Robust Linear Model Selection Based on Least Angle Regression — **Source link** ↗

Jafar A. Khan, Stefan Van Aelst, Ruben H. Zamar

Institutions: University of British Columbia, Ghent University

Related papers:

- Regression Shrinkage and Selection via the Lasso

- Least angle regression

- Robust Statistics: Theory and Methods

- Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties

- Least Median of Squares Regression

Robust Linear Model Selection Based on

Least Angle Regression

Jafar A. Khan, Stefan Van Aelst, and Ruben H. Zamar [*]

June 13, 2007

Abstract

In this paper we consider the problem of building a linear prediction model
when the number of candidate predictors is large and the data possibly contains
anomalies that are difficult to visualize and clean. We aim at predicting the non-
outlying cases. Therefore, we need a method that is robust and scalable at the
same time. We consider the stepwise algorithm LARS which is computationally
very efficient but sensitive to outliers. We introduce two different approaches
to robustify LARS. The *plug-in* approach replaces the classical correlations in
LARS by robust correlation estimates. The *cleaning* approach first transforms
the dataset by shrinking the outliers toward the bulk of the data (which we
call *multivariate Winsorization*) and then applies LARS to the transformed
data. We show that the plug-in approach is time-efficient and scalable and

1

that the bootstrap can be used to stabilize its results. We recommend the use of bootstrapped robustified LARS to sequence a number of candidate predictors to form a *reduced set* from which a more refined model can be selected.

KEY WORDS: Stepwise algorithm; Robust prediction; Computational complexity; Winsorization; Bootstrap.

1 INTRODUCTION

Robust model selection has not received much attention in the robustness literature. Most of the papers related to robust model selection in regression have focused on the development of robust selection criteria that can be used to compare models. Seminal papers that address this issue include Ronchetti (1985) and Ronchetti and Staudte (1994) which introduced robust versions of the selection criteria AIC and C_p, respectively. Maronna, Martin and Yohai (2006) proposed a robust Final Prediction Error (FPE) criterion (see also Splus documentation) while Müller and Welsh (2005) proposed a robust selection criterion based on a stratified bootstrap procedure. Robust selection criteria for more general models have been developed by Cantoni and Ronchetti (2001) for generalized linear models and Ronchetti and Trojani (2001) for generalized method of moments. In the latter context model selection can make use of indirect inference (see Genton and Ronchetti 2003; Jiang and Turnbull 2004). Atkinson and Riani (2002) proposed an added-variable t-test for variable selection in the context of regression based on the forward search procedure. Morgenthaler, Welsch, and Zenide (2003) constructed a selection technique to simultaneously identify the correct model structure as well as unusual observations. Ronchetti, Field, and Blanchard (1997) proposed robust model selection by cross-validation.

A major drawback of most robust model selection methods is that they are very

time consuming, as they require the robust fitting of a large number of submodels. One exception is a model selection procedure based on the Wald test (Sommer and Huggins 1996) which requires the computation of estimates only from the full model. However, our purpose is to develop a procedure that can handle a large number d of possible predictors - e.g. several hundreds or even thousands of candidate predictors. In such cases a robust fit of the 'full model' may not be feasible due to the numerical complexity of robust estimates when d is very large (e.g. $d \geq 200$) or simply because d exceeds the number of cases, n.

Our model selection strategy proceeds in two steps. The first step - which we call *sequencing* - quickly screens out unimportant variables to form a "reduced set" for further consideration. Thus, the goal of the first step is a drastic reduction of the number of candidate predictors. The input variables are sequenced to form a list such that the good predictors appear in the beginning. The first m variables of the list then form the reduced set from which the prediction model will be obtained. The second step - which we call *segmentation* - carefully examines the predictors in the reduced set for possible inclusion in the prediction model. For the segmentation in the second step, the aforementioned robust selection techniques can be used because the set of candidate predictors has been reduced to a feasible size.

In this paper we focus on sequencing the candidate predictors in order of importance. One strategy for sequencing the candidate predictors is to use one of the several available stepwise or stagewise procedures such as forward selection (see, e.g. Weisberg 1985, chap. 8) or stagewise forward selection (SFS) (see Hastie, Tibshirani, and Friedman 2001, chap. 10). We focus on a powerful technique recently proposed by Efron, Hastie, Johnstone, and Tibshirani (2004) called least angle regression (LARS) which is computationally very efficient. We show that LARS is based on sample means, variances and correlations. Therefore, it is very fast to compute but yields

poor results when the data is contaminated. This is a potentially serious deficiency. To remedy this, we propose several approaches to strengthen the robustness properties of LARS without affecting its computational efficiency too much and compare their behavior.

Note that affine equivariance and regression equivariance are generally considered to be important properties for (robust) regression estimators. However, these properties are not required in the context of variable selection, because we do not consider general linear transformations of the given covariates. The only transformations that should not affect the selection result are linear transformations of individual variables, i.e., shifts and scale changes. Hence, variable selection methods (e.g. LARS) are often based on correlations among the variables. Therefore, robust variable selection procedures need to be robust against correlation outliers, that is, outliers that affect the classical correlation estimates but can not be detected by looking at the individual variables separately. Our approaches are based on robust correlations estimates. Hence, they are robust against correlation outliers and thus suitable for robust variable selection.

The rest of the paper is organized as follows. In Section 2 we show that LARS can be expressed in terms of the correlation matrix of the data. In Section 3, we illustrate LARS' sensitivity to outliers and introduce two different approaches to robustify LARS. Section 4 presents the results of a simulation study that compares the performance (and computing efficiency) of LARS and its robust alternatives. In this section we also compare the sequences produced by our robust proposals with the sequences produced by random forests (Breiman 2001). In Section 5 we propose to use bootstrap to improve and stabilize the results obtained by robust LARS. In Section 6 we give two real-data applications and compare the results of robust LARS with those of random forests and multiple support vector machine

4

recursive feature elimination (MSVM-RFE), proposed by Duan and Rajapakse (2005) in the context of classification. Section 7 concludes and the Appendix contains some technical derivations.

2   LEAST ANGLE REGRESSION

Efron et al. (2004) proposed Least Angle Regression which is closely related to SFS and LASSO (Tibshirani 1996) procedures. LARS provides an ordering in which the covariates enter a regression model. This sequence is usually the same as in LASSO or SFS but obtained in a computationally efficient way.

The SFS procedure enters variables in small steps in the regression model to prevent correlated predictors from being excluded from the top of the sequence. However, this method often becomes time consuming due to the fact that often a large number of small steps are taken in the direction of the same variable. LARS solves this problem by analytically determining the optimal step size for each variable.

Another convenient feature of LARS is that the resulting sequence of the covariates can be derived from the correlation matrix of the data (without the observations themselves). Let $\mathbf{Y}, \mathbf{X}_1, \ldots, \mathbf{X}_d$ be the variables, standardized using their mean and standard deviation. Let r_j denote the correlation between $\mathbf{X}_j$ and $\mathbf{Y}$, and let $\mathbf{R_X}$ be the correlation matrix of the covariates $\mathbf{X}_1, \ldots, \mathbf{X}_d$. Suppose that $\mathbf{X}_m$ has the maximum absolute correlation r with $\mathbf{Y}$ and denote $s_m = \mathrm{sign}(r_m)$. Then, $\mathbf{X}_m$ becomes the first *active variable* and the initial fit $\hat{\mathbf{Y}} = \mathbf{0}$ should be modified by moving along the direction of $s_m \mathbf{X}_m$ a certain distance γ that can be expressed in terms of the correlations between the variables (see Appendix A for details). By determining γ, LARS simultaneously identifies the new covariate that enters the model, that is, the second active variable.

As soon as we have more than one active variable, LARS modifies the current fit $\hat{\mathbf{Y}}$ along the *equiangular direction* which is the direction that has equal angle (correlation) with all active covariates. Moving along this direction ensures that the correlation of each active covariate with the residual decreases equally. Let A be the set of subscripts corresponding to the active variables. In Appendix B the standardized equiangular vector $\mathbf{B}_A$ is derived. Note that we do not need the direction $\mathbf{B}_A$ itself to decide which covariate enters the model next. We only need the correlation of all variables (active and inactive) with $\mathbf{B}_A$. These correlations can be expressed in terms of the correlation matrix of the variables as shown in Appendix B. LARS modifies the current fit by moving along $\mathbf{B}_A$ up to a certain distance γ_A which, again, can be determined from the correlations of the variables (see Appendix C).

