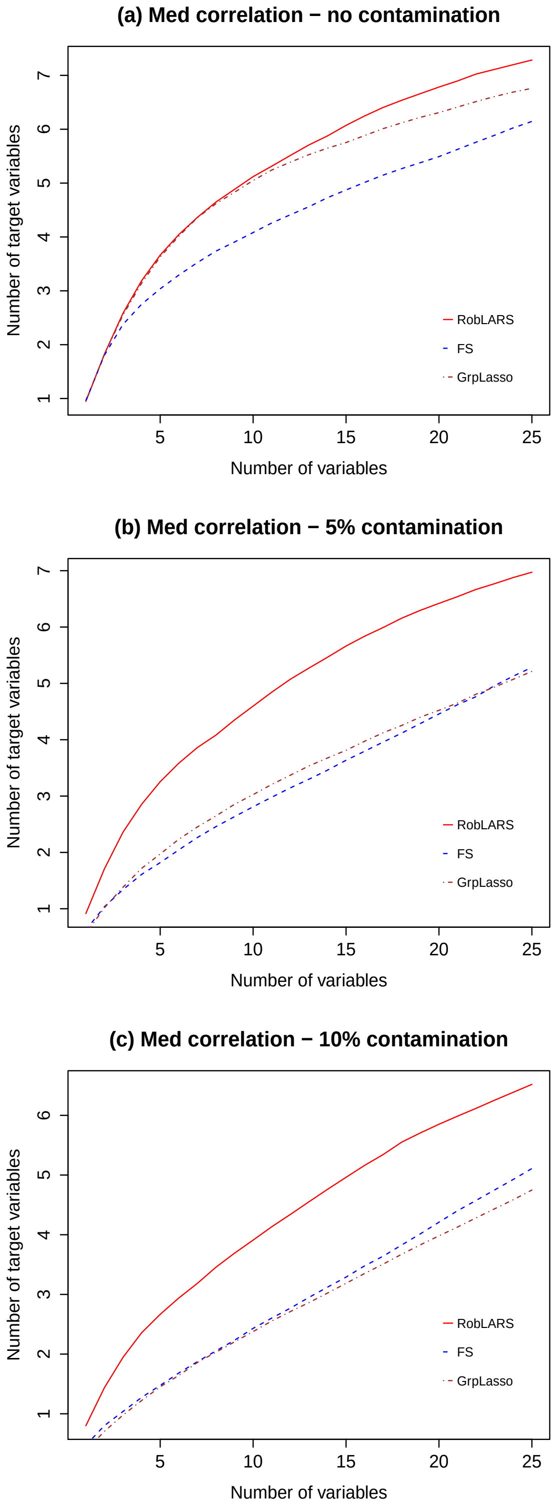


Figure 5.3: Case 3, the numbers of the important covariates that have the top (ten) rankings in the LARS lists and the FS lists

Chapter 6

Applications

6.1 Real data

In this section, the Extended Robust LARS is applied to the real datasets and its performances are evaluated and compared to its competitor methods: the Forward Stepwise (FS) and the Group Lasso (GrpLasso).

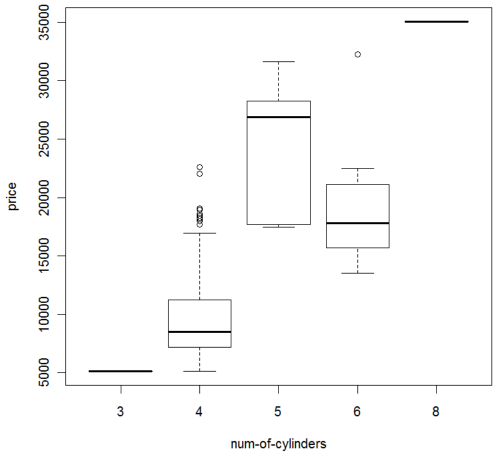
6.1.1 The auto imports data

The auto imports data, which is created by Jeffrey C. Schlimmer in 1987, contains the information of autos in terms of various characteristics. The response variable of interest is the price of the autos ("price"). Besides the response variable, this dataset consists of 21 variables, including 12 quantitative variables and 9 nominal variables. After the elimination of the observations which contain missing values, there are 196 observations left.

