Stable Adaptive Model Selection

Luella Fu and Yingying Fan [*]

May 21, 2014

Abstract

Large scale data analysis has stimulated the developments of various sparse modeling techniques. With high dimensionality but limited sample size, variables with very weak signal strength are indistinguishable from noise. In real application, a stable method that removes noise and very weak variables from the final model can improve model stability and boost the significance of relatively important variables. The tasks of selecting variables and evaluating their importance are typically conducted separately. Yet, it is desirable to use a stabilizing method that performs model selection while adaptively removing noise and very weak variables. In this paper, we suggest the stable adaptive model selection that automatically performs variable selection and significance selection together. It is computationally efficient and, under certain conditions, theoretically proven to enjoy the variable selection sure screening property. The advantages of our method are supported by simulation and real data examples.

Keywords: Greedy algorithm; High dimension; Model selection; Sparse modeling; Threshold

1 Introduction

Model selection and estimation procedures are widely used on high dimensional data to produce sparse models. Particularly fruitful are regularization methods. Amongst these,

[*] Luella Fu is PhD candidate, and Yingying Fan is Assistant Professor, Data Sciences and Operations Department, Marshall School of Business, University of Southern California, Los Angeles, CA 90089 (e-mails: luellafu@usc.edu and fanyingy@marshall.usc.edu). This work was supported by NSF CAREER Award DMS-1150318 and USC Marshall summer research funding.

the most popular and well studied is the L_1 regularization method Lasso (Tibshirani, 1996). As summarized by Zou and Zhang (2009), research shows that Lasso tends to select a larger model than the true one. Many of the selected variables are either noise variables or variables with very weak signal strength. Indeed, with high dimensionality but limited sample size, variables with very weak signal strength are indistinguishable from noise. In addition, for variables selected by Lasso, it is often the case that many of the estimated regression coefficients are close to zero. A model that includes a large number of variables with nearly-zero regression coefficients is undesirable since it increases model complexity, makes the model difficult to interpret, and causes large variations in estimation and prediction. Thus, an important and interesting question is how to remove these noise variables and very weak variables from the selected model.

Variable significance has been less explored than other aspects of high dimensional sparse modeling, resulting in a smaller literature. Wasserman and Roeder (2009) apply a t-test on the model resulting from a multi-stage procedure, which splits and handles data in three parts, to eliminate unimportant variables. Meinshausen and Bühlmann (2010) propose a general subsampling method in which variables that frequently appear are chosen, increasing the probability of selecting significant variables and decreasing the probability of selecting noise variables. Minnier et al. (2011) estimate covariance structure for concave penalty estimators and construct confidence intervals using a perturbation method. Bühlmann (2012) takes advantage of the low variance of the ridge estimator and uses a bias-corrected version of it to conduct hypothesis testing for high dimensional linear models. Zhang and Zhang (2013) develop the low dimensional projection estimator which produces confidence intervals for coefficients. Recently, Lockhart et al. (2013) propose the covariance test statistic for Lasso, which tests the significance of the covariates that enter the current Lasso model along the Lasso solution path.

It is clear from the variety and ingenuity in these approaches that important variables can be assessed in numerous ways. One more idea would be to enfold the selection of significant variables along with model selection into the same procedure. Particularly, compare the advantages of an integrated model to a two-stage method that first selects variables and then evaluates variables' importance. Many variables in the final set may have nearly-zero coefficients, which can increase model instability and cause difficulty in evaluating the significance of truly important variables. More importantly, the two-stage procedure is not adaptive; in the model fitting stage, over the course of generating a sequence of sparse candidate models, the significance of variables can change. Noise covariates may enter. An adaptive method could remove such noise covariates and boost the significance of some relatively weak but still important covariates.

We offer a greedy procedure with these desirable adaptive characteristics. This method, stable adaptive model selection (SAMS), modifies least angle regression, also known as LARS (Efron et al., 2004), to select important variables. In the original LARS algorithm, the covariate with the highest correlation to the residual enters the model first. Its estimate is increased to the point at which another covariate outside the model has equal correlation to the residual. Then, that covariate enters the model, and the process repeats. SAMS changes LARS by halting LARS once it produces a coefficient that exceeds a set threshold. This threshold acts as a critical value. Using the selected covariate, SAMS then produces a regression coefficient using ordinary least squares (OLS) and thresholds the result. SAMS iterates the procedure. As SAMS iterates, it repeatedly thresholds the model estimates so that noise or very weak variables are excluded.

"Stable" in stable adaptive model selection comes from SAMS's ability to correct for and eliminate noise variables. Though a noise variable may be highly correlated to a true covariate and enter LARS early on, the true covariate can overtake the false one before any estimate reaches the threshold. As a result, only the true regressor enters the model. Even when a noise variable appears strong enough to enter during the modified LARS stage of SAMS, it can appear weak in the presence of true variables during ordinary least squares estimation. SAMS will therefore threshold out the spurious variable. Consequently, SAMS enjoys great stability even when handling many noisy or very weak covariates.

Stable adaptive model selection shares common ideas with the methods proposed in Wasserman and Roeder (2009) and in van de Geer et al. (2011). Wasserman and Roeder (2009) consider using thresholded forward stepwise regression to select variables and ordinary least squares to produce estimates. These estimates are then used for inference. The authors show that this procedure can achieve model selection consistency under certain circumstances. van de Geer et al. (2011) study a method in which thresholded Lasso variables are refit once using OLS. The authors also prove that this thresholded procedure can achieve variable selection consistency.

SAMS is similar to both methods in that it refits variables using ordinary least squares after thresholding. It is additionally similar to Wasserman and Roeder (2009) in that it contains a cautious step to check if variables are truly non-zero. One difference is that these methods are procedures that handle variable selection and estimation separately. In SAMS, these tasks participate in one feedback loop. The key distinction is that SAMS adaptively adds, estimates, and thresholds out variables. It is therefore a more dynamic procedure that adjusts the model to screen out noise variables. In this spirit, SAMS is similar to correlation pursuit (Zhong et al., 2013), which applies an adaptive greedy method to a different model setting.

As a greedy algorithm, SAMS has additional advantages. It terminates LARS prematurely and is therefore computationally fast. The early stopping feature quite naturally also prevents overfitting. This property is especially useful for data sets with small numbers of observations that cannot be easily cross-validated. We additionally prove that under certain conditions, SAMS is a method that automatically produces a sequence of sparse models with true covariates. In fact, because SAMS is a thresholding procedure, SAMS falls under the theoretical framework detailed in Fan and Lv (2013).

The theory adds to a still small literature on the statistical properties of greedy methods. Works which explore the properties of classical forward stepwise algorithms include Tropp (2004), Donoho and Stodden (2006), and Wang (2009). Additional modifications to and theory for greedy estimators are given in Zhang (2008), Barron et al. (2008), and Ing and Lai (2011).

In fact greedy methods continue to be explored as useful tools for a variety of high dimensional settings. Yuan and Zou (2009) build upon LARS to create a generalized algorithm while Zhong et al. (2013) advance sufficient dimension reduction by applying an iterative stepwise procedure. Also, Li (2006) implements a greedy method to ameliorate the curse of high dimensionality and perform multi-class classification. Here we offer SAMS, which is also a greedy method, but one which is stable in high dimensions, adaptive, and performs model selection and importance selection simultaneously.

We lay out the algorithm in the next section. In section 3, we detail its theoretical properties. Then, in section 4, we use simulation to demonstrate properties of stable adaptive model selection and give results of applying SAMS to spam detection. We place technical details in the appendix.

2 Stable Adaptive Model Selection

Consider the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_0 + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{y}$ is the n-vector of response, $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_p)$ is the $n \times p$ design matrix, $\boldsymbol{\beta}_0$ is the p-vector of regression coefficients, and $\boldsymbol{\varepsilon}$ is the n-vector of model error which is independent of $\mathbf{X}$. We assume that $\boldsymbol{\beta}_0$ is sparse and has only $s = \|\boldsymbol{\beta}_0\|_0$ nonzero elements. Here, both s and p can diverge with n but we suppress their dependence if no confusion. We next propose a greedy algorithm and give an example to illustrate how it works.

2.1 SAMS algorithm and a broad overview

Like bidirectional elimination, stable adaptive model selection performs three tasks: it adds variables, estimates them, and if any previously added variables become insignificant, drops them. These three tasks are handled by two processes: thresholded least angle regression performs variable selection, and thresholded ordinary least squares performs estimation and variable elimination. Two key aspects of the algorithm are how it adaptively works with residuals and how the threshold τ serves as a test of significance. In fact, τ is the only tuning parameter for the algorithm.

SAMS first brings variables into the model using thresholded LARS. This modified least angle regression procedure always uses the residuals produced by SAMS at the previous step as the dependent variable: $\mathbf{r}^k = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{(k-1)}$, where $\mathbf{y}$ is the original response variable, $\mathbf{X}$ is the predictor vector, and $\widehat{\boldsymbol{\beta}}^{(k-1)}$ is the previous estimated coefficient vector from SAMS. Otherwise, thresholded LARS differs from LARS in only one way: the thresholded algorithm stops when it finds the estimate whose magnitude first equals a predetermined threshold τ. This variable is then selected into the SAMS model.

Next, SAMS estimates the coefficients of variables chosen by thresholded LARS and drops weak ones by using thresholded OLS. Ordinary least squares provides estimates using the dependent variable $\mathbf{y}$, but the τ threshold is incorporated so that should any estimate fall below τ, the corresponding variable is taken out of the SAMS model. This model produced by thresholded least squares regression gives us the $\widehat{\boldsymbol{\beta}}$ estimates. We use these to form the residuals $\mathbf{r} = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$ in the next iteration, which begins again from thresholded least angle regression.

We present the algorithm after explaining some notation. Variables selected into the SAMS model are active variables. They are in the SAMS active set, denoted by $\mathcal{A}$. These active variables are separate from the active variables selected into the LARS model during the thresholded LARS step, so we denote the LARS active variables by $\mathcal{B}$. Additionally, $\mathcal{I}$ is the set of variables not in the LARS active set. The threshold is τ. The coefficients estimated by SAMS are represented by the vector $\widehat{\boldsymbol{\beta}} = (\widehat{\beta}_1, \cdots, \widehat{\beta}_p)^T$. The vector $\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$ forms the residuals $\mathbf{r}$. The coefficients estimated by least angle regression and ordinary least squares that run inside SAMS are, respectively, the vector $\widetilde{\boldsymbol{\beta}} = (\widetilde{\beta}_1, \cdots, \widetilde{\beta}_p)^T$ and the vector $\breve{\boldsymbol{\beta}} = (\breve{\beta}_1, \cdots, \breve{\beta}_p)^T$.

For a given set $\mathcal{A}$, let $\mathbf{X}_{\mathcal{A}}$ be the submatrix formed by columns of $\mathbf{X}$ with indices in $\mathcal{A}$, and let $\boldsymbol{\beta}_{\mathcal{A}}$ be the subvector formed by active coordinates of $\boldsymbol{\beta}$. For a given vector of measured responses $\mathbf{y}$ and data matrix $\mathbf{X}$, the SAMS algorithm is described below.

SAMS Algorithm

0. Initialize variables for SAMS:
   the active set is the empty set, $\mathcal{A} = \emptyset$;
   the SAMS coefficients are zero, $\widehat{\boldsymbol{\beta}} = \mathbf{0}$;
   and the residuals are the response vector, $\mathbf{r} = \mathbf{y}$.

1. Thresholded LARS

   (a) Initialize variables for least angle regression:
   the response variable is the residual SAMS vector $\mathbf{r}$,
   while the LARS active set, inactive set, and estimated vector are the usual
   $\mathcal{B} = \emptyset$,
   $\mathcal{I} = \{1, \cdots, p\}$,
   and $\widetilde{\boldsymbol{\beta}} = 0$.

   (b) Run least angle regression until one LARS active coefficient first reaches the threshold:
   $|\widetilde{\beta}_i| = \tau$ for some $i \in \mathcal{B}$.

   (c) If least angle regression reaches the full OLS solution, skip to step 5.

   (d) Else, add the variable that first reached the threshold to the SAMS active set:
   $\mathcal{A} \leftarrow \mathcal{A} \cup i$;

$\mathcal{I} \leftarrow \mathcal{I} \backslash i$.