We now summarize the LARS algorithm in terms of correlations r_j between $\mathbf{X}_j$ and $\mathbf{Y}$, and the correlation matrix $\mathbf{R_X}$ of the covariates:

1. Set the active set, $A = \emptyset$, and the sign vector $\boldsymbol{s}_A = \emptyset$.

2. Determine $m = \underset{j}{\operatorname{argmax}} |r_j|$, and $s_m = \operatorname{sign}\{r_m\}$. Let $r = s_m r_m$.

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

4. Let $\mathbf{R}_A$ be the submatrix of $\mathbf{R_X}$ corresponding to the active variables. If $\det \mathbf{R}_A = 0$ then stop. Otherwise, calculate $a = [\mathbf{1}'_A (\mathbf{D}_A \mathbf{R}_A \mathbf{D}_A)^{-1} \mathbf{1}_A]^{-1/2}$, where $\mathbf{1}_A$ is a vector of 1's and $\mathbf{D}_A = \operatorname{diag}(\boldsymbol{s}_A)$. Calculate $\boldsymbol{w}_A = a (\mathbf{D}_A \mathbf{R}_A \mathbf{D}_A)^{-1} \mathbf{1}_A$, and $a_j = (\mathbf{D}_A \boldsymbol{r}_{jA})' \boldsymbol{w}_A$, for $j \in A^c$, where $\boldsymbol{r}_{jA}$ is the vector of correlations between $\mathbf{X}_j$ and the active variables. (Note that, when there is only one active covariate $\mathbf{X}_m$, the above quantities simplify to $a = 1$, $w = 1$, and $a_j = r_{jm}$.)

5. For $j \in A^c$, calculate $\gamma_j^+ = (r - r_j)/(a - a_j)$, and $\gamma_j^- = (r + r_j)/(a + a_j)$, and let $\gamma_j = \min(\gamma_j^+, \gamma_j^-)$. Determine $\gamma = \min\{\gamma_j, j \in A^c\}$, and m, the index

6

corresponding to the minimum $\gamma = \gamma_m$. If $\gamma_m = \gamma_m^+$, set $s_m = +1$. Otherwise, set $s_m = -1$. Update $r \leftarrow r - \gamma a$, and $r_j \leftarrow r_j - \gamma a_j$, for $j \in A^c$.

6. Repeat steps 3, 4 and 5.

3 ROBUST LARS

From the results in Section 2, it is not surprising to see that LARS is sensitive to contamination in the data. To illustrate this, we use a dataset on executives obtained from Mendenhall and Sincich (2003). The annual salary of 100 executives is recorded as well as 10 potential predictors (7 quantitative and 3 qualitative) such as education, experience etc. We label the candidate predictors from 1 to 10. LARS sequences the covariates in the following order: $(1, 3, 4, 2, 5, 6, 9, 8, 10, 7)$. We contaminate the data by replacing one small value of predictor 1 (less than 5) by the large value 100. When LARS is applied to the contaminated data, we obtain the following completely different sequence of predictors: $(\mathbf{7}, 3, \mathbf{2}, \mathbf{4}, 5, \mathbf{1}, \mathbf{10}, \mathbf{6}, \mathbf{8}, \mathbf{9})$. Predictor 7, which was selected last (10th) in the clean data, now enters the model first. The position of predictor 1 changes from first to sixth. Predictors 2 and 4 interchange their places. Thus, changing a single number in the data set completely changes the predictor sequence. As one can expect, a nonrobust follow up analysis (similar as for the examples in Section 6) based on the LARS sequence of the contaminated data would lead to an inferior model (in terms of prediction error) that excludes the important first predictor. This example thus illustrates the sensitivity of LARS to contamination.

We now introduce two approaches to robustify LARS which we call *plug-in robust LARS* and *cleaning robust LARS*.

3.1 Plug-in Robust LARS

The plug-in robust LARS approach consists of replacing the non-robust building blocks of LARS (mean, variance and correlation) by robust counterparts. The choices of fast computable robust center and scale measures are straightforward: median (med) and median absolute deviation (mad) which are used to robustly standardize the data. Unfortunately, good available robust correlation matrix estimators are computed from the d-dimensional data and therefore are very time consuming (see e.g. Rousseeuw and Leroy 1987). Therefore, we resort to robust approaches that first calculate pairwise correlations one at the time and assemble them to form the correlation matrix (see Huber 1981).

To obtain robustness against two-dimensional structural outliers we can use robust correlations derived from a pairwise affine equivariant covariance estimator. A computationally efficient choice is the bivariate M-estimator defined by Maronna (1976). Alternatively, the robust correlation estimator of Gnanadesikan and Kettenring (1972) or the related OGK estimator (Maronna and Zamar 2002) can be used. For very large, high-dimensional data we need an even faster robust correlation estimator. Huber (1981) introduced the idea of univariate Winsorization of the data, and suggested that classical correlation coefficients be calculated from the Winsorized data. Alqallaf, Konis, Martin, and Zamar (2002) re-examined this approach for the estimation of individual elements of a high-dimensional correlation matrix. For n univariate observations $x_1, x_2 \ldots, x_n$, the transformation is given by $u_i = \psi_c((x_i - \mathrm{med}(x_i))/\mathrm{mad}(x_i)), \quad i = 1, 2, \ldots, n$, where the Huber score function $\psi_c(x)$ is defined as $\psi_c(x) = \min\{\max\{-c, x\}, c\}$, with c a tuning constant chosen by the user, e.g., $c = 2$ or $c = 2.5$. Note that in our case $\mathrm{med}(x_i) = 0$ and $\mathrm{mad}(x_i) = 1$ because we used med and mad to robustly standardize the data.
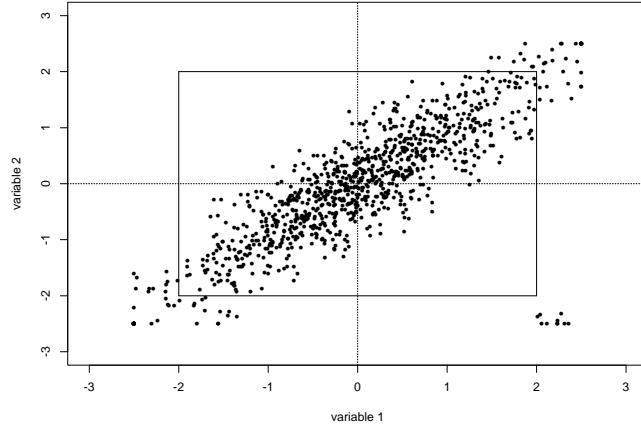
Figure 1: Illustration of the limitations of separate univariate Winsorizations ($c = 2$) when computing robust correlation estimates. The correlation outliers are only shrunken to the boundary of the square.

The univariate Winsorization approach is very fast to compute but unfortunately it does not take into account the orientation of the bivariate data. It brings the outlying observations to the boundary of a $2c \times 2c$ square, as shown in Figure 1. This plot clearly shows that the univariate approach does not resolve the effect of the obvious correlation outliers at the bottom right which are shrunken to the corner $(2, -2)$, and thus are left almost unchanged.

Bivariate Winsorization. To remedy this problem, we propose a *bivariate Winsorization* of the data based on an initial robust bivariate correlation matrix $\mathbf{R}_0$ and corresponding tolerance ellipse. Outliers are shrunken to the border of this ellipse by using the bivariate transformation $\boldsymbol{u} = \min(\sqrt{c/D(\boldsymbol{x})},\ 1)\ \boldsymbol{x}$ with $\boldsymbol{x} = (x_1, x_2)^t$. Here $D(\boldsymbol{x})$ is the Mahalanobis distance based on some initial bivariate correlation matrix $\mathbf{R}_0$. For the tuning constant c we used $c = 5.99$, the 95% quantile of the χ_2^2 distribution. The choice of $\mathbf{R}_0$ is discussed below.

