Solar: fast and accurate variable selection in high dimensional and large scale data*

Ning Xu*

School of Economics, University of Sydney, Australia

Timothy C.G. Fisher

School of Economics, University of Sydney, Australia

Jian Hong

School of Economics, University of Sydney, Australia

Abstract

We propose a new algorithm for variable selection in high dimensional and large scale data, *subsample-ordered least-angle regression (solar)*, and a coordinate descent generalization, *solar-cd*. We show that the solar variable selection consistency and the variable ranking accuracy on the avergae L_0 path under the general framework of forward selection. Using simulations, examples, and real-world data, we demonstrate the following advantages of solar: (i) solar yields, without any increase in computation load, substantial improvements over lasso in terms of the sparsity (37-64% reduction in redundant variable selection), stability, and accuracy of variable selection; (ii) compared with the lasso safe/strong rule and variable screening, solar largely avoids selection of redundant variables and rejection of informative variables in the presence of complicated dependence structures and harsh settings of the irrepresentable condition; (iii) the sparsity of solar conserves residual degrees of freedom for data-splitting hypothesis testing, improving the efficiency and accuracy of post-selection inference; and (iv) replacing lasso with solar in subsampling selection (e.g., the bootstrap lasso or stability selection) produces a multi-layer variable ranking scheme that improves selection sparsity, ranking accuracy, and computation load (a saving of at least 96% in runtime, exceeding the theoretical maximum speedup for parallelizing lasso-type algorithms). Because solar is based on averaging and re-ordering lasso paths via the L_0 norm, it is easy to adapt to lasso variants. In a supplementary file and at the dedicated Github page, we provide a parallel computing package for solar (solarpy) that uses a Python interface and an Intel MKL Fortran/C++ compiler.

Keywords: Variable selection, sparsity, computation time, complicated dependence structure, lasso rules, irrepresentable condition, bolasso, subsampling selection, variable screening.

*The authors would like to thank Pierre Del Moral and Peter Exterkate for careful comments as well as seminar participants at Google Research, NICTA, Monash University, University of Melbourne, and UNSW.

*Principal corresponding author.

Email addresses: n.xu@sydney.edu.au (Ning Xu), tim.fisher@sydney.edu.au (Timothy C.G. Fisher), jian.hong@sydney.edu.au (Jian Hong)

1. Introduction

Recent innovations to lasso-type algorithms (Efron et al., 2004; Friedman et al., 2007, 2010) have largely addressed selection of redundant variables, rejection of informative variables, and poor performance under high multicollinearity in high dimensional ($p > n$) and large scale data (large p and large n). However, in alleviating old problems, the innovations have revealed new challenges.

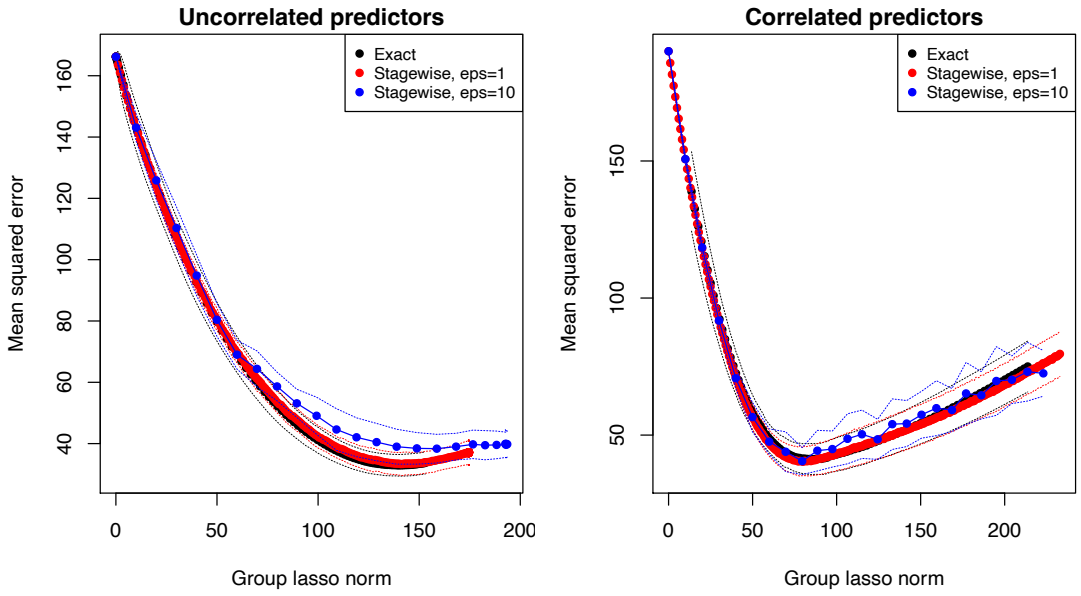
Bootstrap variable selection, for example, markedly improves variable selection sparsity (Bach, 2008; Meinshausen and Bühlmann, 2010), yet it relies on repeating cross-validated lasso on tens to hundreds of bootstrap subsamples for variable selection. Xu et al. (2012) and Sections 4.4 and 4.6 below illustrate that bootstrap selection methods exponentially increase computation load, limiting applicability in large scale data such as DNA sequencing, image recognition, and natural language processing (where both p and n are often over $10,000$).