2. Thresholded OLS

   (a) Get least square estimate $\breve{\boldsymbol{\beta}}$ by regressing $\mathbf{y}$ on the set of active variables $\mathbf{X}_{\mathcal{A}}$.

   (b) Move variables with estimates below the threshold out of the active set.
   For $i : |\breve{\beta}_i| < \tau$, update $\mathcal{A} \leftarrow \mathcal{A} \backslash i$ and $\mathcal{I} \leftarrow \mathcal{I} \cup i$.

   (c) If any variables were dropped in (b), update the OLS estimate:
   $\breve{\boldsymbol{\beta}} = \mathbf{y} \sim \mathbf{X}_{\mathcal{A}}$.

3. Update the SAMS estimate and residual:
   $\widehat{\boldsymbol{\beta}} = \breve{\boldsymbol{\beta}}$;
   $\mathbf{r} = \mathbf{y} - \mathbf{X}_{\mathcal{A}} \widehat{\beta}_{\mathcal{A}}$.

4. If after steps 1 and 2, new variables have been added to $\mathcal{A}$, repeat from step 1.

5. Output the SAMS estimate, $\widehat{\boldsymbol{\beta}}$.

2.2 An example using stable adaptive model selection

We use a simple example to demonstrate how SAMS estimates and returns important variables. We use the notation from the algorithm. Say in a set of ten covariates that X_1, X_3, and X_5 are the true variables. Let the true coefficients of these variables be respectively $(0.8, -0.5, 0.7)$. Then ideally τ should be less than but close to 0.5. For now, let us set τ at 0.3 for illustrative purposes. Our initial estimate, $\widehat{\boldsymbol{\beta}}$, is the null model with 0 in all ten entries. Therefore, the SAMS active set $\mathcal{A}$ is $\emptyset$.

Each time thresholded LARS runs, its coefficients $\widetilde{\boldsymbol{\beta}}$ begin as the $\mathbf{0}$ vector. Also, since our method works with the residual $\mathbf{r}$, we let the first residual vector be $\mathbf{r} = \mathbf{y}$. LARS initially increases the coefficient estimate of X_1. Say it estimates $\widetilde{\beta}_1 = 0.1$. Then it picks up X_5 as well and increases both coefficients to $\widetilde{\boldsymbol{\beta}}_{\{1,5\}} = (0.2, 0.1)$. Next, it adds X_3 so that $\widetilde{\boldsymbol{\beta}}_{\{1,3,5\}} = (0.5, -0.2, 0.4)$. Our method departs from LARS here. Instead of estimating more variables as least angle regression would, SAMS realizes that both $\widetilde{\beta}_1$ and $\widetilde{\beta}_5$ exceed 0.3. It then finds the $\widetilde{\boldsymbol{\beta}}$ where the first variable, X_1, passed the threshold and returns the other variables to 0. After this process, we obtain $\widetilde{\boldsymbol{\beta}}_{\{1,3,5\}} = (0.3, 0, 0)$. Upon discovering that X_1 is the first variable to exceed τ, SAMS adds the index 1 to the active set $\mathcal{A}$.

In what follows, we disregard the LARS estimates. Instead, we focus on the active variables in $\mathcal{A}$ for thresholded ordinary least squares. OLS regresses $\mathbf{r}$ on $\mathbf{X}_{\mathcal{A}}$. This is currently equivalent to regressing $\mathbf{y}$ on X_1. The estimate is $\breve{\beta}_1 = 0.5$. Since the size of the estimate lies above τ, it need not be thresholded back to 0. If the OLS estimate did fall below 0.3, it would drop out of the active set. This thresholded ordinary least squares estimate is taken as the SAMS estimate $\widehat{\boldsymbol{\beta}}$.

This first iteration updated the $\mathcal{A}$ set and $\widehat{\boldsymbol{\beta}}$ parameters, so that the residual is updated to $\mathbf{r} = \mathbf{y} - 0.5\mathbf{x}_1$. Using this new $\mathbf{r}$ as the response, thresholded least angle regression begins anew. The algorithm repeats.

After just three iterations, our Theorem 2 suggests that SAMS will recover the three true variables so that $\mathcal{A} = \{1, 3, 5\}$. SAMS produces the estimates $\widehat{\boldsymbol{\beta}}_{\mathcal{A}} = (0.8, -0.5, 0.7)$. In general, we do not expect a perfect match to the true coefficients, but we simplify estimates for the example.

Then, it is interesting to consider what SAMS does after only noise variables are left. The algorithm returns to thresholded least angle regression. Thresholded LARS starts afresh, using as its response variable the updated residual containing only noise. Theoretically, both noise variables and active variables in $\mathcal{A}$ are independent of the residual once the true variables have been extracted. As a result, LARS is both unlikely to pick a variable already in $\mathcal{A}$ and also unlikely to find estimates with magnitude larger than τ. Thus, SAMS will terminate and take the vector of estimates $\widehat{\boldsymbol{\beta}}_{\mathcal{A}}$ from the previous ordinary least squares step as its final solution. The early stopping property of SAMS will be formally discussed in Theorem 1.

2.3 How to choose and interpret the threshold

So far we have assumed that we know which threshold τ to use. Though some instances exist when this is the case, we otherwise choose τ through validation. After running SAMS on a training set over a grid of possible thresholds to produce different sets of estimated $\widehat{\boldsymbol{\beta}}$, we choose the vector of estimates which produces the smallest prediction error on a validation set. In this way, we tune τ for our Section 4 simulations and Section 5 data application. The cross validation method can also be used to tune τ.

A potential problem arises with tuning: SAMS becomes an algorithm with three nested loops. SAMS iterates as a loop, LARS iterates as a loop within SAMS, and now for each τ we additionally re-run SAMS. Though it may appear computationally daunting, a few factors speed up performance. Least angle regression is already a quick algorithm because of its geometric properties. SAMS additionally halts LARS early using τ. The grid for τ also does not need to be very fine because the threshold is robust. We say robust in the sense that τ is equally effective anywhere between the smallest support coefficient and the noise level. In our example, we set τ to 0.3 but we could have set it slightly below 0.3 or up to 0.5 without affecting the results.

This threshold is the only tuning parameter in our model and its interpretation deserves some mention. The role τ plays is like that of a critical value in traditional hypothesis testing. Estimates above τ are judged significant and those below τ are considered the mere byproduct of random fluctuations in data. In our example, $\tau = 0.3$ means that it is sufficiently unlikely that any estimate above 0.3 would be due to noise alone. The threshold τ also gives us a sense of the size of regressors' effects on y. The tuning parameter is thus directly related to the estimates which we seek to understand.

Through the example, we suggested that stable adaptive model selection has desirable properties. We indicated that it adds true variables first and terminates immediately after estimating all significant variables. We follow up with theoretical justification to explain why SAMS should perform in these desirable ways.

3 Theoretical properties

In this section, we present theoretical results which can provide insights and help us understand the SAMS method. Although SAMS is built on the LARS algorithm, to simplify the presentation we consider a sightly different setting and prove our theorems based on the Lasso method. Without loss of generality, we standardize the columns of $\mathbf{X}$ to have L_2-norm equal to $\sqrt{n}$. The Lasso method estimates the true coefficient vector $\boldsymbol{\beta}_0$ by minimizing the following regularization problem

$$L_n(\boldsymbol{\beta}; \lambda) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1, \tag{2}$$

where $\lambda \geq 0$ is the regularization parameter controlling model complexity. All theoretical results are conditional on the design matrix $\mathbf{X}$.

As in Fan and Lv (2013), we introduce the definition of the robust spark.

Definition 1. *The robust spark κ_c of the $n \times p$ design matrix $\mathbf{X}$ is defined as the smallest possible positive integer such that there exists an $n \times \kappa_c$ submatrix of $n^{-1/2}\mathbf{X}$ having a singular value less than a given positive constant c.*

The robust spark κ_c is always a positive integer no larger than $n+1$ and can be some large number diverging with n. In fact, it has been proven in Fan and Lv (2013) that if the design matrix $\mathbf{X}$ is generated from a Gaussian distribution, then with asymptotic probability 1, there exist positive constants c and $\tilde{c}$ such that the robust spark $\kappa_c \geq \tilde{c}\sqrt{n/(\log p)}$. We make the following assumption on the true model size s and the threshold level τ.

Condition 1. *It holds that $s = o(n/\log p)$ and $s < \kappa_c$ for some constant $c > 0$. The threshold τ is chosen such that $\tau\sqrt{n/(s \log p)} \to \infty$.*

Let $\lambda_0 = c_0\sqrt{(\log p)/n}$ with c_0 being some positive constant. Define the event

$$\mathcal{E}_1 = \{\frac{1}{n}\|\mathbf{X}^T\boldsymbol{\varepsilon}\|_\infty \leq \lambda_0\}. \tag{3}$$

We make the following assumption on the model error distribution.

Condition 2. *It holds that $P(\mathcal{E}_1) = 1 - O(p^{-c_1})$ for some positive constant c_1 that can be sufficiently large for large enough c_0 .*

Condition 2 was also imposed in Fan and Lv (2013). As discussed therein, it holds for Gaussian, bounded, or light-tailed error distributions, with no or mild conditions on design matrix $\mathbf{X}$.

For a given regularization parameter $\lambda > 0$, denote by $\widehat{\boldsymbol{\beta}}^{\lambda}$ a global minimizer of (2). In implementation, the Lasso solution is calculated over a range of regularization parameters $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, which creates a sequence of candidate models. The upper bound $\lambda_{\max}$ corresponds to the sparsest solution and can be chosen as a large enough number such that $\widehat{\boldsymbol{\beta}}^{\lambda_{\max}} = \mathbf{0}$, and the lower bound $\lambda_{\min}$ corresponds to the most dense solution $\widehat{\boldsymbol{\beta}}^{\lambda_{\min}}$ in this range. The following theorem describes that if the true regression coefficient $\boldsymbol{\beta}_0 = \mathbf{0}$, then the active set selected by SAMS is empty with overwhelming probability, which is consistent variable selection.

Theorem 1. *Assume that Conditions 1 and 2 hold and the true regression coefficient vector $\boldsymbol{\beta}_0 = \mathbf{0}$. If $\lambda_{\min}$ is chosen so that the corresponding Lasso solution satisfies*

$\|\widehat{\boldsymbol{\beta}}^{\lambda_{\min}}\|_0 \leq \min\{\kappa_c, c^2\tau^2 n/(4c_0^2 \log p)\}$, *then with probability at least* $1 - O(p^{-c_1})$, *for all* $\lambda \geq \lambda_{\min}$, *we have* $\widehat{\boldsymbol{\beta}}^{\lambda} = \mathbf{0}$ *and thus the active set of SAMS is always empty.*

Theorem 1 also indicates that our algorithm stops automatically when all true variables are recruited into the model. To understand this, note that in each step of the algorithm, we work with residuals. As remarked in Bühlmann and van de Geer (2011) (section 2.5), the Lasso estimated model with the regularization parameter λ on the order of $\sqrt{(\log p)/n}$ has the variable screening property under some conditions on the design matrix and the signal strength, where the variable screening property means that the Lasso estimated model supp$(\widehat{\boldsymbol{\beta}}^{\lambda})$ includes the true model supp$(\boldsymbol{\beta}_0)$ with overwhelming probability. If after some steps all true variables are included in the model, then the residual from the least squares fit is uncorrelated with any predictor. So in some sense, with the residual as the new response, the underlying population model reduces to a linear regression model with $\boldsymbol{\beta}_0 = \mathbf{0}$. Then according to Theorem 1, with overwhelming probability, our algorithm will not select any additional variables and therefore stops automatically.

The constraint in Theorem 1 on $\|\widehat{\boldsymbol{\beta}}^{\lambda_{\min}}\|_0$ is equivalent to assuming that the minimum regularization parameter $\lambda_{\min}$ should not be too small. We remark that even for a very small regularization parameter, SAMS can still enjoy the property of model selection consistency. To understand this, note that if by chance, some noise variable enters the Lasso solution path and has a coefficient exceeding τ, then it will enter the active set. However, thanks to the least squares refit step, the refitted coefficient of this noise variable will still be less than τ with asymptotic probability one. Thus, it will be removed from the active set with asymptotic probability one and the active set remains empty.