The initial correlation estimate. Choosing an appropriate initial correlation matrix $\mathbf{R}_0$ is an essential part of bivariate Winsorization. In principle, we could use any robust bivariate scatter estimate but for computational convenience we propose a

9

new method called adjusted Winsorization. This method considers quadrants relative to the coordinatewise medians (which in our case are zero due to the robust standardization of the data) and uses two tuning constants to perform univariate Winsorization of the data. A larger tuning constant c_1 is used to Winsorize the points lying in the two diagonally opposed quadrants that contain the majority of the standardized data (called the "major quadrants"). A smaller tuning constant c_2 is used to Winsorize the remaining data. In this paper we use $c_1 = 2$ and $c_2 = \sqrt{h}c_1$, where $h = n_2/n_1$, n_1 is the number of observations in the major quadrants and $n_2 = n - n_1$. The initial correlation matrix $\mathbf{R}_0$ is obtained by computing the classical correlation matrix of the adjusted Winsorized data.
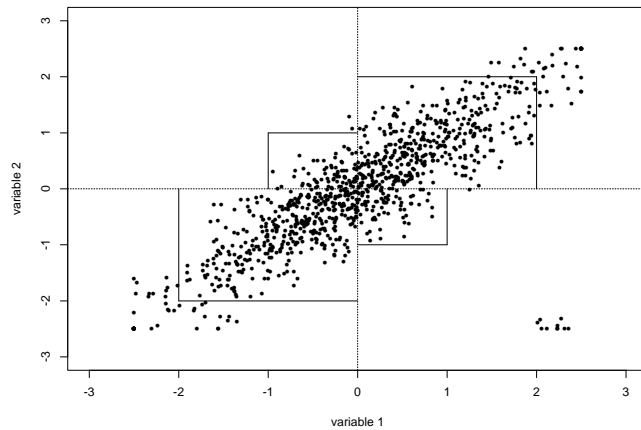

Figure 2: Adjusted Winsorization for computing the initial robust correlation estimate $\mathbf{R}_0$ (with $c_1 = 2$ and $c_2 = \sqrt{h}c_1$). The outlying points are shrunken to the edges or corners of the squares.

Figure 2 shows how adjusted Winsorization deals with bivariate outliers, which are now shrunken to the boundary of the smaller square. Thus, adjusted Winsorization handles correlation outliers much better than univariate Winsorization. Figure 3 shows the tolerance ellipses used for bivariate Winsorization of both the complete

data set of Figure 1 and the data set excluding the outliers. The ellipse for the contaminated data is only slightly larger than that for the clean data. By using bivariate Winsorization the outliers are shrunken to the boundary of the larger ellipsoid, and thus appropriately downweighted so that a robust correlation estimate is obtained. Although the initial adjusted Winsorization and the resulting bivariate Winsorization are not affine equivariant, they are very fast to compute and appropriately handle correlation outliers as illustrated in Figures 2 and 3.
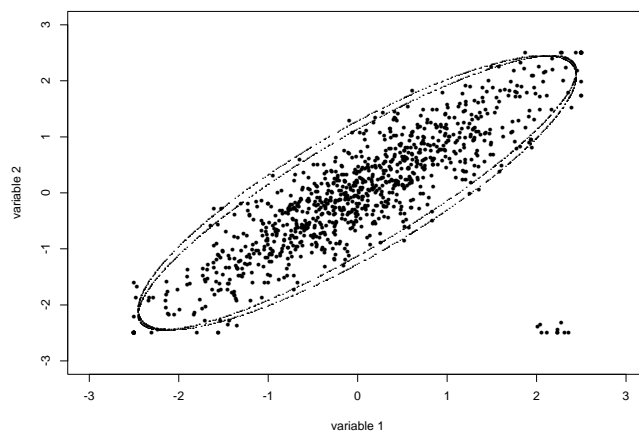

Figure 3: Bivariate Winsorization tolerance ellipses for clean (smaller ellipse) and contaminated (larger ellipse) data. The ellipses connect points of equal Mahalanobis distance (2.45), which is based on the coordinatewise median as robust center and the adjusted Winsorized correlation matrix $\mathbf{R}_0$. To calculate the bivariate Winsorized correlation estimate of the contaminated data, the points outside the largest ellipse are shrunken towards the boundary of that ellipse.

Note that the correlations based on univariate Winsorization and adjusted Winsorization both can be computed in $\mathcal{O}(n \log n)$ time. The bivariate Winsorized estimate and Maronna's M-estimate also require $\mathcal{O}(n \log n)$ time, but Maronna's M-estimate has a larger multiplication factor depending on the number of iterations required. Thus for large n, the bivariate Winsorized estimate is much faster to compute than Maronna's M-estimate.

11

We conducted a small numerical experiment to compare the computation times of the univariate and adjusted Winsorized correlation estimates which are the two candidates to serve as initial estimators for the bivariate Winsorized correlation estimator. We also report the computing times for the bivariate Winsorized correlation estimate (using adjusted Winsorization for the initial correlation estimate $\mathbf{R}_0$) and Maronna's bivariate correlation M-estimate. Figure 4 shows for each of the four correlation estimates the mean cpu times in seconds (based on 100 replicates) for 5 different sample sizes: 10000, 20000, 30000, 40000 and 50000. From this plot we see that calculating the adjusted Winsorized correlation estimate for a particular sample size n requires slightly more time than the univariate Winsorized estimate. However, the adjusted Winsorized correlation estimate is much more accurate in the presence of bivariate outliers as illustrated above. Hence, by using the adjusted Winsorized correlation estimate as initial estimate for the bivariate Winsorized estimate we gain much robustness for a very small increase in computation time. The results in Figure 4 also confirm that the bivariate Winsorized estimate is faster to compute than Maronna's M-estimate and the time difference increases with sample size. Numerical results (not presented here) showed that the bivariate Winsorized estimate is almost as accurate as Maronna's M-estimate in the presence of contamination.

3.2   Data Cleaning Robust LARS

If the dimension d is not too large and the relative sample size is not too small ($d \leq 50$ and $n/d \geq 3$, say), an alternative approach to robustify LARS is to apply it on cleaned data. For example, each standardized d-dimensional data point $\boldsymbol{x} = (x_1, \ldots, x_d)^t$ can be replaced by its Winsorized counterpart $\boldsymbol{u} = \min(\sqrt{c/D(\boldsymbol{x})},\ 1)\ \boldsymbol{x}$ in the d-dimensional space. Here $D(\boldsymbol{x}) = \boldsymbol{x}^t \mathbf{V}^{-1} \boldsymbol{x}$, is the Mahalanobis distance of $\boldsymbol{x}$ based on $\mathbf{V}$, a fast computable, robust initial correlation matrix. A reasonable choice for the
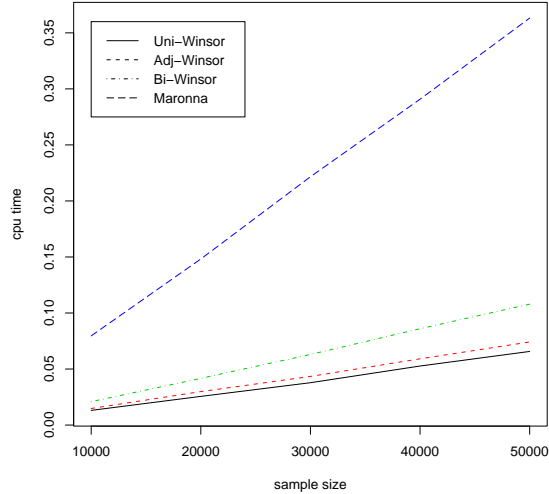
Figure 4: Average computing times for four different correlation estimators. The correlation estimators are the univariate Winsorized (Uni-Winsor), adjusted Winsorized (Adj-Winsor) and bivariate Winsorized (Bi-Winsor) correlation estimators as well as Maronna's correlation M-estimator (Maronna).

tuning distance c is $c = \chi_d^2(0.95)$, the 95% quantile of the χ_d^2 distribution.

The initial correlation matrix V. The choice of the initial correlation matrix **V** is an essential part of the Winsorization procedure. Most available high-breakdown, affine-equivariant methods are inappropriate for our purposes because they are too computationally intensive. Therefore, we again resort to pairwise approaches, that is, methods in which each entry of the correlation matrix is estimated separately (see Alqallaf et al. 2002). As before we will use bivariate Winsorization to compute the entries of **V** and positive-definiteness of the resulting matrix can be restored, if needed, using the approach in Maronna and Zamar (2002).