One strategy to improve lasso selection sparsity without increasing computation burden is to use a post-selection rule to screen variables selected by lasso. Post-lasso selection rules [e.g., the 'safe rule' (Ghaoui et al., 2010) and the 'strong rule' (Tibshirani et al., 2012)] are capable of reducing the number of variables to enhance computational efficiency in lasso. However, recent research (Wang et al., 2014; Zeng et al., 2017) and Section 3.2 suggest both rules may be prone to rejecting informative variables, selecting redundant variables, or proposing repeated modifications (e.g., rejecting a variable in an early round and adding it back in a later round).

Data-splitting hypothesis tests are another way to screen variables selected by lasso (Wasserman and Roeder, 2009; Meinshausen et al., 2009; Barber and Candès, 2019; Romano and DiCiccio, 2019; DiCiccio et al., 2020). The original data are divided into two: one part for variable selection, the other part for testing. However, to improve test power, data splitting is repeated on each bootstrap subsample, raising similar computational concerns as bootstrapping variable selection (Bach, 2008; Meinshausen and Bühlmann, 2010). DiCiccio et al. (2020) also argue that because data splitting reserves some of the data for variable selection, it reduces the degrees of freedom for testing on the remaining data, presenting a challenge when sample size is limited.

Specifically designed to address the challenges of high dimensional data, the variable screening algorithm (Fan and Lv, 2008; Hall and Miller, 2009; Hall et al., 2009; Li et al., 2012a,b) ranks the absolute values of unconditional correlations between each covariate and the response variable, selecting only the top-ranked variables. However, Fan and Lv (2008), Barut et al. (2016), and Section 3.2 below show that variable screening also suffers from selection of redundant variables and rejection of informative variables when the dependence structures are complicated.

Research has studied forward selection (stagewise or stepwise) thoroughly and established various statistical properties under different assumptions (Efroymson, 1966; Draper and Smith, 1966; Friedman et al., 2001). According to Friedman et al. (2001); Weisberg (2004), forward selection was