We next show that the first variable selected by the algorithm is the true one with significant probability. For any set $\mathcal{S} \subsetneqq \{1, \cdots, p\}$, let $\mathbf{X}_{\mathcal{S}}$ be the submatrix formed by columns of $\mathbf{X}$ in $\mathcal{S}$. Define the event

$$\mathcal{E}_2 = \{\|(\mathbf{X}_{\mathcal{S}_0}^T \mathbf{X}_{\mathcal{S}_0})^{-1} \mathbf{X}_{\mathcal{S}_0}^T \boldsymbol{\varepsilon}\|_\infty \leq \delta_n, \quad \frac{1}{n}\|\mathbf{X}_{\mathcal{S}_0^c}^T (I_n - \mathbf{P}_{\mathcal{S}_0})\boldsymbol{\varepsilon}\|_\infty \leq \lambda_0\}, \tag{4}$$

where $\delta_n = c_2\sqrt{(\log n)/n}$ with $c_2 > 0$ some constant, I_n is the $n \times n$ identity matrix, and $\mathbf{P}_{\mathcal{S}_0} = \mathbf{X}_{\mathcal{S}_0}(\mathbf{X}_{\mathcal{S}_0}^T \mathbf{X}_{\mathcal{S}_0})^{-1} \mathbf{X}_{\mathcal{S}_0}$ is the projection matrix. We need the following condition on (4).

Condition 3. *It holds that* $P(\mathcal{E}_2) > 1 - o(n^{-c_3})$ *with* $c_3 > 0$ *some constant.*

Similar to Condition 2, Condition 3 holds for bounded or Gaussian errors without any extra assumptions and holds for unbounded non-Gaussian errors with mild assumptions on the design matrix. See the appendix for more detailed discussions on Condition 3.

We further introduce the stability neighborhood as

$$N_{\delta_n}(\boldsymbol{\beta}_0) = \{\boldsymbol{\beta} \in \mathbf{R}^p : \text{supp}(\boldsymbol{\beta}) = \text{supp}(\boldsymbol{\beta}_0), \|\boldsymbol{\beta} - \boldsymbol{\beta}_0\|_\infty \le \delta_n\}.$$

Then $N_{\delta_n}(\boldsymbol{\beta}_0)$ defines a neighborhood around the true regression coefficient $\boldsymbol{\beta}_0$. For each $\boldsymbol{\beta}_1 \in N_{\delta_n}(\boldsymbol{\beta}_0)$ and each $\lambda > 0$, define the deterministic vector

$$\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda) = \arg\min\{(2n)^{-1}\|\mathbf{X}\boldsymbol{\beta}_1 - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\}. \tag{5}$$

Then $\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda)$ is the population version of the solution to (1) when the underlying true regression coefficient is $\boldsymbol{\beta}_1$. The intuition for defining $N_{\delta_n}(\boldsymbol{\beta}_0)$ and $\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda)$ is that a good variable selection procedure should enjoy the stability property and thus $\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda)$ and $\boldsymbol{\beta}^*(\boldsymbol{\beta}_0, \lambda)$ are expected to be close if $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_0$ are close to each other.

For a given threshold τ, define $\lambda^*(\boldsymbol{\beta}_1, \tau)$ as the largest λ such that $\|\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda)\|_\infty$ just increases to τ. For the ease of presentation, we drop the dependence of $\lambda^*(\boldsymbol{\beta}_1, \tau)$ on $\boldsymbol{\beta}_1$ and τ and write it as λ^* whenever there is no confusion. For each $\boldsymbol{\beta}_1 \in N_{\delta_n}(\boldsymbol{\beta}_0)$, define

$$\mathcal{A}^*(\boldsymbol{\beta}_1, \lambda^*) = \text{supp}\big(\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda^*)\big).$$

Then by the definition of λ^*, the set $\mathcal{A}^*(\boldsymbol{\beta}_1, \lambda^*)$ has at least one element. Define

$$\mathcal{S}_0 = \text{supp}(\boldsymbol{\beta}_0)$$

as the support of the true variables. We make the following assumption on the stability property of the population version of the algorithm.

Condition 4. *For any $\boldsymbol{\beta}_1 \in N_{\delta_n}(\boldsymbol{\beta}_0)$, the vector $\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda^*) = (\beta_1^*(\boldsymbol{\beta}_1, \lambda^*), \cdots, \beta_p^*(\boldsymbol{\beta}_1, \lambda^*))^T$ satisfies that $|\beta_j^*(\boldsymbol{\beta}_1, \lambda^*)| \le (1 - c_4)\tau$ if $j \in \mathcal{S}_0^c$ for some constant $c_4 > 0$ independent of $\boldsymbol{\beta}_1$. In addition, for every $\boldsymbol{\beta}_1 \in N_{\delta_n}(\boldsymbol{\beta}_0)$, it holds that $|\mathcal{A}^*(\boldsymbol{\beta}_1, \lambda^*)| < \kappa_c$, where κ_c is the robust spark of the design matrix $\mathbf{X}$. Moreover, $|\mathcal{S}_0^c \cap \mathcal{A}^*(\boldsymbol{\beta}_1, \lambda^*)| < \big(cc_4\tau/\lambda_0\big)^2$.*

Condition 5. *It holds that $\tau^{-1}\min_{j\in\mathcal{S}_0}|\beta_{0j}| \to \infty$.*

We remark that the order of λ^* defined above is generally greater than $O(\sqrt{(\log p)/n})$. To understand this, note that under the restricted eigenvalue condition, using similar arguments as in Bickel et al. (2009) it can be proved that

$$\|\boldsymbol{\beta}_1 - \boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \tilde{\lambda})\|_2 \le O(\sqrt{s(\log p)/n})$$

for the regularization parameter $\tilde{\lambda}$ of the order $\sqrt{(\log p)/n}$. Moreover, since $\boldsymbol{\beta}_1 \in N_{\delta_n}(\boldsymbol{\beta}_0)$ with $\delta_n = c_2\sqrt{(\log n)/n}$, it follows that $\boldsymbol{\beta}_1$ also satisfies the minimum signal strength condition $\tau^{-1} \min_{j\in\mathcal{S}_0} |\beta_{1j}| \to \infty$. The above two results together with Condition 1 ensure that both $\|\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \tilde{\lambda})\|_\infty$ and $\|\boldsymbol{\beta}_1\|_\infty$ have orders larger than τ. Thus, $\lambda^* = \lambda^*(\boldsymbol{\beta}_1, \tau)$ should be of an order greater than $\tilde{\lambda} = O(\sqrt{(\log p)/n})$. The exact order of $\lambda^*(\boldsymbol{\beta}_1, \tau)$ can be obtained in the orthogonal design case, where $\mathbf{X}^T\mathbf{X} = nI_n$. In this case, it can be derived that

$$\boldsymbol{\beta}_{1\mathcal{A}^*} = \operatorname{sgn}\big(\boldsymbol{\beta}^*_{\mathcal{A}^*}(\boldsymbol{\beta}_1, \lambda^*)\big) \circ \big(\big|\boldsymbol{\beta}^*_{\mathcal{A}^*}(\boldsymbol{\beta}_1, \lambda^*)\big| + \lambda^*\big),$$

where $\circ$ stands for the Hadamard product of two vectors. Thus, the value of λ^* is $\max_{j\in\mathcal{S}_0} |\beta_{1j}| - \tau$, which is of a order larger than $O(\sqrt{(\log p)/n})$. Condition 4 puts constraints on the population solution $\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda^*)$ for all parameters in the neighborhood $N_{\delta_n}(\boldsymbol{\beta}_0)$, and thus it is on the stability of the variable selection procedure.

Note that the LARS algorithm, which works with correlation, can be used to solve the Lasso problem (5). Variables in the active set of LARS have larger correlations with the response than variables outside of it. From the point of view of the LARS algorithm, as $\tau \to 0$, the above Condition 4 can be understood as the correlation condition. Intuitively, it assumes that the response $\mathbf{X}\boldsymbol{\beta}_1$ has larger correlations with some signal covariates than with any of the noise covariates so that some signal covariates enter the model ahead of noise covariates. In general when $\tau > 0$, Condition 4 is weaker than the correlation condition. It accommodates the case where some noise variables enter the solution path first but their coefficients increase to τ slower than those for signal variables afterwards.

Condition 6. *It holds that*

$$\sup_{\boldsymbol{\beta}_1\in N_{\delta_n}(\boldsymbol{\beta}_0)} \left\{ \frac{1}{\lambda^*} \max_{j\in\mathcal{A}^{*c}} \left| n^{-1}\mathbf{x}_j^T\mathbf{X}(\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda^*) - \boldsymbol{\beta}_1) \right| \right\} \le 1 - c_5,$$

where $c_5 \in (0, 1)$ is some constant independent of $\boldsymbol{\beta}_1$.

Condition 6 is a deterministic condition on the population solution $\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda^*)$. It is a stronger version of the KKT condition. To understand this, note that by the definitions of λ^* and $\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda^*)$ and the KKT conditions, we obtain that for every $\boldsymbol{\beta}_1 \in N_{\delta_n}(\boldsymbol{\beta}_0)$,

$$\max_{j\in\mathcal{A}^{*c}} \left| n^{-1}\mathbf{x}_j^T\mathbf{X}\big(\boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda^*) - \boldsymbol{\beta}_1\big) \right| \le \lambda^*.$$

Condition 6 assumes that this result holds uniformly over all $\boldsymbol{\beta}_1 \in N_{\delta_n}(\boldsymbol{\beta}_0)$ with a slightly smaller upper bound.

We also need the following condition, which is on the collinearity level of predictors.

Condition 7. *It holds that*

$$\sup_{\boldsymbol{\beta}_1 \in N_{\delta_n}(\boldsymbol{\beta}_0)} \left\{ \frac{1}{\lambda^*(\boldsymbol{\beta}_1, \tau)} \sqrt{|\mathcal{S}_0^c \cap \mathcal{A}^*|} \left\| n^{-1} \mathbf{X}_{\mathcal{A}^{*c}}^T \mathbf{X}_{\mathcal{A}^*} \right\|_{\infty,2} \right\} = o\big(1/\lambda_0\big),$$

where $\|\cdot\|_{\infty,2}$ *is the norm defined as* $\|\mathbf{A}\|_{\infty,2} = \sup_{\{\|\mathbf{a}\|_2=1\}} \|\mathbf{A}\mathbf{a}\|_\infty$ *for matrix* $\mathbf{A}$ *and vector* $\mathbf{a}$ *of appropriate dimensions.*

As discussed above, when the minimum signal condition (Condition 5) is satisfied, the regularization parameter λ^* is of an order larger than $\lambda_0 = c_0\sqrt{(\log p)/n}$. If the active set $\mathcal{A}^* = \mathcal{A}^*(\boldsymbol{\beta}_1, \tau)$ in the population algorithm contains none of the noise variables, then $\mathcal{S}_0^c \cap \mathcal{A}^* = \emptyset$ and Condition 7 is satisfied automatically; if $\mathcal{A}^*$ contains some noise variables, then the cardinality $|\mathcal{S}_0^c \cap \mathcal{A}^*|$ is nonzero and Condition 7 restricts how fast sample correlations among covariates can grow with dimensionality.

As with Condition 5, Conditions 6 and 7 are for all $\boldsymbol{\beta}$ in $N_{\delta_n}(\boldsymbol{\beta}_0)$, and thus they are about the stability of the population algorithm as well.

Proposition 1. *Assume Conditions 1–7 hold and* $\lambda^*(\boldsymbol{\beta}_1, \tau)/\lambda_0 \to \infty$ *for every* $\boldsymbol{\beta}_1 \in N_{\delta_n}(\boldsymbol{\beta}_0)$*. Then with probability at least* $1 - o(n^{-c_3})$*, where* c_3 *is defined in Condition 3, as* λ *decreases to* $\lambda_{\min}$*, the first variable entering the model using SAMS belongs to set* $\mathcal{S}_0$*.*

Although Proposition 1 is about the first variable recruited by our algorithm, it has deeper implications. Just notice that in each step, SAMS works with residuals. By treating the current residuals as the new response variable, if the model satisfies Conditions 2–7, then Proposition 1 guarantees that the next variable entering the active set will be a true variable (i.e., variable in $\mathcal{S}_0$) with asymptotic probability one.