4 SIMULATIONS

To investigate the behavior of our robust LARS proposals we consider a simulation setting similar to that in Frank and Friedman (1993). We first create a linear model

$$y = L_1 + L_2 + \cdots + L_k + \sigma\varepsilon, \tag{1}$$

with k latent variables, where $L_1, L_2, \ldots, L_k$ and ε are independent standard normal variables. The value of σ is chosen so that the signal to noise ratio is equal to 3. A set of d candidate predictors is created as follows. Let $e_1, ..., e_d$ be independent standard normal variables and let

$$
\begin{aligned}
X_i &= L_i + \tau e_i, \qquad i = 1, ..., k \\
X_{k+1} &= L_1 + \delta e_{k+1} \\
X_{k+2} &= L_1 + \delta e_{k+2} \\
X_{k+3} &= L_2 + \delta e_{k+3} \\
X_{k+4} &= L_2 + \delta e_{k+3} \\
&\quad\vdots \\
X_{3k-1} &= L_k + \delta e_{3k-1} \\
X_{3k} &= L_k + \delta e_{3k}
\end{aligned}
$$

and
$$X_i = e_i \qquad i = 3k+1, ..., d$$

The constants $\delta = 5$ and $\tau = 0.3$ are chosen so that $\mathrm{corr}\left(X_1, X_{k+1}\right) = \mathrm{corr}\left(X_1, X_{k+2}\right) = \mathrm{corr}\left(X_2, X_{k+3}\right) = \cdots = \mathrm{corr}\left(X_k, X_{3k}\right) = 0.5$. Note that covariates $X_1, ..., X_k$ are "low noise" perturbations of the latent variables and constitute our "target covariates". Variables $X_{3k+1}, ..., X_d$ are independent noise covariates and variables $X_{k+1}, ..., X_{3k}$

are noise covariates which are correlated with the target covariates.

To allow for a fraction ϵ of outliers we considered the following sampling distributions, listed in increasing order of difficulty:

(1) $\varepsilon \sim N(0, 1)$, no contamination;

(2) $\varepsilon \sim (1 - \epsilon)N(0, 1) + \epsilon\, N(0, 1)\,/\text{Uniform}(0, 1)$, symmetric, slash contamination;

(3) $\varepsilon \sim (1 - \epsilon)N(0, 1) + \epsilon\, N(20, 1)$, asymmetric, shifted normal contamination;

(4) same as (2), except that contaminated cases come along with high leverage $X-$ values (normal random variables with mean 50 and variance 1 in our simulation);

(5) same as (3), but with high leverage outliers, as described in (4).

To compare classical LARS with our two robust alternatives we generated 500 independent samples of size $n = 150$ from the five simulation designs described above, with $k = 6$ latent variables and $d = 50$ candidate covariates. For each of these datasets we sequenced the variables using LARS, plug-in robust LARS and cleaning robust LARS. We also compare our robust procedures with the sequences generated by random forests (Breiman 2001). Random forests can sequence the covariates using two measures of "covariate importance" which are part of the output in the random forest R package implementation. The first measure is based on "out-of-bag predictions" (OOB) and the second measure is based on "impurity" (IMP). Variable selection using random forests is often based on these measures (see e.g. Díaz-Uriarte and Alvarez de Andrés 2006; Torkkola and Tuv 2006).

To summarize the simulation results, we determine for each sequence the number t_m of target variables included in the first m sequenced variables, with m ranging between 1 and 20. Figure 5 shows the average (over the 500 datasets) of t_m for each

of the methods and sampling situations. We display here the results for the case $\epsilon = 0.10$. Other levels of contamination have been considered as well and the results lead to the same conclusions as shown in the accompanying technical report available at *http://www.amstat.org/publications/jasa/supplemental_materials.*

From Figure 5 (a) we see that the five procedures perform well in the uncontaminated case. Figures 5 (b)-(e) show that, as expected, the performance of LARS considerably deteriorates under contamination. On the other hand, the robustified LARS procedures are much less affected by contamination. In the designs without leverage, plug-in robust LARS shows the best performance, while in the high leverage designs cleaning robust LARS is better. Random forests shows robust behavior under symmetric contamination (Figures 5(b) and (d)) while its performance deteriorates under asymmetric contamination (Figures 5(c) and (e)). Note that plug-in robust LARS is also affected, to some extend, by high leverage asymmetric outliers (Figure 5(e)).

Now, we compare the computational complexity of the different methods. Classical LARS sequences all d covariates in only $\mathcal{O}(nd^2)$ time. The plug-in and cleaning procedures based on M-estimators both require $\mathcal{O}((n \log n)d^2)$ time. Based on Winsorization these procedures also require $\mathcal{O}((n \log n)d^2)$ time, but with a much smaller multiplication factor. Moreover, if we are only interested in sequencing the top fraction of a large number of covariates, then the plug-in approach is much faster than the cleaning approach, because the plug-in approach only calculates the required correlations along the way instead of the 'full' correlation matrix. In this case, the complexity for plug-in methods reduces to $\mathcal{O}((n \log n)dm)$, where m is the number of sequenced variables.

Figure 6 shows the mean cpu times based on 10 replicates for LARS and plug–in robust LARS for different dimensions d with a fixed sample size $n = 2000$. For

16

(a) No contamination

(b) Slash contamination   (c) Normal contamination

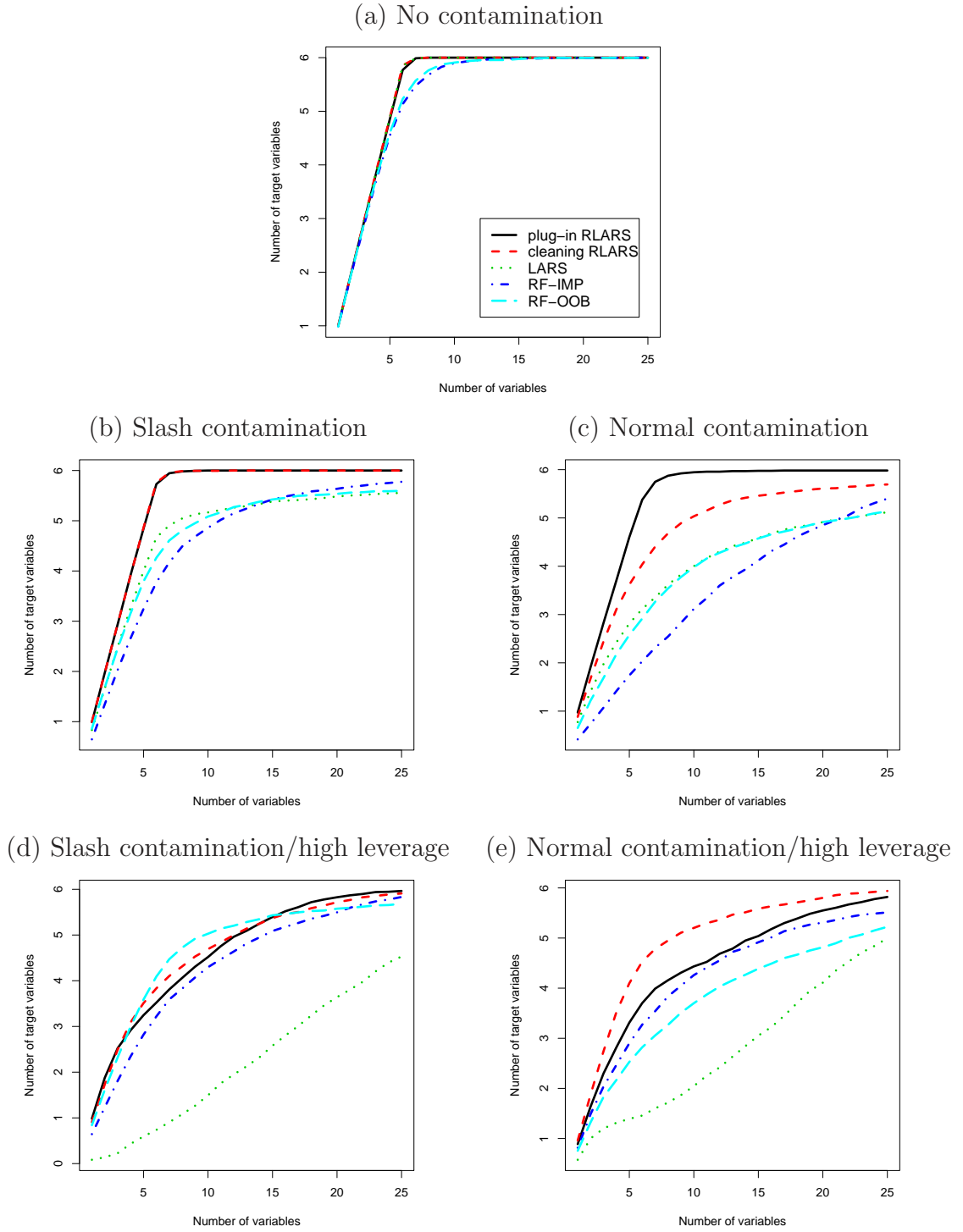(d) Slash contamination/high leverage   (e) Normal contamination/high leverage

Figure 5: Averages of the number of target variables t_m versus m for each of the methods and sampling situations considered. We generated datasets with $d = 50$ predictors, $k = 6$ latent variables, and 10% of contamination ($\epsilon = 0.1$). The lines shown in all plots follow the legend of figure (a).