We will first sequence the covariates by the Extended Robust LARS, and then compare the list of sequenced covariates to those generated by the Forward Stepwise (FS) and the Group Lasso (GrpLasso) method.

After carrying out some exploratory data analysis, we find that the logarithm transformation should be applied to the response variable in order to make the response and predictor variables more linearly related. Also, it is noticed that one of the nominal variables "num-of-cylinders" can be considered as either a quantitative variable or a nominal variable, because its categories such as "three" or "five" can be transformed to integers. Since the sequenced list of the covariates generated by the Extended Robust LARS differ between the cases of nominal "num-of-cylinders" and integer "num-of-cylinders", we should take a close look at this variable. We first explore the relationship between "num-of-cylinders" and the response variable "price". The boxplot of "price" against "num-of-cylinders" is shown in Figure 6.1.1

Figure 6.1: Boxplot of the response variable "*price*" against "$num - of - cylinders$"



From Figure 6.1.1, we can see that the "price" tends to be quite different as the categories of "num-of-cylinders" differ, which indicates a quite strong association between these two variables. If we transform "num-of-cylinders" to a quantitative variable, the correlation coefficient between "num-of-cylinders" and "price" calculated by the Bivariate Windsorization method is 0.583, which does not seem to reflect the underlying strong association. Note that although the "price" tends to increase along with "num-of-cylinders", it is not the case for the cars with 5 cylinders and 6 cylinders in their engines. Generally, the cars with 5 cylinders are more expensive than the cars with 6 cylinders. So if we keep "num-of-cylinders" as a nominal variable and apply the correlation defined by Definition 1, we may be able to capture the underlying association better. Actually, the correlation coefficient calculated according to Definition 1 is 0.717, which shows a stronger relationship between "num-of-cylinders" and "price" (compared to the case when "num-of-cylinders" is transformed to quantitative variable). Since we can capture the correlation between the response variable and "num-of-cylinders" much better if we keep "num-of-cylinders" as a nominal variable, we will consider it nominal from now on.

We implement the Extended Robust LARS (E-RLARS) in the R package for sequencing both the quantitative and nominal candidate covariates. In this example, the sequenced list of the covariates obtained is as follows:

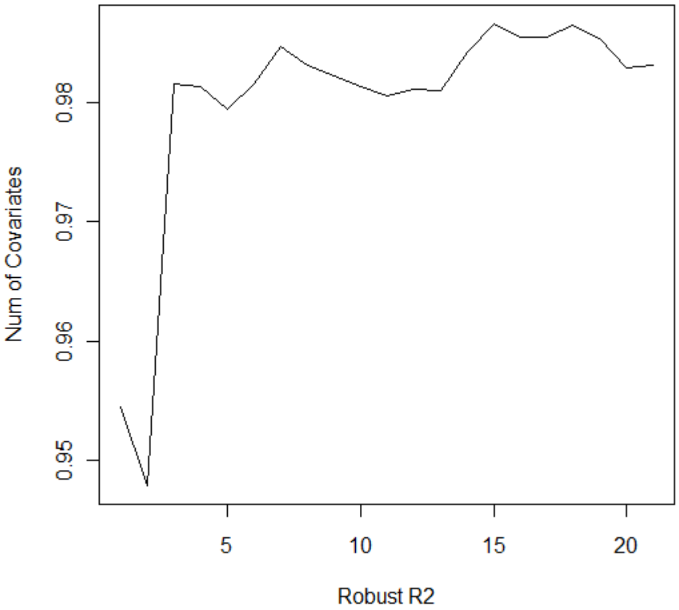
```
 [1] "curb-weight"       "horsepower"        "make"              "highway-mpg"
 [5] "drive-wheels"      "num-of-cylinders"  "fuel-system"       "symboling"
 [9] "height"            "width"             "peak-rpm"          "num-of-doors"
[13] "engine-size"       "engine-type"       "city-mpg"          "aspiration"
[17] "fuel-type"         "length"            "body-style"        "compression-ratio"
[21] "wheel-base"
```