We formally characterize the aforementioned result in the next theorem. For any set $\mathcal{S} \subsetneqq \{1, \cdots, p\}$, let $\boldsymbol{\beta}_\mathcal{S}$ be the subvector formed by entries of $\boldsymbol{\beta}$ in $\mathcal{S}$. Then our model can be written as

$$\mathbf{y} = \mathbf{P}_\mathcal{A}(\mathbf{X}_\mathcal{A}\boldsymbol{\beta}_{0,\mathcal{A}} + \boldsymbol{\varepsilon}) + (I_n - \mathbf{P}_\mathcal{A})(\mathbf{X}_{\mathcal{A}_1}\boldsymbol{\beta}_{0,\mathcal{A}_1} + \boldsymbol{\varepsilon}), \tag{6}$$

where $\mathbf{P}_\mathcal{A} = \mathbf{X}_\mathcal{A}(\mathbf{X}_\mathcal{A}^T\mathbf{X}_\mathcal{A})^{-1}\mathbf{X}_\mathcal{A}^T$ is the projection matrix and $\mathcal{A}_1 = \mathcal{A}^c \cap \mathcal{S}_0$. If the current active set is $\mathcal{A}$, then the response will be regressed on variables in $\mathcal{A}$, and the residual vector becomes

$$\tilde{\mathbf{y}} = (I_n - \mathbf{P}_\mathcal{A})\mathbf{y} = (I_n - \mathbf{P}_\mathcal{A})\mathbf{X}_{\mathcal{A}_1}\boldsymbol{\beta}_{0,\mathcal{A}_1} + (I_n - \mathbf{P}_\mathcal{A})\boldsymbol{\varepsilon}.$$

Define a $(p-|\mathcal{A}^c|)$-dimensional vector $\tilde{\boldsymbol{\beta}}_0$ with $\tilde{\boldsymbol{\beta}}_{0\mathcal{A}_1} = (\mathbf{X}_{\mathcal{A}_1}^T\mathbf{X}_{\mathcal{A}_1})^{-1}\mathbf{X}_{\mathcal{A}_1}^T(I_n-\mathbf{P}_{\mathcal{A}})\mathbf{X}_{\mathcal{A}_1}\boldsymbol{\beta}_{0,\mathcal{A}_1}$ and $\tilde{\boldsymbol{\beta}}_{0\mathcal{A}_1^c} = \mathbf{0}$. It is easy to check that the above residual vector becomes

$$\tilde{\mathbf{y}} = \mathbf{X}_{\mathcal{A}_1}\tilde{\boldsymbol{\beta}}_{0,\mathcal{A}_1} + \tilde{\boldsymbol{\varepsilon}}, \tag{7}$$

where $\tilde{\boldsymbol{\varepsilon}} = (I_n - \mathbf{P}_{\mathcal{A}_1})(I_n - \mathbf{P}_{\mathcal{A}})\mathbf{X}_{\mathcal{A}_1}\boldsymbol{\beta}_{0,\mathcal{A}_1} + (I_n - \mathbf{P}\ _{\mathcal{A}})\boldsymbol{\varepsilon}$ is the new model error. Then the new model (7) has the same form as the original model (1) with different underlying regression coefficient vector and model error vector.

Condition 8. *For any set $\mathcal{A} \subset \mathcal{S}_0$, it holds that $\|n^{-1}\mathbf{X}_{\mathcal{S}_0^c}^T(I_n-\mathbf{P}_{\mathcal{A}_1})(I_n-\mathbf{P}_{\mathcal{A}})\mathbf{X}_{\mathcal{A}_1}\boldsymbol{\beta}_{0,\mathcal{A}_1}\|_\infty \leq c_6\sqrt{(\log p)/n}$, where $c_6 > 0$ is some constant.*

Theorem 2. *Assume $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 I_n)$ with σ^2 as the variance of model error. For a given active set $\mathcal{A} \subsetneqq \mathcal{S}_0$, assume Condition 8 holds and Conditions 1 and 4–7 hold with $\boldsymbol{\beta}_0$ being replaced with $\tilde{\boldsymbol{\beta}}_0$ defined in model (7). Then with probability at least $1 - o(p^{-c_7})$, the next variable that enters the active set using SAMS belongs to set $\mathcal{S}_0 \setminus \mathcal{A}$, where c_7 is some positive constant.*

The following corollary on model selection consistency follows immediately from Theorem 2.

Corollary 1. *Assume Conditions of Theorem 2 hold for every active set $\mathcal{A} \subsetneqq \mathcal{S}_0$. If the number of true covariates $s = \|\boldsymbol{\beta}_0\|_0$ is finite, then SAMS has the property of model selection consistency, i.e., the final set of variables selected by SAMS is equal to the true set $\mathcal{S}_0$ with asymptotic probability one.*

4 Numeric Studies

In this section, we explore how theoretical properties play out in implementation on both simulation and real data. When SAMS runs according to the algorithm of Section 2, it performs well on large sample sizes but deteriorates as n becomes small. For small sample sizes, we modify the previously discussed version of SAMS.

4.1 Implementation

In implementation, we continue to use the residual $\mathbf{r} = \mathbf{y} - \mathbf{X}_{\mathcal{A}}\widehat{\boldsymbol{\beta}}_{\mathcal{A}}$ as the response vector for thresholded LARS, but each time we re-run thresholded LARS, LARS no longer needs to begin from the empty model. Instead, the initial vector of LARS estimates picks up from the last estimates of SAMS: we let LARS start at $\widetilde{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}$. We also let the corresponding initial active set of LARS begin with the variables which are already active in SAMS: initial $\mathcal{B} = \mathcal{A}$. Accordingly, the inactive LARS set begins with $\mathcal{I} = \mathcal{A}^c$.

This version of SAMS performs similarly to the original version of SAMS under large sample sizes and benefits from additional stability as the number of observations shrinks. This modification makes SAMS more adaptive because thresholded LARS makes use of information from the previous thresholded OLS step. Since it is a more sophisticated algorithm that, in the spirit of stable adaptability, is suited to both large and small sample sizes, we use this version in the following simulations and data analysis.

4.2 Simulation

4.2.1 Settings

We examine how stable adaptive model selection performs in two settings. In one environment, we assume noise in the data is light-tailed, and we generate Gaussian errors. In the second setting, we challenge our method's ability to detect signals from much noisier data by drawing errors from a t-distribution.

We first describe the set-up that both environments share. For both settings, we generate 1,000 regressors using a multivariate normal distribution, $N(\mathbf{0}, \boldsymbol{\Sigma})$. The correlation $\boldsymbol{\Sigma}_{jk}$ between any two regressors X_j and X_k is $0.5^{|j-k|}$ so that, as the distance between X_j and X_k increases, the correlation tapers off. We use a linear combination of only ten out of 1,000 variables to generate $\mathbf{y}$ so that the vast majority of regressors are noise. We index the true variables at every third position. Thus, the true variables' indices form the support $\mathcal{S}_0 = \{1, 4, 7, \cdots, 25, 28\}$. The true coefficient vector, $\boldsymbol{\beta}_0$, is non-zero only on the support, and at these locations, we place three small and seven large values: $\boldsymbol{\beta}_{\mathcal{S}} = \{1, -0.5, 0.05, 0.7, -1.2, 0.01, -0.9, -0.01, 0.5, 0.55\}$.

Then, we generate the data from model (1). The dimensionality p is fixed at 1,000 for all our simulations, while the sample size n varies in different cases. The distribution of model error $\boldsymbol{\varepsilon}$ varies according to the descriptions given in the next two subsections.

We use one model fitting procedure for all simulations. In the training stage, we generate $\mathbf{y}$ and $\mathbf{X}$. Then, on this training data, we run SAMS on a sequence of τ

thresholds. We select τ to minimize prediction error on a validation set that has the same characteristics as the training set. The τ produced represents the standard of significance for the model. Once we select τ, we have a final model and calculate performance measures using independently generated test data.

In each simulation, along with our own method, we run five competing methods and the oracle procedure. Of the competing methods, three are the greedy algorithms: forward selection (forward), adaptive forward-backward greedy algorithm (FoBa), and least angle regression (LARS). The other two approaches are regularization methods: Lasso and adaptive Lasso (Ada-Lasso)(Zou, 2006). We include adaptive Lasso because of its theoretical connection to Lasso. To help implement these methods, we use R packages: foba to perform forward and FoBa; and lars to run LARS and Lasso. We develop our own implementation of adaptive Lasso, in which we tune three parameters: $\gamma = \{0.5, 1, 2\}$ as in Zou (2006); the initial weights $\hat{w}$; and λ_n for the final adaptive Lasso coefficients. Lastly, we use the oracle procedure, in which the oracle already knows the sparse set of variables used to generate $\mathbf{y}$. Consequently, the oracle procedure is an ordinary least squares estimation of $\mathbf{y}$ on only $\mathbf{X}_{\mathcal{S}}$, the $n \times 10$ data matrix with the true covariates.

We use standard performance measures: prediction error (PE), L_1-loss, L_2-loss, false positives (FP), and false negatives (FN). Since weak signals are particularly difficult to detect, we present the last measure as two kinds of false negatives: FN-strong for the seven large, or strong, coefficients and FN-weak for the three small, or weak, coefficients. This separation helps explain interesting trade-offs in how these methods perform.

4.2.2 Robust performance under normal errors

In the first setting, we draw errors independently and identically distributed from a normal distribution with mean of 0 and variance of 0.4^2. We generate $n = 80$ observations and then reduce n to 60. These sets of observations demonstrate how methods compare against one another even as available data decreases.

The results are in Table 1. Table 1 tends to group the methods into three collections based on their performance: in the first group are forward, FoBa, and SAMS; in the second group are LARS and Lasso; and in the third group is adaptive Lasso alone. As the data shrinks from 80 to 60 observations, the trend suggests that SAMS becomes better at picking out a much sparser set than the other members of the first group.

Another clear pattern we see is that the prediction error for SAMS increases slower than the errors for other methods. More broadly, for all measures except false negatives, and for both n, SAMS possesses the lowest mean values. Also, for both values of n and for all performance measures, SAMS has the smallest standard errors.

4.2.3 Impact of heavy-tailed errors on accuracy and stability

Because real data is usually noisy, we create a second simulation to examine how SAMS performs on data with heavier tails. We let ε follow a t-distribution with ten degrees of freedom. Though we perform analysis on a sequence of n similar to that of the prior setting, we here describe only the case where $n = 60$. The patterns for all n mimic those of the Gaussian setting, so we summarize performance using the most severe case. Additional tables with the omitted results can be requested from the authors.

Though performance errors increase across all methods, SAMS maintains superior comparative performance and stability. Even in the most challenging case where n is reduced to 60 observations, we see from Table 2 that SAMS remains closer to the oracle's prediction error than any other method. Again, we notice that SAMS exhibits remarkably low standard errors across all metrics.

4.3 Real data analysis

For real data, we look at the publicly available spambase data set from the University of California at Irvine Machine Learning Repository. The data includes 56 email characteristics that we pairwise interact to create 1,540 additional variables that potentially predict spam. The 56 original characteristics are all numeric. They include: word frequency, such as the number of times "free" appears in an email divided by total number of words in the email; symbol frequency, such as the number of times "$" appears in an email divided by the email's total number of characters; or string attributes, such as the longest run of capital letters in an email. After eliminating interactions that occur less that 1% of the time in the data set, we are left with 1,253 total covariates. By using interaction terms, we may discover that emails heavy in words such as "receive" only predict spam if symbols such as "000" also appear frequently. We might see this particular combination crop up in spam about receiving cash rewards.