17

comparison purposes we also show the computing time of plug–in robust LARS when the robust pairwise correlations are computed using Maronna's bivariate M-estimator. The times required by the corresponding cleaning methods are not shown because they are similar to the plug-in times, since we sequenced all the covariates. The approach based on Maronna's M-estimates are clearly more time consuming and the difference increases fast with dimension. Moreover, simulation results given in the technical report at *http://www.amstat.org/publications/jasa/supplemental_materials* show that plug-in robust LARS based on bivariate Winsorization performs better than plug-in robust LARS based on bivariate correlation M-estimates.
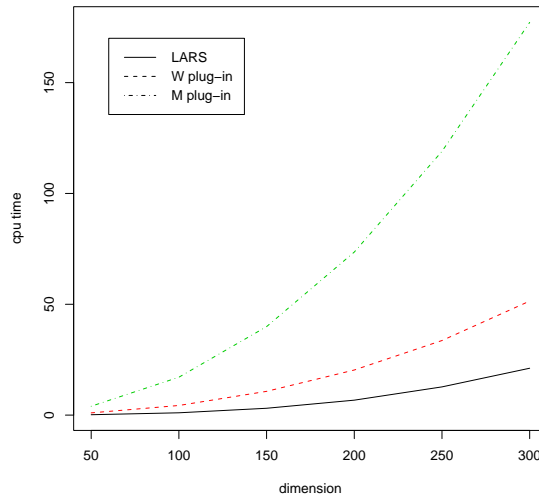

Figure 6: Average computation times for the nonrobust LARS procedures and its plug-in robustifications using bivariate Winsorization (W plug-in) or bivariate correlation M-estimates (M plug-in).

The plug-in approach can be considerably less time-consuming when only a part of the predictors are sequenced. This feature is not shared by the cleaning approach. Moreover the plug-in approach has a wider applicability as it can be used even when the dimension d exceeds the sample size n. Since plug-in has a reasonable performance

compared to the other methods, this method is to be preferred, specially for large, high-dimensional datasets.

The performance of plug-in robust LARS is studied further below. For simplicity we will call this method *robust LARS* from now on. In particular, we show that the performance of robust LARS can be improved using the bootstrap.

5 BOOTSTRAPPED SEQUENCING

To obtain more stable and reliable results we can combine robust LARS (RLARS) with bootstrap. This idea has been used in random forests and in other settings (see for example Hastie et al. 2001). We generate B bootstrap samples from the original dataset, and for each bootstrap sample use RLARS to sequence of covariates. For each covariate we then calculate the average rank over the B bootstrap samples. The m covariates with the smallest average ranks form the reduced set.

When dealing with high-dimensional datasets it may not be convenient (or even possible) to sequence all the covariates for each bootstrap sample. Note that the original sample would already be singular if the dimension d of the data exceeds the sample size (e.g. $n = 150, d = 200$ in our simulation below). We easily overcome this problem by sequencing only the first $m < n$ covariates for each bootstrap sample. We then rank the covariates according to the number of times (out of B) they are actually sequenced. When ties occur, the order of the covariates is determined by their average rank in the sequences. The resulting procedure is called *bootstrap robust LARS* and denoted B-RLARS.

We ran a simulation to compare B-RLARS with the initial RLARS and random forests. In our simulation, we generated 250 datasets according to the simulation design described in Section 4, with $d = 200$ candidate covariates, $k = 10$ target

covariates and 10% high leverage outliers (that is, using the sampling distributions in cases (4) and (5) of the previous section). We generated $B = 50$ bootstrap samples from each of the simulated datasets and for each bootstrap sample we sequenced the first 50 covariates.

(a) Slash contamination/high leverage   (b) Normal contamination/high leverage
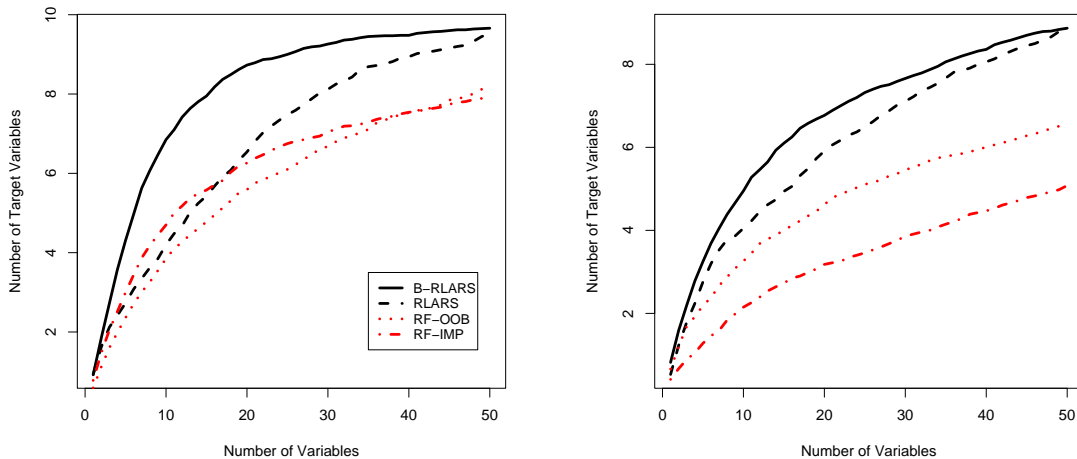

Figure 7: Averages (over 250 datasets) of the number of target variables t_m versus m for each of the methods and sampling situations considered. We generated datasets with $d = 200$ predictors, $k = 10$ latent variables, and 10% of contamination ($\epsilon = 0.1$).

Figure 7 shows that applying the bootstrap considerably improves the performance of RLARS. Note that the performance of random forests is worse in this high-dimensional setting. Moreover, it is not clear which of the two measures of importance (impurity or out-of-bag) should be preferred.

6 EXAMPLES

In this section we use two real datasets to further illustrate the performance of B-RLARS.

20

In practice we often don't know the number of covariates that are needed in the model. Hence, a graphical tool to select the size of the reduced set would be useful. We use the following plot: Starting with the first variable in the sequence we increase the number of variables (along the sequence) and each time fit a robust regression model to compute a robust R^2 measure such as $R^2 = 1 - \text{Median}(\boldsymbol{e}^2)/\text{MAD}^2(\mathbf{Y})$, where $\boldsymbol{e}$ is the vector of residuals from the robust fit (see also Rousseeuw and Leroy 1987). We then plot these robust R^2 values against the number of variables in the model to obtain a *learning curve* (see also Croux, Filzmoser, Pison, and Rousseeuw 2003). The size of the reduced set, m, can be selected as the point where the learning curve does not have a considerable slope anymore. Note that since algorithms for robust regression only provide an approximate solution, it can occur that the robust R^2 does not always increase with the number of covariates. The learning curve could be extended by computing the robust R^2 values for a number of bootstrap samples to obtain standard error bars around the actual values. Similarly as in Hastie et al. (2001), these standard errors can be used to select m as the size of the smallest model that has robust R^2 within the one-standard error range of the robust R^2 value where the curve levels off.

Demographic data. This dataset contains demographical information on the 50 states of the United States for 1980. The response variable is the murder rate per 100,000 residents. There are 25 predictors which we number from 1 to 25. Exploration of the data using robust estimates and graphical tools revealed one clear outlier. To select an optimal prediction model we used least squares regression on the dataset without the outlier (called the "clean dataset"). We considered all possible subsets of predictors and estimated the prediction error by using 5-fold CV. We selected the "full CV-model" that yielded smallest CV prediction error. This model contains the following 7 covariates $(6, 9, 13, 14, 19, 20, 25)$.

Figure 8 shows the learning curve for the Demographic data based on B-RLARS. This plot suggests a reduced set of at most size 12, which includes the following covariates $(25, 18, 17, 20, 13, 12, 24, 10, 23, 11, 6, 21)$. We applied all subsets selection
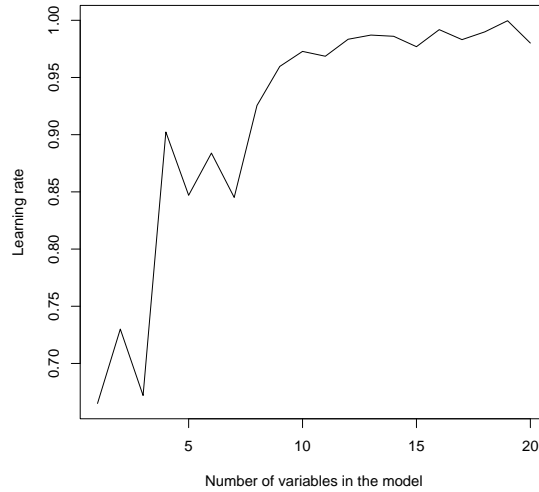

Figure 8: Learning curve for Demographic data.

to these 12 variables using 5-fold CV on the clean dataset. The model selected in this case, called B-RLARS CV-model, has the following 6 covariates: $(6, 13, 18, 20, 24, 25)$.