Based on the sequenced list generated from E-RLARS, we can first generate the corresponding "reduced set" which includes the first m top ranking covariates, and then try to select the prediction model based on these m covariates in the reduced set. However, in most cases m (i.e. the number of covariates needed in the model) is unknown. In order to determine it, we use a graphical tool called *learning curve* [5]. The learning curve can be obtained as follows: We first fit a robust regression model with only the first covariate in the sequenced list as predictor, and then we add another covariate in the model (one variable a time) by following the orders of the covariates in the sequenced list. Each time we increase the number of variables (along the sequence), we fit a robust regression model to calculate a robust R^2 measure, e.g. $R^2 = 1 - Median(e^2)/MAD^2(Y)$, where e is the vector of residuals obtained from the corresponding robust fit (see Rousseeuw and Leroy 1987). Then the learning curve is obtained by plotting these robust R^2 values against the number of variables in the model. The size of the reduced set, m, can be chosen as the point where the learning curve does not have a considerable (increasing) slope anymore. Figure 6.1.1 shows the learning curve for this dataset based on E-RLARS.

Figure 6.1.1 suggests a reduced set of size 3 (the Robust R^2 is over 0.98 and the slope is not considerable after $m = 3$), which includes the following covariates ("$curb - weight$", "$horsepower$", "$make$"). So the model selected in this case, called E-RLARS model, has the following 3 covariates ("$curb - weight$", "$horsepower$", "$make$").

The Forward Stepwise (FS) procedure, which can also be used to se-

Figure 6.2: Learning curve for auto imports data based on E-RLARS



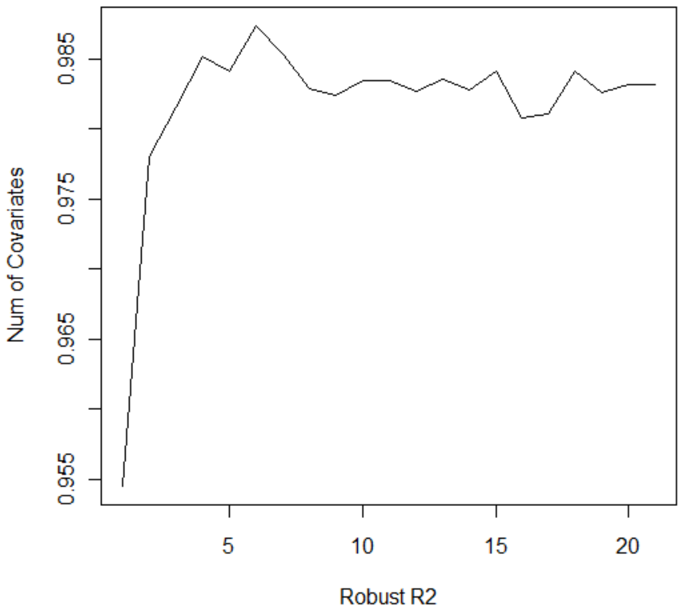
quence the covariates in this dataset, generates the sequenced list as follows:

```
 [1] "curb-weight"       "make"              "horsepower"        "body-style"
 [5] "fuel-system"       "engine-type"       "height"            "wheel-base"
 [9] "compression-ratio" "aspiration"        "num-of-cylinders"  "drive-wheels"
[13] "num-of-doors"      "length"            "city-mpg"          "highway-mpg"
[17] "width"             "symboling"         "engine-size"       "peak-rpm"
[21] "fuel-type"
```

Figure 6.1.1 shows the learning curve for Auto Imports data based on FS, and it suggests a reduced set of at most size 6. We applied all subsets selection to these 6 variables using 2-fold cross validation. The model selected in this case, called FS CV-model, has the following 5 covariates:

(“$curb-weight$”, “$make$”, “$horsepower$”, “$body-style$”, “$fuel-system$”).