Spambase was previously examined by Hall et al. (2013) to explore simple tiered

		SAMS	Forward	FoBa	LARS	Lasso	Ada-Lasso	Oracle
$n = 80$	PE	18.23	18.25	18.28	40.53	40.82	22.37	18.08
	$(\times 10^{-2})$	(0.32)	(0.33)	(0.33)	(1.27)	(1.31)	(0.65)	(0.33)
	L_1-loss	35.6	36.06	36.14	235.36	227.74	75.01	38.97
	$(\times 10^{-2})$	(0.87)	(0.92)	(0.96)	(7.62)	(6.85)	(4.45)	(1.02)
	L_2-loss	14.22	14.3	14.34	51.06	51.1	23.5	14.91
	$(\times 10^{-2})$	(0.38)	(0.38)	(0.39)	(1.15)	(1.14)	(0.97)	(0.38)
	FP	0.07	0.13	0.12	41.38	39.54	4.48	0
		(0.03)	(0.04)	(0.04)	(1.50)	(1.42)	(0.46)	(0)
	FN-strong	0	0	0	0	0	0	0
		(0)	(0)	(0)	(0)	(0)	(0)	(0)
	FN-weak	3	3	3	2.79	2.79	2.97	0
		(0)	(0)	(0)	(0.04)	(0.04)	(0.02)	(0)
$n = 60$	PE	21.23	33.19	33.7	76.82	77.3	47.75	18.98
	$(\times 10^{-2})$	(2.18)	(7.36)	(7.36)	(4.97)	(4.93)	(3.69)	(0.36)
	L_1-loss	49.08	71.2	74.47	341.53	336.39	180	44.66
	$(\times 10^{-2})$	(5.28)	(14.7)	(16.92)	(12.46)	(11.83)	(11.64)	(1.14)
	L_2-loss	18.83	24.32	24.81	76.7	76.8	51.18	17.46
	$(\times 10^{-2})$	(1.78)	(3.46)	(3.87)	(2.8)	(2.78)	(2.9)	(0.4)
	FP	0.35	1.46	1.32	38.15	38.18	9.44	0
		(0.17)	(0.90)	(0.85)	(0.89)	(0.96)	(0.78)	(0)
	FN-strong	0.05	0.15	0.18	0.22	0.2	0.35	0
		(0.04)	(0.08)	(0.09)	(0.06)	(0.05)	(0.07)	(0)
	FN-weak	3	2.99	3	2.86	2.87	2.94	0
		(0)	(0.01)	(0)	(0.03)	(0.04)	(0.02)	(0)

Table 1: SAMS's performance against five competitors' and the oracle procedure's when data are generated with Gaussian error. The table shows mean with standard error given in parentheses.

	SAMS	Forward	FoBa	LARS	Lasso	Ada-Lasso	Oracle
PE	29.42	41.61	42.81	90.23	89.24	61.01	24.88
($\times 10^{-2}$)	(4.75)	(7.77)	(8.6)	(5.48)	(5.42)	(4.67)	(0.52)
L_1-loss	53.95	88.33	79.17	356.7	348.8	199.79	51.46
($\times 10^{-2}$)	(7.92)	(19.7)	(17.3)	(13.35)	(12.55)	(13.04)	(1.47)
L_2-loss	20.94	27.85	27.1	81.6	81.56	57.26	19.98
($\times 10^{-2}$)	(2.32)	(3.84)	(3.98)	(3.07)	(3.03)	(3.32)	(0.52)
FP	0.29	2.21	1.23	36.66	36.35	9.33	0
	(0.10)	(1.13)	(0.80)	(1.03)	(1.01)	(0.85)	(0)
FN-strong	0.06	0.22	0.27	0.32	0.30	0.48	0
	(0.05)	(0.08)	(0.11)	(0.07)	(0.06)	(0.09)	(0)
FN-weak	3	3	3	2.84	2.85	2.94	0
	(0)	(0)	(0)	(0.04)	(0.04)	(0.02)	(0)

Table 2: SAMS's performance against six methods' on heavy-tailed data. Data is generated with random noise following a t-distribution with 10 degrees of freedom. We show the case where $n = 60$.

classifiers. In that application, a subset of just five regressors was used. Considering the limited number of features, the classifiers did well. For misclassification rate, the proportion of wrongly classified emails, the simple tiered methods returned errors as low as 13.3%. Here, however, we have the luxury of considering much richer information, so we expect any reasonably good method to give a lower misclassification rate.

Because real data is especially noisy, we make a small modification to the implementation of SAMS. To increase the stability of SAMS in the face of noisy data, we add a second tuning parameter. Since SAMS generates multiple sets of coefficients from when it begins to when it terminates, we can specifically select the set of $\widehat{\boldsymbol{\beta}}$ which yields the lowest misclassification rate. The difference is that in the original procedure, for each value of threshold τ, we take the estimate of $\widehat{\boldsymbol{\beta}}$ as the last $\widetilde{\boldsymbol{\beta}}$ in the LARS solution path, but here we may select an intermediate solution as $\widehat{\boldsymbol{\beta}}$.

Real data also renders it too computationally and memory intensive to tune adaptive Lasso for the best triplet $(\gamma, \hat{w}, \lambda_n)$. Instead, we create the initial weights $\hat{w}$ from the tuned Lasso parameters and then tune the remaining pair of parameters (γ, λ_n).

To use the 4,601 emails in the data set, we split them into a one-fifth testing set,

	SAMS	Forward	FoBa	LARS	Lasso	Ada-Lasso
MCR	10.16	11.88	11.8	11.53	11.35	11.34
	(0.11)	(0.12)	(0.13)	(0.1)	(0.11)	(0.12)

Table 3: Predicting Spam. MCR stands for the misclassification rate. Values are reported as percents. The table shows mean with standard error given in parentheses.

two-fifths validation set, and two-fifths training set. On the training set of 1,840 emails, we generate candidate models using the methods described in the simulation section. We predict that an email is spam if a model produces a value greater than 0.5 and not spam otherwise. On the validation set, we choose tuning parameters to minimize misclassification rate, and we measure the error on the testing set. We repeat the procedure on 100 random splits and average the results. The average misclassification rate is shown in Table 3.

Table 3 shows that SAMS has the lowest misclassification rate at 10.16% and is followed by adaptive Lasso, which has a misclassification slightly more than 1% higher. SAMS produces a misclassification rate lower than the 13.3% found in Hall et al. (2013). SAMS also exhibits the second lowest standard error, a slim 0.01% higher than the standard error of LARS.

Particularly interesting in this application are the distinct email characteristics that different methods believe indicate spam. SAMS tends to be very selective, and most of its variables appear reasonable. Consistently in over 95% of the runs, SAMS indicates that higher frequency of “$”, greater volumes of “000”, bigger font size, more of the word “free”, and increased number of “!” indicate spam emails. SAMS is also more conservative than other methods when selecting important interaction terms.

As a whole, the data indicates that, amongst these methods, SAMS does the best job at achieving two desirable outcomes: correctly sorting email and selecting a set of high-signal, stable spam indicators out of an overwhelming number of possibilities.

A Proofs of main results

A.1 Proof of Theorem 1

We prove the theorem conditioning on event $\mathcal{E}$, which happens with probability at least $1 - o(p^{-c_1})$, as guaranteed by Condition 2. The key is to prove that conditioning on $\mathcal{E}_1$, we have

$$\|\widehat{\boldsymbol{\beta}}^{\lambda}\|_\infty < \tau, \tag{8}$$

for any $\lambda \in [\lambda_{\min}, \infty)$. Then the result in the theorem follows automatically.

We proceed to prove (8). For a given λ, a vector $\widehat{\boldsymbol{\beta}}^{\lambda} = (\widehat{\beta}_1^{\lambda}, \cdots, \beta_p^{\lambda})^T \in \mathbf{R}^p$ is the global minimizer of (2) if and only if the following KKT conditions are satisfied

$$\begin{aligned} &- n^{-1}\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\lambda}) + \lambda \mathrm{sgn}(\widehat{\beta}_j^{\lambda}) = 0, \text{ for } j \in \mathrm{supp}(\widehat{\boldsymbol{\beta}}^{\lambda}) \\ &|\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\lambda})| \le \lambda, \text{ for } j \in \mathrm{supp}(\widehat{\boldsymbol{\beta}}^{\lambda})^c. \end{aligned}$$

Since $\boldsymbol{\beta}_0 = \mathbf{0}$, the above conditions are equivalent to

$$- n^{-1}\mathbf{x}_j^T(\boldsymbol{\varepsilon} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\lambda}) + \lambda \mathrm{sgn}(\widehat{\beta}_j^{\lambda}) = 0, \text{ for } j \in \mathrm{supp}(\widehat{\boldsymbol{\beta}}^{\lambda}) \tag{9}$$

$$|\mathbf{x}_j^T(\boldsymbol{\varepsilon} - \mathbf{X}\widehat{\boldsymbol{\beta}}^{\lambda})| \le \lambda, \text{ for } j \in \mathrm{supp}(\widehat{\boldsymbol{\beta}}^{\lambda})^c. \tag{10}$$

Thus, if $\lambda \ge c_0\sqrt{(\log p)/n}$ with c_0 in (3), then the Lasso solution $\widehat{\boldsymbol{\beta}}^{\lambda} = \mathbf{0}$ conditioning on the event $\mathcal{E}_1$. Therefore, the desired results hold for $\lambda \in [c_0\sqrt{(\log p)/n}, \infty)$.

Next we consider the range $\lambda \in [\lambda_{\min}, c_0\sqrt{(\log p)/n})$. We use the method of proof by contradiction. Suppose that as λ decreases to some $\lambda_1 \in [\lambda_{\min}, c_0\sqrt{(\log p)/n})$, there exists a coordinate j_1 such that the magnitude of the j_1th coordinate of $\widehat{\boldsymbol{\beta}}^{\lambda_1}$ just increased to τ. Then by the continuity of Lasso solution path, it is known that $\|\widehat{\boldsymbol{\beta}}^{\lambda_1}\|_\infty \le \tau$. Let $A = \mathrm{supp}(\widehat{\boldsymbol{\beta}}^{\lambda_1})$, i.e., the support of $\widehat{\boldsymbol{\beta}}^{\lambda_1}$. Using matrix notation, the KKT condition (9) can be rewritten as

$$n^{-1}\mathbf{X}_A^T\mathbf{X}_A\widehat{\boldsymbol{\beta}}_A^{\lambda_1} = n^{-1}\mathbf{X}_A^T\boldsymbol{\varepsilon} - \lambda_1 \mathrm{sgn}(\widehat{\boldsymbol{\beta}}_A^{\lambda_1}). \tag{11}$$

We first consider the left hand side of (11). Note that $\widehat{\boldsymbol{\beta}}^{\min}$ is the most dense solution; we have $|A| \le \|\widehat{\boldsymbol{\beta}}^{\min}\|_0 < \kappa_c$. Thus, it follows from the definition of κ_c that $\lambda_{\min}(n^{-1}\mathbf{X}_A^T\mathbf{X}_A) \ge c^2$, which entails

$$\|n^{-1}\mathbf{X}_A^T\mathbf{X}_A\widehat{\boldsymbol{\beta}}_A^{\lambda_1}\|_2 \ge \lambda_{\min}(n^{-1}\mathbf{X}_A^T\mathbf{X}_A)\|\widehat{\boldsymbol{\beta}}_A^{\lambda_1}\|_2 \ge c^2\|\widehat{\boldsymbol{\beta}}_A^{\lambda_1}\|_2 \ge c^2\tau. \tag{12}$$

Now consider the right hand side of (11). On the event $\mathcal{E}_1$, by matrix calculus,

$$\|n^{-1}\mathbf{X}_A^T\boldsymbol{\varepsilon} - \lambda_1 \mathrm{sgn}(\widehat{\boldsymbol{\beta}}_A^{\lambda_1})\|_2 \leq \sqrt{|A|}\|n^{-1}\mathbf{X}_A^T\boldsymbol{\varepsilon} - \lambda_1 \mathrm{sgn}(\widehat{\boldsymbol{\beta}}_A^{\lambda_1})\|_\infty \leq (c_0\sqrt{(\log p)/n} + \lambda_1)\sqrt{|A|}.$$

Combining this with (11) and in view of (12) we have

$$|A| \geq c^4\tau^2(c_0\sqrt{(\log p)/n} + \lambda_1)^{-2} \geq c^4\tau^2 n/(4c_0^2 \log p),$$

which contradicts the theorem assumption. This completes the proof of Theorem 1.