For comparison, we also obtained the optimal 5-fold CV-model (again using the clean dataset) starting with the first 12 predictors sequenced by the standard nonrobust LARS. This yielded the model with the following 8 predictors: $(4, 11, 13, 14, 17, 19, 20, 25)$, called the LARS-CV model. This model has only three variables in common with the B-RLARS CV-model.

Random forests and regression support vector machines are two techniques that are frequently used in the machine learning and bioinformatics communities because they can handle high-dimensional data efficiently. Both techniques can perform variable selection and can be used to sequence the variables. For random forests we use

22

out of bag (OOB) prediction errors. For support vector machines (SVM), Duan and Rajapakse (2005) proposed the multiple support vector machine recursive feature elimination (MSVM-RFE) procedure in the context of classification (we have easily adapted this procedure for regression support vector machines). Since both methods are considered to have some degree of robustness, we compared these two techniques with our robust approach.

To select the reduced set using random forests we first sequenced the variables according to the out of bag (OOB) importance measure. Then we calculated the OOB prediction error for each of the 25 possible subsequences formed by the first m variables ($m = 1, 2, ..., 25$) in the sequence. Finally, we selected the subsequence with the smallest OOB prediction error. This yielded the model with covariates $(17, 25, 18, 20, 24)$, called the RF-SEL model. Applying 5-fold CV to the clean data using these 5 variables yielded the model with covariates $(18, 20, 24, 25)$, called the RF-SEL CV-model. For completion, we also selected the best model by 5-fold CV starting from the reduced set of the first 12 variables according to the OOB importance measure in random forests. This lead to the model with covariates $(4, 18, 20, 24, 25)$ which we call the RF-RED CV-model.

MSVM-RFE performs variable selection for SVM with linear kernel by using backward elimination based on the prediction error estimated by multiple runs of 5-fold CV. At each step, a measure of importance of the predictors is calculated based on the size of its regression coefficient in each of the SVM fits in the CV procedure. The least important predictor according to this measure is then eliminated. The MSVM-RFE procedure based on 20 runs of 5-fold CV selected a model with 8 predictors, $(2, 6, 9, 13, 15, 18, 20, 25)$, which we call the MSVM-RFE model. We then applied 5-fold CV to the clean data using this set of size 8. The model selected in this case, called the MSVM-RFE CV-model, has the following 6 covariates: $(6, 9, 13, 18, 20, 25)$.
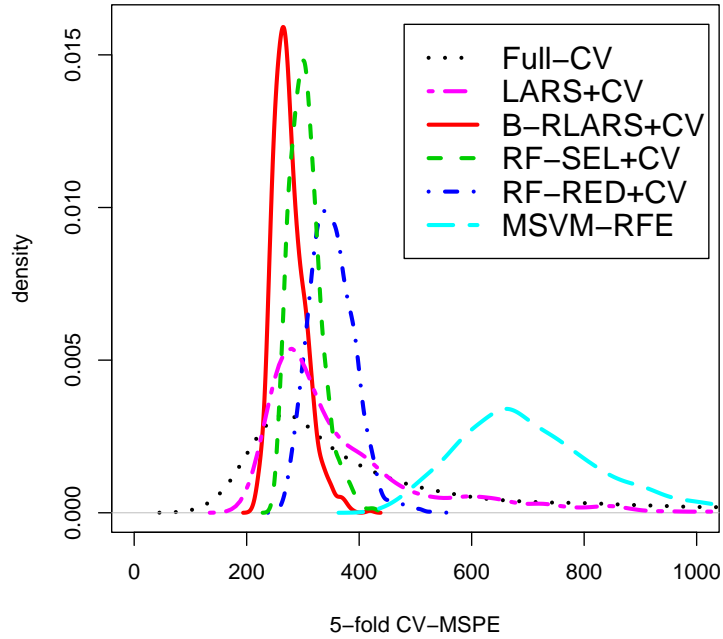
Figure 9: Densities based of 5-fold CV mean squared prediction error of the selected models for the Demographic data. The densities are based on 1000 estimates of the 5-fold CV error. The densities of the RF-SEL and MSVM-RFE CV-models are not shown because they are almost indistinguishable from the RF-SEL CV and B-RLARS CV-models respectively.

To compare the models selected by the different procedures, we estimated the mean squared prediction error (MSPE) for each of these models 1000 times using 5-fold CV. The density curves are shown in Figure 9. From this plot we clearly see that the full CV-model is not stable. It yields highly variable CV-MSPEs. In fact, some of the mean squared prediction errors were so large that we did not include them in the plot. The LARS CV-model yields only a small improvement on the full CV-model. The variance of the CV-MSPEs is still high. The models that resulted from B-RLARS and random forest are far more stable. The B-RLARS CV-model yields the best solution with the smallest average CV-MSPE as well as a small variance. The 5-fold CV-MSPEs of the RF-SEL CV-model have the same variance but a larger

mean and the CV-MSPEs of the RF-RED CV-model have an even larger mean as well as a larger variance. We also determined the 5-fold CV-MSPEs of the RF-SEL model with all five predictors selected by the OOB procedure. Although this model has one predictor more than the RF-SEL CV-model, their CV-MSPE densities are almost indistinguishable. The MSVM-RFE model with 8 predictors yields an CV-MSPE density with a large mean as well as a large variance, and thus an undesirable result. However, the MSVM-RFE CV-model considerably improves the initial model. Its CV-MSPE density is not shown in Figure 9 as it overlaps with the CV-MSPE density of the B-RLARS CV-model. Note that the B-RLARS CV and MSVM-RFE CV-models both contain six predictors and they have 5 predictors in common. The only difference between both models is that the B-RLARS CV-model contains predictor 24 whereas the MSVM-RFE CV-model contains predictor 9 instead.

Finally, note that we needed almost 10 days to find the best full CV-model, while it took less than 5 minutes to determine the B-RLARS CV-model or the RF CV-models and a little bit longer to determine the MSVM-RFE CV-model.

Protein data. This dataset of $n = 145,751$ protein sequences was used for the KDD-Cup 2004. Each of the 153 blocks correspond to a native protein, and each data-point of a particular block is a candidate homologous protein. There are 75 variables in the dataset: the block number (categorical) and 74 measurements of protein features. We use the first feature as the response. Though this analysis may not be of particular scientific interest, it will demonstrate the scalability and stability of the robust LARS algorithm.

We applied RLARS to this dataset, and obtained a reduced subset of size 25 from the original $d = 225$ covariates (152 block indicators + 73 features) in only 30 minutes. Given the huge computational burden of other robust variable selection procedures, our algorithm may be considered extremely suitable for computations of
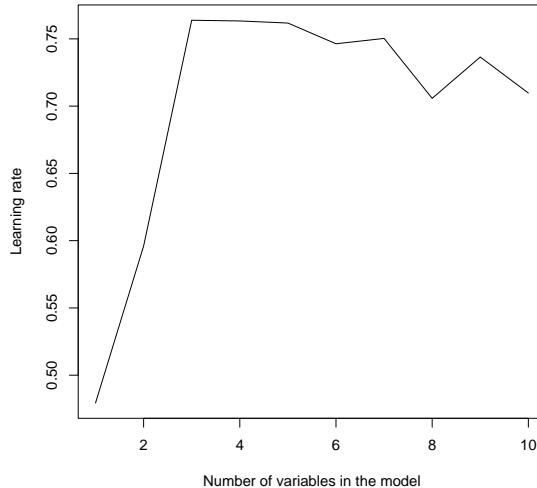
Figure 10: Learning curve for Protein data.

this magnitude.

To further investigate the performance of B-RLARS with this dataset, we select 5 blocks with a total of $n = 4141$ protein sequences. These blocks were chosen because they contain the highest proportions of homologous proteins (and hence the highest proportions of potential outliers). We split the data of each block into two almost equal parts to get a training sample of size $n = 2072$ and a test sample of size $n = 2069$. The number of covariates is $d = 77$, consisting of 4 block indicators (variables $1 - 4$) and 73 features. We applied B-RLARS with $B = 100$ bootstrap samples and for each bootstrap sample we sequenced the first 50 variables. The resulting learning curve is shown in Figure 10. This plot suggests a drastic reduction to a subset of only 5 covariates. The first 5 predictors found by B-RLARS are $(14, 13, 5, 7, 76)$. By fitting all possible submodels with MM-estimators and using robust FPE (see e.g. Maronna et al. 2006) as selection criterion, we checked whether a better submodel of the 5 predictor model exists, but no submodel yielded a lower robust FPE value.