Figure 6.3: Learning curve for auto imports data based on FS

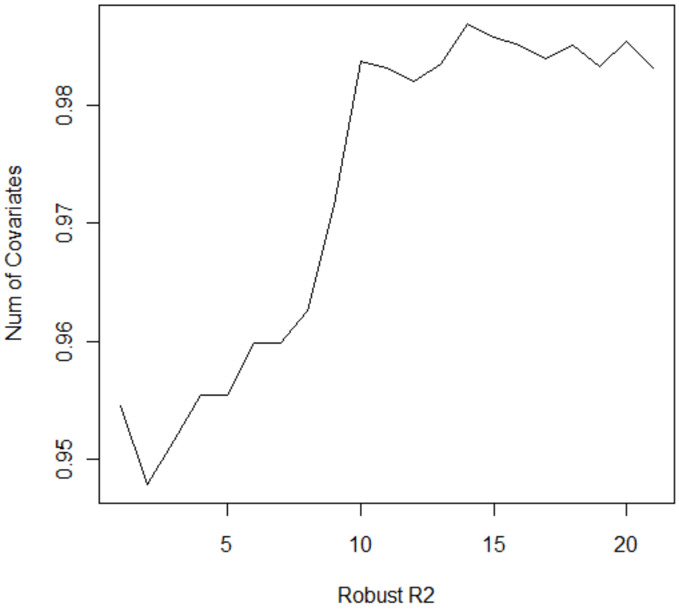


As introduced before, another approach, the Group Lasso (GrpLasso), can also be used to sequence the covariates that contain both quantitative and nominal variables. In this example, the sequenced list generated by GrpLasso is as below:

```
 [1] "curb-weight"       "horsepower"        "drive-wheels"      "highway-mpg"
 [5] "length"            "width"             "peak-rpm"          "body-style"
 [9] "compression-ratio" "engine-size"       "fuel-system"       "engine-type"
[13] "num-of-cylinders"  "make"              "fuel-type"         "aspiration"
[17] "num-of-doors"      "symboling"         "wheel-base"        "height"
[21] "city-mpg"
```

Figure 6.1.1 shows the learning curve for Auto Imports data based on GrpLasso, and it suggests a reduced set of at most size 10. Again, we applied all subsets selection to these 10 variables using 2-fold cross validation. The model selected in this case, called GrpLasso CV-model, has the following 6 covariates: (“$curb-weight$”, “$horsepower$”, “$city-mpg$”, “$compression-ratio$”, “$body-style$”, “$make$”).

Figure 6.4: Learning curve for auto imports data based on GrpLasso



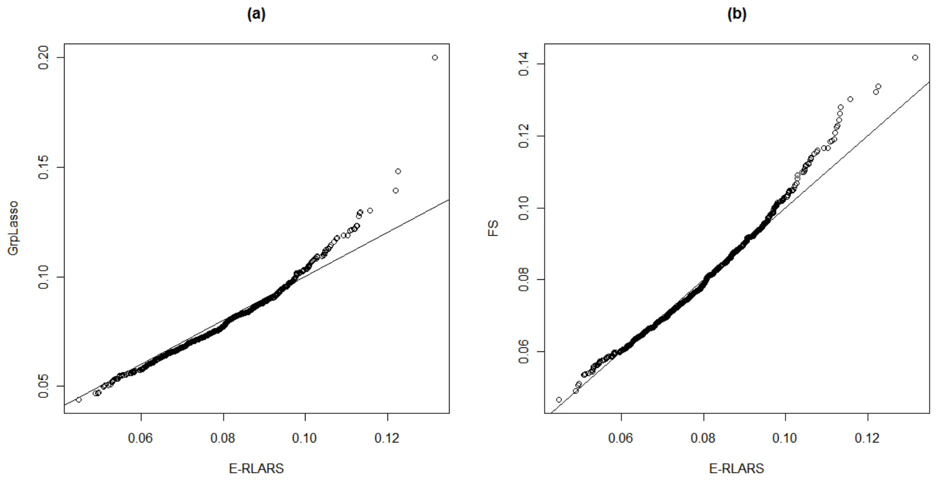
To compare the models selected by these three different procedures, we estimated the mean squared prediction error (MSPE) for each of these three models 1000 times using 2-fold CV. The averages and the standard deviations of these 1000 CV-MSPEs are shown in Table 6.1.