A.2 Proof of Proposition 1

The key is to prove that, conditioning on event $\mathcal{E}_2$ defined in Condition 3, the minimizer $\widehat{\boldsymbol{\beta}}$ of $L_n(\boldsymbol{\beta}; \lambda^*)$ defined in (2) satisfies

$$|\widehat{\boldsymbol{\beta}}_j| < \tau, \text{ for } j \in \mathcal{S}_0^c. \tag{13}$$

Then the variable whose coefficient has magnitude reaching τ must be in the set $\mathcal{S}_0$, and the desired result in Theorem 1 follows automatically.

To prove (13), we need to characterize $\widehat{\boldsymbol{\beta}}$. We achieve this goal by constructing the minimizer $\widehat{\boldsymbol{\beta}}$ and comparing it with the population minimizer $\boldsymbol{\beta}^* = \boldsymbol{\beta}^*(\boldsymbol{\beta}_1, \lambda^*)$, where $\boldsymbol{\beta}_1$ is some vector in $N_{\delta_n}(\boldsymbol{\beta}_0)$ to be introduced later, and $\lambda^* = \lambda^*(\boldsymbol{\beta}_1, \tau)$ is defined in Section 3. We will first prove that the minimizer $\widehat{\boldsymbol{\beta}}$ satisfies $\mathrm{supp}(\widehat{\boldsymbol{\beta}}) \subseteq \mathcal{A}^* \equiv \mathrm{supp}(\boldsymbol{\beta}^*)$, and moreover,

$$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 \leq \sqrt{|\mathcal{S}_0^c \cap \mathcal{A}^*|}\lambda_0/c^2. \tag{14}$$

Then with the above result (14), we can prove (13) using the method of proof by contradiction. Specifically, suppose (13) does not hold and for some $j \in \mathcal{S}_0^c$, $|\widehat{\beta}_j| \geq \tau$. Then by Condition 4, we have $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 \geq |\widehat{\beta}_j - \beta_j^*| \geq c_2\tau$. This together with (14) entails that

$$|\mathcal{S}_0^c \cap \mathcal{A}^*| \geq \big(cc_2\tau/\lambda_0\big)^2,$$

which contradicts Condition 4. Thus, (13) is proved and the result in the theorem follows immediately.

It remains to prove (14). To this end, we first introduce $\boldsymbol{\beta}_1$. Note that the model error $\boldsymbol{\varepsilon}$ can be decomposed into two parts, $\boldsymbol{\varepsilon}_\| = \mathbf{P}_{\mathcal{S}_0}\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}_\perp = (I_n - \mathbf{P}_{\mathcal{S}_0})\boldsymbol{\varepsilon}$, where $\mathbf{P}_{\mathcal{S}_0}$ is the same as in (4). Thus, the regression model can be written as

$$\mathbf{y} = \mathbf{X}_{\mathcal{S}_0}\boldsymbol{\beta}_{0\mathcal{S}_0} + \boldsymbol{\varepsilon}_\| + \boldsymbol{\varepsilon}_\perp = \mathbf{X}_{\mathcal{S}_0}\boldsymbol{\beta}_{1\mathcal{S}_0} + \boldsymbol{\varepsilon}_\perp, \tag{15}$$

where $\boldsymbol{\beta}_{1\mathcal{S}_0} = \boldsymbol{\beta}_{0\mathcal{S}_0} + (\mathbf{X}_{\mathcal{S}_0}^T\mathbf{X}_{\mathcal{S}_0})^{-1}\mathbf{X}_{\mathcal{S}_0}^T\boldsymbol{\varepsilon}$. Let $\boldsymbol{\beta}_1$ be a vector with support $\mathcal{S}_0$ and on its support it is the same as $\boldsymbol{\beta}_{1\mathcal{S}_0}$. Then conditioning on event $\mathcal{E}_2$, $\boldsymbol{\beta}_1 \in N_{\delta_n}(\boldsymbol{\beta}_0)$.

Next we construct $\widehat{\boldsymbol{\beta}}$ and show that it satisfies $\operatorname{supp}(\widehat{\boldsymbol{\beta}}) \subseteq \mathcal{A}^*$ and (14) holds. Consider the objective function

$$Q_n(\boldsymbol{\beta};\lambda) = (2n)^{-1}\|\mathbf{X}\boldsymbol{\beta}_1 - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1.$$

Then $\boldsymbol{\beta}^*$ is the minimizer of $Q_n(\boldsymbol{\beta};\lambda^*)$. By (15) and the definition of $\boldsymbol{\beta}_1$,

$$\begin{aligned}
L_n(\boldsymbol{\beta};\lambda) &= (2n)^{-1}\|\mathbf{X}(\boldsymbol{\beta}_1 - \boldsymbol{\beta}) + \boldsymbol{\varepsilon}_\perp\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \\
&= (2n)^{-1}\|\mathbf{X}(\boldsymbol{\beta}_1 - \boldsymbol{\beta})\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 + (2n)^{-1}\|\boldsymbol{\varepsilon}_\perp\|_2^2 + n^{-1}\boldsymbol{\varepsilon}_\perp^T\mathbf{X}(\boldsymbol{\beta}_1 - \boldsymbol{\beta}) \\
&= Q_n(\boldsymbol{\beta};\lambda) - \sum_{j\in\mathcal{S}_0^c} a_j\beta_j + (2n)^{-1}\|\boldsymbol{\varepsilon}_\perp\|_2^2,
\end{aligned}$$

where $a_j = n^{-1}\boldsymbol{\varepsilon}_\perp^T\mathbf{x}_j$, and in the last step above we have used the decomposition $\mathbf{X}(\boldsymbol{\beta}_1 - \boldsymbol{\beta}) = \mathbf{X}_{\mathcal{S}_0}(\boldsymbol{\beta}_{1\mathcal{S}_0} - \boldsymbol{\beta}_{\mathcal{S}_0}) - \mathbf{X}_{\mathcal{S}_0^c}\boldsymbol{\beta}_{\mathcal{S}_0^c}$ and $\boldsymbol{\varepsilon}_\perp^T\mathbf{X}_{\mathcal{S}_0} = 0$. Denote by $\mathcal{A}^* = \operatorname{supp}(\boldsymbol{\beta}^*)$. We first study the minimizer $\widehat{\boldsymbol{\beta}}$ when restricted to the subspace $\mathbf{R}^{\mathcal{A}^*} = \{\boldsymbol{\beta} = (\beta_1,\cdots,\beta_p)^T : \beta_j = 0 \text{ for } j \notin \mathcal{A}^*\}$. Let $\mathbf{e} = (e_1,\cdots,e_p)^T \in \mathbf{R}^{\mathcal{A}^*}$ be a unit vector satisfying $\|\mathbf{e}\|_2 = 1$. For some $0 \neq \delta \in \mathbf{R}$, by a Taylor expansion and the fact that $\boldsymbol{\beta}_{\mathcal{A}^*}^*$ minimizes $Q_n(\boldsymbol{\beta};\lambda^*)$ when restricted to $\mathbf{R}^{\mathcal{A}^*}$, we have

$$\begin{aligned}
L_n(\boldsymbol{\beta}^* + \delta\mathbf{e};\lambda^*) - L_n(\boldsymbol{\beta}^*;\lambda^*) &= Q_n(\boldsymbol{\beta}^* + \delta\mathbf{e};\lambda^*) - Q_n(\boldsymbol{\beta}^*;\lambda^*) - \delta\sum_{j\in\mathcal{S}_0^c\cap\mathcal{A}^*} a_je_j \qquad (16)\\
&= \delta^2\mathbf{e}^T\nabla^2Q_n(\tilde{\boldsymbol{\beta}}^*;\lambda^*)\mathbf{e} - \delta\sum_{j\in\mathcal{S}_0^c\cap\mathcal{A}^*} a_je_j,
\end{aligned}$$

where $\tilde{\boldsymbol{\beta}}^*$ lies on the line segment connecting $\boldsymbol{\beta}^*$ and $\boldsymbol{\beta}^* + \delta\mathbf{e}$. Thus, it follows that $\operatorname{supp}(\tilde{\boldsymbol{\beta}}^*) \subseteq \mathcal{A}^*$. By Condition 4, we have $\lambda_{\min}(n^{-1}\mathbf{X}_{\mathcal{A}^*}^T\mathbf{X}_{\mathcal{A}^*}) \geq c^2$. Hence,

$$\lambda_{\min}(\nabla^2Q_n(\tilde{\boldsymbol{\beta}}^*;\lambda^*)) = \lambda_{\min}(n^{-1}\mathbf{X}_{\mathcal{A}^*}^T\mathbf{X}_{\mathcal{A}^*}) \geq c^2. \qquad (17)$$

Since $\max_{j\in\mathcal{S}_0^c}|a_j| \leq \lambda_0$ conditioning on $\mathcal{E}_2$, it follows from (16) and (17) that

$$\begin{aligned}
L_n(\boldsymbol{\beta}^* + \delta\mathbf{e};\lambda^*) - L_n(\boldsymbol{\beta}^*;\lambda^*) &\geq c^2\delta^2 - \delta\sum_{j\in\mathcal{S}_0^c\cap\mathcal{A}^*} a_je_j \geq c^2\delta^2 - \delta\{\sum_{j\in\mathcal{S}_0^c\cap\mathcal{A}^*} a_j^2\}^{1/2} \\
&\geq c^2\delta^2 - \delta\lambda_0\sqrt{|\mathcal{S}_0^c\cap\mathcal{A}^*|}.
\end{aligned}$$

Thus, $L_n(\boldsymbol{\beta}^* + \delta\mathbf{e}_j) - L_n(\boldsymbol{\beta}^*;\lambda^*) > 0$ for $|\delta| > \lambda_0\sqrt{|\mathcal{S}_0^c\cap\mathcal{A}^*|}/c^2$. This ensures that the minimizer $\widehat{\boldsymbol{\beta}}_{\mathcal{A}^*}$ of $L_n(\boldsymbol{\beta};\lambda^*)$ restricted on $\mathbf{R}^{\mathcal{A}^*}$ satisfies

$$\|\widehat{\boldsymbol{\beta}}_{\mathcal{A}^*} - \boldsymbol{\beta}_{\mathcal{A}^*}^*\|_2 \leq \sqrt{|\mathcal{S}_0^c\cap\mathcal{A}^*|}\lambda_0/c^2. \qquad (18)$$

Let $\mathcal{A} = \mathrm{supp}(\widehat{\boldsymbol\beta}_{\mathcal{A}^*})$. Then we have $\mathcal{A} \subseteq \mathcal{A}^*$, and when restricted to $\mathbf{R}^{\mathcal{A}^*}$, the estimator $\widehat{\boldsymbol\beta}_{\mathcal{A}}$ will satisfy the following KKT conditions:

$$-n^{-1}\mathbf{x}_j^T\mathbf{X}_{\mathcal{A}^*}(\boldsymbol\beta_{1\mathcal{A}^*} - \widehat{\boldsymbol\beta}_{\mathcal{A}^*}) + \lambda^*\mathrm{sgn}(\widehat{\boldsymbol\beta}_j) - a_j 1_{\{j\in\mathcal{A}\cap\mathcal{S}_0^c\}} = 0, \text{ if } j \in \mathcal{A}, \tag{19}$$

$$|n^{-1}\mathbf{x}_j^T\mathbf{X}_{\mathcal{A}^*}(\boldsymbol\beta_{1\mathcal{A}^*} - \widehat{\boldsymbol\beta}_{\mathcal{A}^*}) + a_j 1_{\{j\in\mathcal{A}^c\cap\mathcal{S}_0^c\}}| \le \lambda^*, \text{ if } j \in \mathcal{A}^* \setminus \mathcal{A}. \tag{20}$$

Now we construct the vector $\widehat{\boldsymbol\beta}$ in such a way that its support is $\mathcal{A}$ and it takes value $\widehat{\boldsymbol\beta}_{\mathcal{A}}$ on its support. We will show that $\widehat{\boldsymbol\beta}$ constructed in this way is indeed the global minimizer of $L_n(\boldsymbol\beta;\lambda^*)$ in the entire parameter space $\mathbf{R}^p$. To this end, we only need to prove

$$|n^{-1}\mathbf{x}_j^T\mathbf{X}(\boldsymbol\beta_1 - \widehat{\boldsymbol\beta}) + a_j 1_{\{j\in\mathcal{A}^c\cap\mathcal{S}_0^c\}}| \le \lambda^*, \text{ if } j \in \mathcal{A}^{*c}. \tag{21}$$

Then (19) – (21) together form the KKT conditions guaranteeing that $\widehat{\boldsymbol\beta}$ defined above is the global minimizer of $L_n(\boldsymbol\beta;\lambda^*)$.