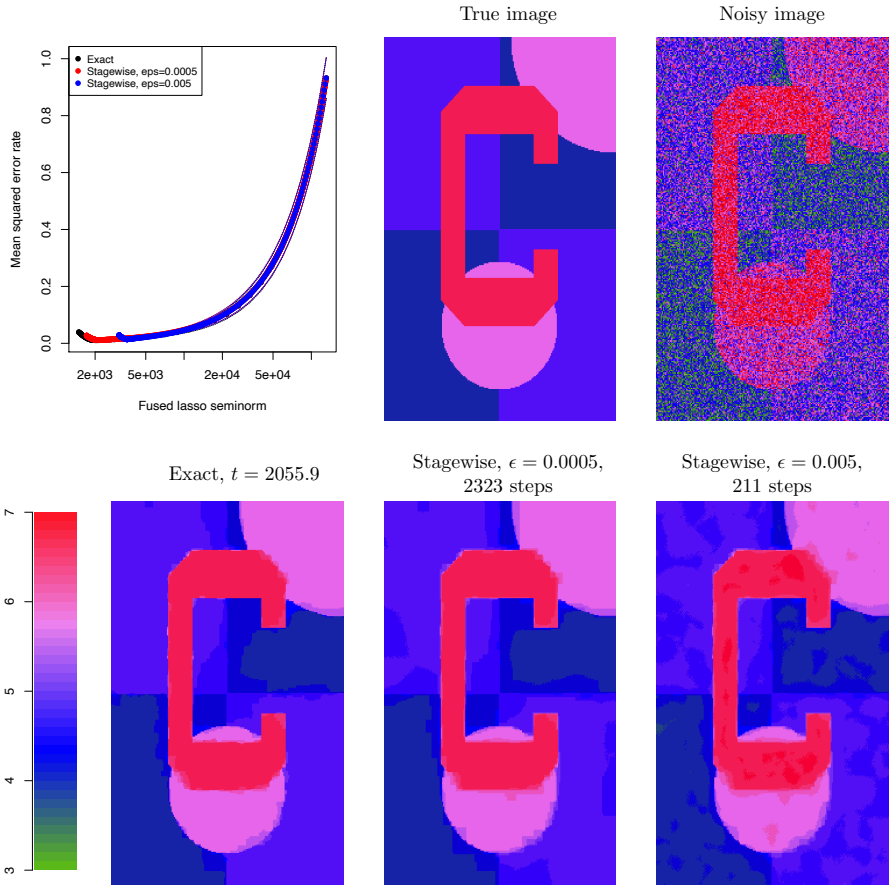


Algorithm timings		
Method	Uncorrelated case	Correlated case
Exact: coordinate descent, 100 solutions	9.08 (1.06)	78.64 (17.92)
Stagewise: $\epsilon = 1$, 250 estimates	0.93 (0.00)	0.94 (0.01)
Stagewise: $\epsilon = 10$, 25 estimates	0.09 (0.00)	0.10 (0.01)
Frank-Wolfe: within 1% of criterion value	67.73 (10.37)	92.91 (8.37)
Frank-Wolfe: within 1% of mean squared error	1.30 (0.56)	13.17 (26.26)

Figure 1: Statistical and computational comparisons between group lasso (coordinate descent) and the forward selection with $n = 200, p = 4000$.

historically dismissed due to inefficiency and sensitivity to sampling randomness, multicollinearity, noise and outliers in high-dimensional spaces due to the iterative refitting of the residual. However, Tibshirani (2015) illustrates that, from a modern perspective, forward selection can present considerable benefits in terms of the generalization error of the fitted models (not only in regression, but across variety of settings). Furthermore, forward selection is computationally cheap by modern standards: to trace out a path of regularized estimates, we repeat very simple iterations (each requiring at most p inner products) that could be trivially parallelized. Tibshirani (2015) illustrates his point in a number of large-scale applications. We summarize two of them in Figure 1 and 2.

- With $n = 200, p = 4000$, the top row of Figure 1 shows that forward selection can achieve competitive mean squared errors to that of group lasso under two different setups for the predictors in group lasso regression: uncorrelated and block correlated. (The curves were averaged over 10 simulations, with standard deviations denoted by dotted lines.) The lower table reports runtimes in seconds (averaged over 10 simulations, with standard deviations in parentheses) for the various algorithms considered, and shows that the stagewise algorithm represents a computationally



Algorithm timings	
Method	Runtime
Exact: maximum flow, 100 solutions	109.04 (6.21)
Stagewise: $\epsilon = 0.0025$, 6000 estimates	15.11 (0.18)
Stagewise: $\epsilon = 0.25$, 500 estimates	1.26 (0.02)

Figure 2: Statistical and computational comparisons between fused lasso and forward selection on image denoising.

attractive alternative to the coordinate descent approach and the Frank-Wolfe algorithm.

- Figure 2 compares between fused lasso and forward selection on image denoising. The true underlying 300×200 image is displayed in the middle of the top row. Over 10 noisy pertubations of this underlying image, with one such example shown in the right plot of the top row, we compare averaged mean squared errors of the exact solutions and stagewise estimates, in the left plot of the top row. Average timings for these methods are given in the bottom table. (Standard deviations are denoted by dotted lines in the error plots, and are in parentheses in the table.) The stagewise estimates have competitive mean squared errors and are fast to compute.

1.1. Main results

To address the issues above while boosting the forward selection advantages, we propose a new for-

ward selection algorithm, *subsample-ordered least-angle regression (solar)*, and its coordinate-descent generalization, *solar-cd*. Because solar is based on averaging and re-ordering lasso paths via the L_0 norm, it is easily adaptable to many lasso variants. We show that the solar variable selection consistency and the variable ranking accuracy on the avergae L_0 path under the general framework of forward selection.

Using simulations, examples, and real-world data, we demonstrate the following advantages of solar: (i) solar yields, without any increase in computation load, substantial improvements over lasso in terms of the sparsity (37-64% reduction in redundant variable selection), stability, and accuracy of variable selection; (ii) compared with the lasso safe/strong rule and variable screening, solar largely avoids selection of redundant variables and rejection of informative variables in the presence of complicated dependence structures and harsh settings of the irrepresentable condition; (iii) the sparsity of solar conserves residual degrees of freedom for data-splitting hypothesis testing, improving the efficiency and accuracy of post-selection inference; (iv) replacing lasso with solar in subsampling selection (e.g., the bootstrap lasso or stability selection) produces a multi-layer variable ranking scheme that improves selection sparsity, ranking accuracy, and computation load (a saving of at least 96% in runtime, exceeding the theoretical maximum speedup for parallelizing lasso-type algorithms). We provide a parallel computing package for solar (`solarpy`) that uses a Python interface and an Intel MKL Fortran/C++ compiler in a supplementary file and dedicated Github page.