Model	Avag(CV-MSPEs)	SD(CV-MSPEs)
E-RLARS	0.0789	0.01213
FS	0.0789	0.01323
GrpLasso	0.0777	0.01426

Table 6.1: Averages and standard deviations of CV-MSPE in the auto import data, obtained by the E-RLARS, FS CV and GrpLasso CV models

From this table we can see that the three approaches have almost the same average CV-MSPE. However, the standard deviations provided indicate that E-RLARS model yields the least variable CV-MSPEs. The Q-Q plots of the CV-MSPEs for E-RLARS and GrpLasso as well as for E-RLARS and FS are shown in Figure 6.1.1.

Figure 6.5: Q-Q plots of CV-MSPEs (a) E-RLARS vs. GrpLasso (b) E-RLARS vs. FS



We can see that almost all the points in Figure 6.1.1 lie above the line $y = x$, which shows that generally, the CV-MSPEs from E-RLARS are distributed on the left (smaller) side of those from either GrpLasso or FS. Also, the E-RLARS model contains only 3 variables as predictors, while the FS CV-model and GrpLasso CV-model contain 5 and 7 predictors respectively. Hence, the E-RLARS procedure clearly managed to identify the three most important predictors among the 21 candidate variables.

Chapter 7

Conclusion

In this thesis, we focus on the *sequencing* step of the two-step model building procedure. The goal of the sequencing step is to rank the covariates in order of importance, and then we can pick the first m selected candidate covariates to further build a final prediction or explanatory model. Least Angle Regression (LARS) is a powerful algorithm that can be used to sequence the covariates, however, in [5] Khan et al. pointed out that when used in the sequence step, LARS is not robust against outliers, so they proposed the Robust LARS to remedy this problem. Our work can be considered as a continuation of [5]. We further robustified the Robust LARS and propose the Extended Robust LARS by introducing a "generalized correlation matrix". While Robust LARS can only sequence the variables that are quantitative, the Extended Robust LARS is applicable when a dataset contains both quantitative and nominal variables. Our simulation study shows that comparing to its competitors such as Group Lasso and Forward Selection, the Extended Robust LARS works well (i.e. tends to rank the important variables at the top of the sequenced list) when we have both quantitative and nominal variables and it is quite robust against outliers. For the future works, we will try to further refine the definition of the generalized correlation matrix so it can possess the property of “positive definite”, also, we need to further speed up our algorithm so it can be less computational burdensome.

Bibliography

[1] Cramer, H. 1999. *Mathematical methods of statistics*, volume 9. Princeton Univ Pr.

[2] Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. 2004. Least angle regression. *The Annals of statistics*, 32(2):407–499.

[3] Friedman, J., Hastie, T., and Tibshirani, R. 2001. *The elements of statistical learning*, volume 1. Springer Series in Statistics.

[4] Huber, P., Ronchetti, E., and MyiLibrary 1981. *Robust statistics*, volume 1. Wiley Online Library.

[5] Khan, J., Van Aelst, S., and Zamar, R. 2007. Robust linear model selection based on least angle regression. *Journal of the American Statistical Association*, 102(480):1289–1299.

[6] Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B Methodological*, 58(1):267–288.

[7] Yuan, M. and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67.