It remains to prove (21). Conditioning on event $\mathcal{E}_2$, we have

$$\max_{j\in\mathcal{A}^{*c}} |a_j 1_{\{j\in\mathcal{A}^c\cap\mathcal{S}_0^c\}}| \le \max_{j\in\mathcal{S}_0^c} |a_j| \le \lambda_0. \tag{22}$$

In addition, by the triangle inequality,

$$\max_{j\in\mathcal{A}^{*c}} |n^{-1}\mathbf{x}_j^T\mathbf{X}(\boldsymbol\beta_1 - \widehat{\boldsymbol\beta})| \le \max_{j\in\mathcal{A}^{*c}} \left|n^{-1}\mathbf{x}_j^T\mathbf{X}(\boldsymbol\beta_1 - \boldsymbol\beta^*)\right| + \max_{j\in\mathcal{A}^{*c}} \left|n^{-1}\mathbf{x}_j^T\mathbf{X}(\boldsymbol\beta^* - \widehat{\boldsymbol\beta})\right|. \tag{23}$$

The first term on the right hand side above is deterministic and bounded by $(1-c_5)\lambda^*$, as assumed in Condition 6. For the second term on the right hand side, since $\mathcal{A} \subset \mathcal{A}^*$, it follows from (14) and Condition 7 that

$$\max_{j\in\mathcal{A}^{*c}} \left|n^{-1}\mathbf{x}_j^T\mathbf{X}(\boldsymbol\beta^* - \widehat{\boldsymbol\beta})\right| \le n^{-1}\|\mathbf{X}_{\mathcal{A}^{*c}}^T\mathbf{X}_{\mathcal{A}^*}\|_{\infty,2}\|\boldsymbol\beta^*_{\mathcal{A}^*} - \widehat{\boldsymbol\beta}_{\mathcal{A}^*}\|_2 = o(\lambda^*).$$

Combining the above results and in view of (22) and (23) we complete the proof of (21). Hence we have proved that $\widehat{\boldsymbol\beta}$ constructed above is a global minimizer of $L_n(\boldsymbol\beta;\lambda^*)$ and satisfies (14). This completes the proof of the proposition.

A.3 Proof of Theorem 2

Note that for any given active set $\mathcal{A} \subsetneqq \mathcal{S}_0$, the model can be written as (7). So we only need to prove that under Condition 8 and the extra distribution assumption $\boldsymbol\varepsilon \sim N(\mathbf{0}, \sigma^2 I_n)$, Conditions 2–3 hold under the new model setting (7) with asymptotic probability one. Then using a similar idea to the proof of Proposition 1, we can prove that the desired results in Theorem 2 hold. In the proof below, we use C to denote some generic positive constant.

It remains to prove that, with asymptotic probability one, the following events hold:

$$\tilde{\mathcal{E}}_1 = \{n^{-1}\|\mathbf{X}^T_{\mathcal{A}^c}\tilde{\boldsymbol{\varepsilon}}\|_\infty \le C\sqrt{(\log p)/n}\}$$
$$\tilde{\mathcal{E}}_2 = \{\|(\mathbf{X}^T_{\mathcal{A}_1}\mathbf{X}_{\mathcal{A}_1})^{-1}\mathbf{X}^T_{\mathcal{A}_1}\tilde{\boldsymbol{\varepsilon}}\|_\infty \le C\sqrt{(\log n)/n}, \quad n^{-1}\|\mathbf{X}^T_{\mathcal{S}_0^c}(I_n - \mathbf{P}_{\mathcal{A}_1})\tilde{\boldsymbol{\varepsilon}}\|_\infty \le C\sqrt{(\log p)/n}\}.$$

We first consider event $\tilde{\mathcal{E}}_1$. By definition of $\tilde{\boldsymbol{\varepsilon}}$ we have

$$n^{-1}\mathbf{X}^T_{\mathcal{A}^c}\tilde{\boldsymbol{\varepsilon}} = A_1 + A_2,$$

where $A_1 = n^{-1}\mathbf{X}^T_{\mathcal{A}^c}(I_n - \mathbf{P}_{\mathcal{A}})\boldsymbol{\varepsilon}$ and $A_2 = n^{-1}\mathbf{X}^T_{\mathcal{A}^c}(I_n - \mathbf{P}_{\mathcal{A}_1})(I_n - \mathbf{P}_{\mathcal{A}})\mathbf{X}_{\mathcal{A}_1}\boldsymbol{\beta}_{0,\mathcal{A}_1}$. Since $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 I_n)$, it follows that $A_1 = n^{-1}\mathbf{X}^T_{\mathcal{A}^c}(I_n - \mathbf{P}_{\mathcal{A}})\boldsymbol{\varepsilon} \sim N(0, n^{-2}\sigma^2\mathbf{X}^T_{\mathcal{A}^c}(I_n - \mathbf{P}_{\mathcal{A}})\mathbf{X}_{\mathcal{A}^c})$. In addition, since each column of $\mathbf{X}$ is standardized to have L_2-norm $\sqrt{n}$, we obtain that each entry of random vector A_1 has a normal distribution with variance bounded from above by σ^2/n. Thus, using classical Gaussian tail probability bounds we can prove that

$$P\big(\|A_1\|_\infty > C\sqrt{(\log p)/n}\big) \le |\mathcal{A}^c| P(|Z| > C\sqrt{(\log p)}) \le Cp\exp(-C\log p)/\sqrt{\log p} = o(p^{-C}),$$

where Z is a standard normal random variable. Thus, with probability at least $1 - o(p^{-C})$, the vector A_1 satisfies $\|A_1\|_\infty \le C\sqrt{(\log p)/n}$ where C is some generic positive constant. Moreover, since $\mathcal{A}^c = \mathcal{A}_1 \cup \mathcal{S}_0^c$ and $\mathbf{X}^T_{\mathcal{A}_1}(I_n - \mathbf{P}_{\mathcal{A}_1}) = \mathbf{0}$, by Condition 8, we have $\|A_2\|_\infty = \|n^{-1}\mathbf{X}^T_{\mathcal{A}^c}(I_n - \mathbf{P}_{\mathcal{A}_1})(I_n - \mathbf{P}_{\mathcal{A}})\mathbf{X}_{\mathcal{A}_1}\boldsymbol{\beta}_{0,\mathcal{A}_1}\|_\infty = \|n^{-1}\mathbf{X}^T_{\mathcal{S}_0^c}(I_n - \mathbf{P}_{\mathcal{A}_1})(I_n - \mathbf{P}_{\mathcal{A}})\mathbf{X}_{\mathcal{A}_1}\boldsymbol{\beta}_{0,\mathcal{A}_1}\|_\infty \le c_6\sqrt{(\log p)/n}$. Combining these results entails that with probability at least $1 - O(p^{-C})$, event $\tilde{\mathcal{E}}_1$ holds.

Next we consider event $\tilde{\mathcal{E}}_2$. By the definition of $\tilde{\boldsymbol{\varepsilon}}$ we have

$$(\mathbf{X}^T_{\mathcal{A}_1}\mathbf{X}_{\mathcal{A}_1})^{-1}\mathbf{X}^T_{\mathcal{A}_1}\tilde{\boldsymbol{\varepsilon}} = B_1 + B_2,$$

where $B_1 = (\mathbf{X}^T_{\mathcal{A}_1}\mathbf{X}_{\mathcal{A}_1})^{-1}\mathbf{X}^T_{\mathcal{A}_1}(I_n - \mathbf{P}_{\mathcal{A}})\boldsymbol{\varepsilon}$ and $B_2 = (\mathbf{X}^T_{\mathcal{A}_1}\mathbf{X}_{\mathcal{A}_1})^{-1}\mathbf{X}^T_{\mathcal{A}_1}(I_n - \mathbf{P}_{\mathcal{A}_1})(I_n - \mathbf{P}_{\mathcal{A}})\mathbf{X}_{\mathcal{A}_1}\boldsymbol{\beta}_{0,\mathcal{A}_1}$. Since $\boldsymbol{\varepsilon} \sim N(0, \sigma^2 I)$, it follows that $B_1 \sim N(0, \sigma^2(\mathbf{X}^T_{\mathcal{A}_1}\mathbf{X}_{\mathcal{A}_1})^{-1}\mathbf{X}^T_{\mathcal{A}_1}(I_n - \mathbf{P}_{\mathcal{A}})\mathbf{X}_{\mathcal{A}_1}(\mathbf{X}^T_{\mathcal{A}_1}\mathbf{X}_{\mathcal{A}_1})^{-1})$. In addition, since $I_n - \mathbf{P}_{\mathcal{A}}$ is also a projection matrix, we obtain that each entry of the vector B_1 has variance bounded from above by the largest eigenvalue of $\sigma^2(\mathbf{X}^T_{\mathcal{A}_1}\mathbf{X}_{\mathcal{A}_1})^{-1}$, which can be further bounded from above by $c^{-2}\sigma^2$ since the model size $|\mathcal{A}_1| < |\mathcal{S}_0| < \kappa_c$. Thus,

$$P(\|B_1\|_\infty > C\sqrt{(\log n)/n}) \le |\mathcal{A}_1| P(|Z| > C\sqrt{(\log n)}) \le n\exp(-C\log n)/\sqrt{\log n} = o(n^{-C}),$$

where Z is a standard normal random variable. The above result ensures that $\|B_1\|_\infty \leq C\sqrt{(\log n)/n}$ with probability at least $1 - o(n^{-C})$ with $C > 0$ some generic constant. Since $\mathbf{X}_{\mathcal{A}_1}^T(I_n - \mathbf{P}_{\mathcal{A}_1}) = \mathbf{0}$, it follows that $B_2 = 0$. Combining these results yields

$$\|(\mathbf{X}_{\mathcal{A}_1}^T\mathbf{X}_{\mathcal{A}_1})^{-1}\mathbf{X}_{\mathcal{A}_1}^T\tilde{\boldsymbol{\varepsilon}}\|_\infty \leq C\sqrt{(\log n)/n} \tag{24}$$

with probability at least $1 - o(n^{-C})$.

Next we consider $\mathbf{X}_{\mathcal{S}_0^c}^T(I_n - \mathbf{P}_{\mathcal{A}_1})\tilde{\boldsymbol{\varepsilon}}$. By the definition of $\tilde{\boldsymbol{\varepsilon}}$ we have

$$n^{-1}\mathbf{X}_{\mathcal{S}_0^c}^T(I_n - \mathbf{P}_{\mathcal{A}_1})\tilde{\boldsymbol{\varepsilon}} = D_1 + D_2,$$

where $D_1 = n^{-1}\mathbf{X}_{\mathcal{S}_0^c}^T(I_n - \mathbf{P}_{\mathcal{A}_1})(I_n - \mathbf{P}_{\mathcal{A}})\boldsymbol{\varepsilon}$ and $D_2 = n^{-1}\mathbf{X}_{\mathcal{S}_0^c}^T(I_n - \mathbf{P}_{\mathcal{A}_1})(I_n - \mathbf{P}_{\mathcal{A}})\mathbf{X}_{\mathcal{A}_1}\boldsymbol{\beta}_{0,\mathcal{A}_1}$. Using similar arguments as above we can prove that with probability at least $1 - O(p^{-C})$, the vector D_1 satisfies $\|D_1\|_\infty \leq C\sqrt{(\log p)/n}$. This together with Condition 8 entails

$$\|n^{-1}\mathbf{X}_{\mathcal{S}_0^c}^T(I_n - \mathbf{P}_{\mathcal{A}_1})\tilde{\boldsymbol{\varepsilon}}\|_\infty \leq C\sqrt{(\log p)/n} \tag{25}$$

with probability at least $1 - o(p^{-C})$.

Combining the above results (24) and (25) we can prove that with probability at least $1 - o(n^{-C})$, the event $\tilde{\mathcal{E}}_2$ holds. This completes the proof.