As in the previous example, we used random forests with its OOB importance

measure to sequence the variables and used OOB prediction error to determine the size of the reduced sequence. This yielded a model with 22 variables, that we call the RF model. Since 22 variables are too many to determine an optimal submodel using an exhaustive search, we only considered the 22 submodels that consist of the first k ($k = 1, \ldots, 22$) variables in the reduced sequence of 22 variables (sequenced according to OOB-importance). Fitting these submodels with MM-estimators and using robust FPE selection led to a model with 18 variables. We call this the RF-RFPE model.

We also applied the MSVM-RFE procedure, which also resulted in a model with 22 predictors (of which 10 predictors are in common with the RF model). As with RF, we considered the 22 submodels that consist of the first variables in the MSVM-RFE sequence, sorted according to their MSVM-RFE importance (which corresponds to the order by which variables are eliminated). Using MM-estimators and robust FPE didn't help to reduce the number of variables (the original model with all 22 covariates yielded the smallest robust FPE).

To compare the B-RLARS model with the RF, RF-RFPE, and MSVM-RFE models, we first fitted the models using the training data, and then used the fitted models to predict the test data outcomes. Since the test data is likely to contain outliers as well, we report the 1%, 5% and 10% trimmed means of squared prediction errors in Table 1 (the largest errors are trimmed). From this table we can see that while the MSVM-RFE and RF models have the same number of predictors, the MSVM-RFE model is much worse than the RF-model. The RF-RFPE model does not improve on the initial RF model, but its prediction errors are very similar while containing 4 predictors less. The trimmed prediction errors of the B-RLARS model are only slightly larger than those of the RF and RF-RFPE models despite the large difference in dimensions. Hence, the B-RLARS procedure clearly managed to identify the

	Trimming fraction		
Model	1%	5%	10%
B-RLARS	116.19	97.73	84.67
RF	111.11	93.80	81.30
RF-RFPE	111.30	93.92	81.27
MSVM-RFE	173.70	150.48	133.17

Table 1: Trimmed means of squared prediction errors in the protein test set, obtained by the B-RLARS, RF, RF-RFPE, and MSVM-RFE models fitted on the protein training data.

five most important predictors among the 77 candidate variables.

7 CONCLUSIONS

LARS is a very effective, time-efficient model building tool, but is not resistant to outliers. We introduced two different approaches to construct robust versions of the LARS technique. The plug-in approach replaces the classical Pearson correlations in LARS by easily computable robust correlation estimates. The cleaning approach first transforms the dataset by shrinking the outliers towards the bulk of the data, and then applies LARS on the transformed data. Both approaches use robust bivariate correlation estimates which can be computed efficiently using bivariate Winsorization.

The data cleaning approach is limited in use because the sample size needs to be (much) larger than the number of candidate predictors to ensure that the resulting correlation matrix is positive definite. Moreover, the data cleaning approach is more time consuming than the plug-in approach, certainly when only part of the predictors is being sequenced. Since the plug-in approach has good performance, is faster to compute and more widely applicable, we prefer this method.

We propose to use B-RLARS to sequence the candidate predictors and as such identify a reduced set of most promising predictors from which a more refined model

can be selected in a second segmentation step. In general, the reduced set obtained by B-RLARS contains more of the important covariates than the reduced set obtained by initial RLARS. Software code in R to compute B-RLARS together with a detailed description of its use and the datasets used in the examples, is available at

http://users.ugent.be/∼svaelst/software/RLARS.html.

There is a growing literature on feature selection in Machine Learning (see for example Díaz-Uriarte and Alvarez de Andrés 2006; Torkkola and Tuv 2006; Rajapakse and Wang 2005); our paper makes only a limited comparison with them because the focus is on robustifying LARS. In fact, this is a first attempt to robustify LARS, with an emphasis on computational efficiency. The underlying theory has yet to be developed and there may well be other better approaches. For instance, one could consider solving a robust version of the original problem posed by LARS.

It is important to determine the size of the reduced sequence, that is, the number of predictors that is retained for the second step. This number is a trade-off between success-rate, that is the number of important predictors captured in the reduced set, and feasibility of the segmentation step. A simulation study shown in our technical report available at *http://www.amstat.org/publications/jasa/supplemental_materials* indicated that the reduced set can have size comparable to the actual number of relevant candidate predictors. However, in practice this number is usually unknown. To still get an idea about an appropriate size for the reduced set we propose a robust learning curve that plots robust R^2 values versus dimension. An appropriate size can be selected as the dimension corresponding to the point where the curve starts to level off.

APPENDIX: TECHNICAL DERIVATIONS

A. Determination of γ for One Active Covariate

Assume that the first selected covariate is $+\mathbf{X}_m$. The current fit $\hat{\mathbf{Y}} \leftarrow \mathbf{0}$ should be modified as

$$\hat{\mathbf{Y}} \leftarrow \gamma \, \mathbf{X}_m.$$

The distance γ should be such that the residual $(\mathbf{Y} - \hat{\mathbf{Y}})$ will have equal (maximal) correlation with $+\mathbf{X}_m$ and another signed covariate $\mathbf{X}_j$. We have

$$\text{cor}(\mathbf{Y} - \hat{\mathbf{Y}}, \mathbf{X}_m) = \frac{\mathbf{X}_m'(\mathbf{Y} - \gamma\mathbf{X}_m)/n}{\text{SD}(\mathbf{Y} - \gamma\mathbf{X}_m)} = \frac{r - \gamma}{\text{SD}(\mathbf{Y} - \gamma\mathbf{X}_m)}, \tag{A.1}$$

and

$$\text{cor}(\mathbf{Y} - \hat{\mathbf{Y}}, +\mathbf{X}_j) = \frac{\mathbf{X}_j'(\mathbf{Y} - \gamma\mathbf{X}_m)/n}{\text{SD}(\mathbf{Y} - \gamma\mathbf{X}_m)} = \frac{r_j - \gamma r_{jm}}{\text{SD}(\mathbf{Y} - \gamma\mathbf{X}_m)}. \tag{A.2}$$

Equating (A.1) to (A.2), we have

$$\gamma(+\mathbf{X}_j) = \frac{r - r_j}{1 - r_{jm}}. \tag{A.3}$$

Similarly, equating (A.1) with $\text{cor}(\mathbf{Y} - \hat{\mathbf{Y}}, -\mathbf{X}_j)$ yields

$$\gamma(-\mathbf{X}_j) = \frac{r + r_j}{1 + r_{jm}}. \tag{A.4}$$

The distance γ is now obtained by taking the minimum of (A.3) and (A.4) over all inactive (not yet selected) covariates $\mathbf{X}_j$. The signed covariate corresponding to this minimum is the (signed) covariate that enters the model at this point.

B. Quantities Related to Equiangular Vector $\mathbf{B}_A$

Let A denote the set of indices corresponding to the 'active' covariates. Let $\mathbf{X}_A = (\cdots s_l \mathbf{X}_l \cdots)$, $\quad l \in A$, where s_l is the sign of $\mathbf{X}_l$ as it enters the model. The standardized equiangular vector $\mathbf{B}_A$ is obtained using the following three conditions.