B Technical details for Conditions 2 and 3

The probability bound in Condition 2 has been shown by Fan and Lv (2013) to hold for a wide class of error distributions under mild conditions of design matrix. So we only consider the probability bound in Condition 3.

Write $\mathbf{b}_j = \mathbf{X}_{\mathcal{S}_0}(\mathbf{X}_{\mathcal{S}_0}^T\mathbf{X}_{\mathcal{S}_0})^{-1}\mathbf{e}_j$ for $j \in \mathcal{S}_0$ and $\mathbf{d}_j = (I_n - \mathbf{P}_{\mathcal{S}_0})\mathbf{X}_{\mathcal{S}_0^c}\mathbf{e}_j$ for $j \in \mathcal{S}_0^c$, where $\mathbf{e}_j$ is the vector with jth component 1 and all other components 0. Then

$$\|(\mathbf{X}_{\mathcal{S}_0}^T\mathbf{X}_{\mathcal{S}_0})^{-1}\mathbf{X}_{\mathcal{S}_0}^T\boldsymbol{\varepsilon}\|_\infty = \max_{j\in\mathcal{S}_0}|\mathbf{b}_j^T\boldsymbol{\varepsilon}|, \qquad \|\mathbf{X}_{\mathcal{S}_0^c}^T(I_n - \mathbf{P}_{\mathcal{S}_0})\boldsymbol{\varepsilon}\|_\infty = \max_{j\in\mathcal{S}_0^c}|\mathbf{d}_j^T\boldsymbol{\varepsilon}|.$$

Applying the Bonferroni inequality gives

$$P\big(\|(\mathbf{X}_{\mathcal{S}_0}^T\mathbf{X}_{\mathcal{S}_0})^{-1}\mathbf{X}_{\mathcal{S}_0}^T\boldsymbol{\varepsilon}\|_\infty > c_2\sqrt{(\log n)/n}\big) \leq \sum_{j\in\mathcal{S}_0} P\big(|\mathbf{b}_j^T\boldsymbol{\varepsilon}| > c_2\sqrt{(\log n)/n}\big), \tag{26}$$

$$P\big(n^{-1}\|\mathbf{X}_{\mathcal{S}_0}^T(I_n - \mathbf{P}_{\mathcal{S}_0})\boldsymbol{\varepsilon}\|_\infty > \lambda_0\big) \leq \sum_{j\in\mathcal{S}_0^c} P\big(n^{-1}|\mathbf{d}_j^T\boldsymbol{\varepsilon}| > \lambda_0\big). \tag{27}$$

Thus, the problem reduces to studying the deviation bounds of random variables $\mathbf{b}_j^T\boldsymbol{\varepsilon}$ and $\mathbf{d}_j^T\boldsymbol{\varepsilon}$. We consider two classes of error distributions.

Case 1 (Bounded error): If the errors ε_i, $i = 1, \cdots, n$ all have magnitude bounded by some constant $M > 0$, then by Hoeffding's inequality (Hoeffding, 1963),

$$P\big(|\mathbf{b}_j^T\boldsymbol{\varepsilon}| > c_2\sqrt{(\log n)/n}\big) \le 2\exp\Big(-\frac{c_2^2\log n}{2M^2\|\mathbf{b}_j\|_2^2 n}\Big).$$

Note that since the true model size $|\mathcal{S}_0| < \kappa_c$, it follows that $\|\mathbf{b}_j\|_2^2 = \mathbf{e}_j^T(\mathbf{X}_{\mathcal{S}_0}^T\mathbf{X}_{\mathcal{S}_0})^{-1}\mathbf{e}_j \le c^{-2}n^{-1}$. Therefore, the above inequality can be further bounded as

$$P(|\mathbf{b}_j^T\boldsymbol{\varepsilon}| > c_2\sqrt{(\log n)/n}) \le 2\exp\big(-\frac{c_2^2c^2\log n}{2M^2}\big).$$

This together with (26) entails that for some constant $c_3 \in (0, c_2^2c^2/(2M^2))$,

$$P\big(\|(\mathbf{X}_{\mathcal{S}_0}^T\mathbf{X}_{\mathcal{S}_0})^{-1}\mathbf{X}_{\mathcal{S}_0}^T\boldsymbol{\varepsilon}\|_\infty > c_2\sqrt{(\log n)/n}\big) \le o(n^{-c_3}). \tag{28}$$

Now we consider $P(n^{-1}|\mathbf{d}_j^T\boldsymbol{\varepsilon}| > \lambda_0)$. Using similar arguments and noting that $\|\mathbf{d}_j\|_2^2 = \mathbf{e}_j^T\mathbf{X}_{\mathcal{S}_0^c}^T(I_n - \mathbf{P}_{\mathcal{S}_0})\mathbf{X}_{\mathcal{S}_0^c}\mathbf{e}_j \le \mathbf{e}_j^T\mathbf{X}_{\mathcal{S}_0^c}^T\mathbf{X}_{\mathcal{S}_0^c}\mathbf{e}_j = n$ for each $j \in \mathcal{S}_0^c$, we obtain that

$$P(n^{-1}|\mathbf{d}_j^T\boldsymbol{\varepsilon}| > \lambda_0) \le 2\exp\Big(-\frac{c_0^2\log p}{2M^2}\Big) = O(p^{-c_0^2/(2M^2)}).$$

This together with (27) ensures that

$$P\big(n^{-1}\max_{j\in\mathcal{S}_0^c}|\mathbf{d}_j^T\boldsymbol{\varepsilon}| > \lambda_0\big) \le O(p^{-c_0^2/(2M^2)+1}). \tag{29}$$

Combining (28) with (29) proves

$$P(\mathcal{E}_2^c) \le P\big(\max_{j\in\mathcal{S}_0}|\mathbf{b}_j^T\boldsymbol{\varepsilon}| > c_2\sqrt{(\log n)/n}\big) + P\big(n^{-1}\max_{j\in\mathcal{S}_0^c}|\mathbf{d}_j^T\boldsymbol{\varepsilon}| > \lambda_0\big) \le o(n^{-c_3}),$$

for c_0 large enough. This completes the proof.

Case 2 (light-tailed error): Assume there exist constants $M, \nu_0 \in (0, \infty)$ such that

$$E\Big(\exp(\varepsilon_i/M) - 1 - \frac{\varepsilon_i}{M}\Big)M^2 \le \nu_0^2/2,$$

uniformly over all $i = 1, \cdots, n$. Then by Proposition 4 in Fan and Lv (2011) we have

$$P(|\mathbf{b}_j^T\boldsymbol{\varepsilon}| > c_2\sqrt{(\log n)/n}) \le 2\exp\Big(-\frac{1}{2}\frac{c_2^2\log n}{\nu_0/c^2 + \|\mathbf{b}_j\|_\infty M\sqrt{n\log n}}\Big),$$

for each $j \in \mathcal{S}_0$. The above probability bound becomes $O(n^{-\frac{c_2^2}{2(\nu_0/c^2+d)}})$ for some constant $d > 0$ if we further assume that the maximum absolute element of matrix $\mathbf{X}_{\mathcal{S}_0}(\mathbf{X}_{\mathcal{S}_0}^T\mathbf{X}_{\mathcal{S}_0})^{-1}$

is bounded by $d/(M\sqrt{n\log n})$. Note that in the orthogonal design case, $\mathbf{X}_{\mathcal{S}_0}(\mathbf{X}_{\mathcal{S}_0}^T\mathbf{X}_{\mathcal{S}_0})^{-1} = n^{-1}\mathbf{X}_{\mathcal{S}_0}$ and this additional assumption is very mild.

Similarly, we can show that

$$P(n^{-1}|\mathbf{d}_j^T\boldsymbol{\varepsilon}| > \lambda_0) \leq 2\exp\Big(-\frac{1}{2}\frac{c_0^2\log p}{\nu_0^2 + \|\mathbf{d}_j\|_\infty M\lambda_0}\Big).$$

Therefore, if the maximum absolute element of matrix $(I_n - \mathbf{P}_{\mathcal{S}_0})\mathbf{X}_{\mathcal{S}_0^c}$ is bounded by $\tilde{d}M^{-1}c_0^{-1}\sqrt{n/(\log p)}$ with $\tilde{d} > 0$ some constant, then the above inequality has upper bound of the order $O(p^{-c_0^2/(2(\nu_0^2+\tilde{d}))})$.

Therefore, using similar arguments to those in case 1 we can show that for $c_0 > 0$ large enough, $P(\mathcal{E}_2^c) \leq o(n^{-c_7})$ with $c_7 > 0$ some constant. This completes the proof.

References

Barron, A. R., Cohen, A., Wolfgang, D. and DeVore, R. A. (2008). Approximation and learning by greedy algorithms. *Ann. Statist.* **36**, 64–94

Bickel, P., Ritov, Y. and Tsybakov, A. (2009). Simultaneous analysis of lasso and dantzig selector. *Ann. Statist.* **37**, 1705–32

Bühlmann, P. (2012). Statistical significance in high-dimensional linear models. *Bernoulli* **19**, 1212-42.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications.* Heidelberg; New York: Springer.

Donoho, D. L. and Stodden, V. (2006). Breakdown point of model selection when the number of variables exceeds the number of observations. *Proceedings of the International Joint Conference on Neural Networks* 2006, 1916–21.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression. *Ann. Statist.* **32**, 407–99

Fan, J. and Lv, J. (2011). Nonconcave penalized likelihood with NP-dimensionality. *IEEE Transactions on Information Theory.* **57**, 5467–84.

Fan, Y. and Lv, J. (2013). Asymptotic equivalence of regularization methods in thresholded parameter space. *J. Am. Statist. Assoc.* **108**, 1044–61.

Hall, P., Xia, Y. and Xue, J.-H. (2013). Simple tiered classifiers. *Biometrika* **100**, 431–45.

Hoeffding, W. (1963). Probability Inequalities for Sums of Bounded Random Variables. *J. Am. Statist. Assoc.*, **58**, 13–30.

Ing, C-K. and Lai, T. L. (2011). A stepwise regression method and consistent model selection for high-dimensional sparse linear models. *Statist. Sinica.* **21**, 1473–513.

Li, Ping (2009). ABC-Boost: Adaptive base class boost for multi-class classification. *Proceedings of the 26th Annual International Conference on Machine Learning* 2009, 529–36.

Lockhart, R., Taylor, J., Tibshirani, R. J., and Tibshirani, R. (2013). A significance test for the lasso (with discussion). *Ann. Statist.*, to appear.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *J. R. Statist. Soc.* B **72**(4), 417–73.

Minnier, J., Tian, L. and Cai, T. (2011). A perturbation method for inference on regularized regression estimates. *J. Am. Statist. Assoc.* **106**, 1371–82.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.* B **58**, 267–88.

Tropp, J. A. (2004). Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory* **50**, 2231–42.

van de Geer, S., Bühlmann, P., and Zhou, S.(2011). The adaptive and the thresholded lasso for potentially misspecified models (and a lower bound for the lasso). *Electron. J. Stat.* **5**, 688749.

Wang, H. (2009). Forward regression for ultra-high dimensional variable screening. *J. Am. Statist. Assoc.* **104**, 1512–24.

Wasserman, L. and Roeder, K. (2009). High dimensional variable selection. *Ann. Statist.* **37**, 2178–201.

Yuan, M. and Zou, H.(2009). Efficient global approximation of generalized nonlinear l1-regularized solution paths and its applications. *J. Am. Statist. Assoc.* **104**, 1562–74.

Zhang, C-H. and Zhang, S. (2013). Confidence intervals for low dimensional parameters in high dimensional linear models. *J. R. Statist. Soc.* B **76**, 217–42.

Zhang, T. (2008). Adaptive forward-backward greedy algorithm for sparse learning with linear models. *NIPS* 2008.

Zhong, W., Zhang, T., Zhu, Y., and Liu, J. (2013). Correlation pursuit: forward stepwise variable selection for index models. *J. R. Statist. Soc.* B **74**, 849–70.

Zou, H. (2006). The adaptive lasso and its oracle properties. *J. Am. Statist. Assoc.* **101**, 1418–29.

Zou, H. and Zhang, H. (2009). On the adaptive elastic-net with a diverging number of parameters. *Ann. Statist.* **37**, 1733–51.