1. $\mathbf{B}_A$ is a linear combination of the active signed predictors, so

$$\mathbf{B}_A = \mathbf{X}_A \, \boldsymbol{w}_A , \quad \text{where } \boldsymbol{w}_A \text{ is a vector of weights.} \tag{A.5}$$

2. $\mathbf{B}_A$ has unit variance:

$$\frac{1}{n}\mathbf{B}'_A \mathbf{B}_A = 1. \tag{A.6}$$

3. $\mathbf{B}_A$ has equal correlation (a, say) with each of the active predictors. Since the covariates and $\mathbf{B}_A$ are standardized, this means that

$$\frac{1}{n}\mathbf{X}'_A \mathbf{B}_A = a \, \mathbf{1}_A , \quad \mathbf{1}_A \text{ is a vector of 1's.} \tag{A.7}$$

Using equation (A.5) in equation (A.6) yields

$$\frac{1}{n}\boldsymbol{w}'_A \mathbf{X}'_A \mathbf{X}_A \boldsymbol{w}_A = 1,$$

or

$$\boldsymbol{w}'_A \mathbf{R}_A^{(s)} \boldsymbol{w}_A = 1, \tag{A.8}$$

where $\mathbf{R}_A^{(s)}$ is the correlation matrix of the active signed variables. Using (A.5) in (A.7), we obtain

$$\mathbf{R}_A^{(s)} \boldsymbol{w}_A = a\mathbf{1}_A,$$

so that the weight vector $\boldsymbol{w}_A$ can be expressed as

$$\boldsymbol{w}_A = a \, (\mathbf{R}_A^{(s)})^{-1} \mathbf{1}_A.$$

Let $\mathbf{R}_A$ be the correlation matrix of the unsigned active covariates, i.e., $\mathbf{R}_A$ is a submatrix of $\mathbf{R_X}$. Let $\boldsymbol{s}_A$ be the vector of signs of the active covariates (we get the sign of each covariate as it enters the model). We can then rewrite $\boldsymbol{w}_A$ as

$$\boldsymbol{w}_A = a \, (\mathbf{D}_A \mathbf{R}_A \mathbf{D}_A)^{-1} \mathbf{1}_A, \tag{A.9}$$

where $\mathbf{D}_A$ is the diagonal matrix whose diagonal elements are the elements of $\boldsymbol{s}_A$. Finally, using equation (A.9) in equation (A.8), yields

$$a = [\mathbf{1}_A'(\mathbf{D}_A \mathbf{R}_A \mathbf{D}_A)^{-1} \mathbf{1}_A]^{-1/2}. \tag{A.10}$$

Note that the procedure is stopped when $\det \mathbf{R}_A = 0$ and thus the inverse does not exist.

The correlation of an inactive covariate $\mathbf{X}_j$ with $\mathbf{B}_A$, denoted by a_j, can be expressed as follows

$$a_j = \frac{1}{n}\mathbf{X}_j'\mathbf{B}_A = \frac{1}{n}\mathbf{X}_j'\mathbf{X}_A \boldsymbol{w}_A = (\mathbf{D}_A \boldsymbol{r}_{jA})' \boldsymbol{w}_A, \tag{A.11}$$

where $\boldsymbol{r}_{jA}$ is the vector of correlation coefficients between the inactive covariate $\mathbf{X}_j$ and the (unsigned) selected covariates. Thus, we need only (a part of) the correlation matrix of the data (not the observations themselves) to determine the above quantities.

C. Determination of γ for Two or More Active Covariates

Let us update $r \leftarrow (r - \gamma)$, see (A.1), and $r_j \leftarrow (r_j - \gamma r_{jm})$, see (A.2).

The correlation of an active covariate with the 'current' residual $\mathbf{Y} - \hat{\mathbf{Y}}$ is $r/\text{SD}(\mathbf{Y} - \hat{\mathbf{Y}})$, and the correlation of the active covariate with the current equiangular vector $\mathbf{B}_A$ is 'a'. Therefore, the correlation between an active covariate and the 'modified' residual $(\mathbf{Y} - \hat{\mathbf{Y}} - \gamma_A \mathbf{B}_A)$ is

$$\frac{r - \gamma_A \, a}{\text{SD}(\mathbf{Y} - \hat{\mathbf{Y}} - \gamma_A \mathbf{B}_A)}.$$

An inactive covariate $+\mathbf{X}_j$, $j \in A^c$, has correlation $r_j/\text{SD}(\mathbf{Y} - \hat{\mathbf{Y}})$ with the 'current' residual, and it has correlation a_j with $\mathbf{B}_A$. Therefore, the correlation between $+\mathbf{X}_j$, $j \in A^c$, and the 'modified' residual is

$$\frac{r_j - \gamma_A \, a_j}{\text{SD}(\mathbf{Y} - \hat{\mathbf{Y}} - \gamma_A \mathbf{B}_A)}.$$

Equating the above two quantities, we get

$$\gamma_A(+\mathbf{X}_j) = (r - r_j)/(a - a_j). \tag{A.12}$$

Similarly,

$$\gamma_A(-\mathbf{X}_j) = (r + r_j)/(a + a_j). \tag{A.13}$$

We have to choose the minimum possible γ_A over all inactive covariates. Note that when A has only one covariate, (A.12) and (A.13) reduce to (A.3) and (A.4), respectively.

REFERENCES

Alqallaf, F. A., Konis, K. P., Martin, R. D., and Zamar, R. H. (2002), "Scalable Robust Covariance and Correlation Estimates for Data Mining," *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta*, 14-23.

Atkinson, A. C., and Riani M. (2002), "Forward Search Added-Variable t-Tests and the Effect of Masked Outliers on Model Selection," *Biometrika,* 89, 939-946.

Breiman, L. (2001), "Random Forests," *Machine Learning,* 24, 5-32.

Croux, C., Filzmoser, P., Pison, G., and Rousseeuw, P. J. (2003), "Fitting Multiplicative Models by Robust Alternating Regressions," *Statistics and Computing,* 13, 23-36

Cantoni, E., and Ronchetti, E. (2001), "Robust Inference for Generalized Linear Models," *Journal of the American Statistical Association*, 96, 1022-1030.

Díaz-Uriarte, R., and Alvarze de Andrés, S. (2006). "Gene Selection and Classification of Microarray Data using Random Forest," *BMC Bioinformatics,* 7:3.

Efron, B. E., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression," *The Annals of Statistics*, 32, 407-451.

Frank, I., and Friedman, J. H. (1993), "A Statistical View of Some Chemometrics Regression Tools," *Technometrics*, 35, 109-148.

Genton, M. G., and Ronchetti, E. (2003), "Robust Indirect Inference," *Journal of the American Statistical Association*, 98, 67-76.

Gnanadesikan, R., and Kettenring, J. R. (1972), "Robust Estimates, Residuals, and Outlier Detection With Multiresponse Data," *Biometrics*, 28, 81-124.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning*, New York: Springer-Verlag.

Huber, P. J. (1981), *Robust Statistics*, New York: Wiley.

Jiang, W., and Turnbull, B. (2004), "The Indirect Method: Inference Based on Intermediate Statistics – a Synthesis and Examples," *Statistical Science,* 19, 239-263.

Maronna, R. A. (1976), "Robust M-estimators of Multivariate Location and Scatter," *The Annals of Statistics*, 4, 51-67.

Maronna, R. A., Martin, R. D. and Yohai, V. J. (2006), *Robust Statistics: Theory and Methods.* John Wiley and Sons.

Maronna, R. A., and Zamar, R. H. (2002), "Robust Estimates of Location and Dispersion for High-Dimensional Datasets," *Technometrics,* 44, 307-317.

Mendenhall, W., and Sincich, T. (2003), *A Second Course in Statistics: Regression Analysis* (6th ed.), New Jersey: Pearson Education, Inc.

Morgenthaler, S., Welsch, R. E., and Zenide, A. (2003), "Algorithms for Robust Model Selection in Linear Regression," in *Theory and Applications of Recent Robust Methods*, eds. M. Hubert, G. Pison, A. Struyf, and S. Van Aelst, Basel (Switzerland): Birkhäuser-Verlag.

Müller, S. and Welsh, A. H. (2005), "Outlier Robust Model Selection in Linear Regression," *Journal of the American Statistical Association*, 100, 1297-1310.

Ronchetti, E. (1985), "Robust Model Selection in Regression," *Statistics and Probability Letters*, 3, 21-23.

Ronchetti, E., Field, C., and Blanchard, W. (1997), "Robust Linear Model Selection by Cross-Validation," *Journal of the American Statistical Association*, 92, 1017-1023.

Ronchetti, E., and Staudte, R. G. (1994), "A Robust Version of Mallow's C_p," *Journal of the American Statistical Association*, 89, 550-559.

Ronchetti, E., and Trojani, F. (2001), "Robust Inference with GMM Estimators," *Journal of Econometrics,* 101, 37-69.

Rousseeuw, P. J., and Leroy, A. M. (1987), *Robust Regression and Outlier Detection*, New York: Wiley-Interscience.

Sommer, S., and Huggins, R. M. (1996), "Variable Selection Using the Wald Test and Robust C_p," *Journal of the Royal Statistical Society*, Ser. B, 45, 15-29.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, Ser. B, 58, 267-288.

Torkkola, K., and Tuv, E. (2006), "Ensembles of Regularized Least Squares Classifiers for High-Dimensional Problems," in *Feature Extraction: Foundations and Applications,* eds. I. Guyon, S. Gunn, M. Nikravesh, and L.A. Zadeh), 297-313.

Weisberg, S. (1985), *Applied Linear Regression* (2nd ed.), New York: Wiley-Interscience.