The paper is organized as follows. In Section 2, we introduce the solar algorithm, explain the coordinate descent generalization of solar, and discuss generalizations of solar to variants of lasso. In Section 3, we use examples to demonstrate the advantages of solar over lasso, the safe/strong rules, variable screening, forward regression, and lasso-related bootstrap selection. In Section 4, we use simulations to demonstrate the advantages of solar over lasso-type algorithms in terms of variable selection sparsity, accuracy, and computation load. In Section 5, we use real-world data to show that the improvements from solar are feasible in the presence complicated dependence structures, while lasso and elastic net [the lasso variant alleged (Zou and Hastie, 2005; Jia and Yu, 2010) to have the best selection accuracy and sparsity under multicollinearity] completely lose sparsity. The solar variable selection consistency and the variable ranking accuracy are included in Appendix A. The code and raw result of solarpy is in Supplementary Material.

2. The Solar algorithm

Intuitively, the advantages of solar over lasso and lasso-related bootstrap selection stem from a reallocation of computation. Lasso and its variants allocate most of the computation to optimizing the shrinkage parameter (λ) using cross-validation. By contrast, without any increase in computation load, solar allocates computation to stabilizing the solution path.

Zhang (2009, Theorem 2) reveals that the earlier a variable enters the solution path, the more likely it is to be informative. This suggests an accurate and stable ordering of variables in the solution path may help to identify the informative variables. Since we focus on accuracy, the only relevant feature of the regression coefficients in the solution path is whether $\beta_i = 0$ at each stage. Thus, we re-parameterize the lasso path using the L_0 norm.

Definition 2.1 (L_0 solution path)**.** Define the L_0 **solution path** on (Y, X) to be the order that least angle regression includes variables across all stages. For example, if the least angle regression includes $\mathbf{x}_3$ at stage 1, $\mathbf{x}_2$ at stage 2 and $\mathbf{x}_1$ at stage 3, the corresponding L_0 path is the ordered set $\{\{\mathbf{x}_3\}, \{\mathbf{x}_3, \mathbf{x}_2\}, \{\mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1\}\}$.

2.1. Solar optimized by least angle regression

A solution path is the foundation of L_p-penalized regression. The first step of solar is to reduce the sensitivity of the solution path when $p > n$. The *average* L_0 *solution path estimation* algorithm (summarized in Algorithm 1 and illustrated in Figure 3) accomplishes this step by estimating the *average stage each* $\mathbf{x}_i$ *enters the solution path* of a least angle regression.

Algorithm 1: average L_0 solution path estimation

input : (Y, X).

1 generate K subsamples $\left\{\left(Y^k, X^k\right)\right\}_{k=1}^{K}$ by randomly remove $1/K$ of observations in (Y, X);

2 set $\widetilde{p} = \min\left\{n\left(K-1\right)/K, p\right\}$;

3 **for** *k := 1 to K, stepsize = 1* **do**

4 run an unrestricted least angle regression (or any forward selection algorithm) on $\left(Y^k, X^k\right)$ and record the order of variable inclusion at each stage;

5 define $\widehat{q}^k = \mathbf{0} \in \mathbb{R}^p$;

6 $\forall i, l \in \mathbf{N}^+$, if $\mathbf{x}_i$ is included at stage l and excluded at $l-1$, set $\widehat{q}_i^k = (\widetilde{p} + 1 - l)/\widetilde{p}$, where $\widehat{q}_i^k$ is the i^{th} entry of $\widehat{q}^k$;

7 **end**

8 $\widehat{q} := \frac{1}{K}\sum_{k=1}^{K} \widehat{q}^k$;

9 **return** $\widehat{q}$

After the subsamples are created, lines 4-6 of Algorithm 1 compute $\widehat{q}^k$, which summarizes the order that least angle regression includes each $\mathbf{x}_i$ across all stages (see Figure 3). The unrestricted least angle regression ranks variables by the stage they enter the solution path. As shown in line 6 of Algorithm 1 and Figure 3, variables included at earlier stages have larger $\widehat{q}_i^k$ values: the first variable included is assigned 1, the last is assigned $1/\widetilde{p}$, while the rejected variables are assigned 0 (which occurs only when $p > n$). Thus, theL_0 solution path is obtained by ranking the $\mathbf{x}_i$ according to their $\widehat{q}_i^k$ values.

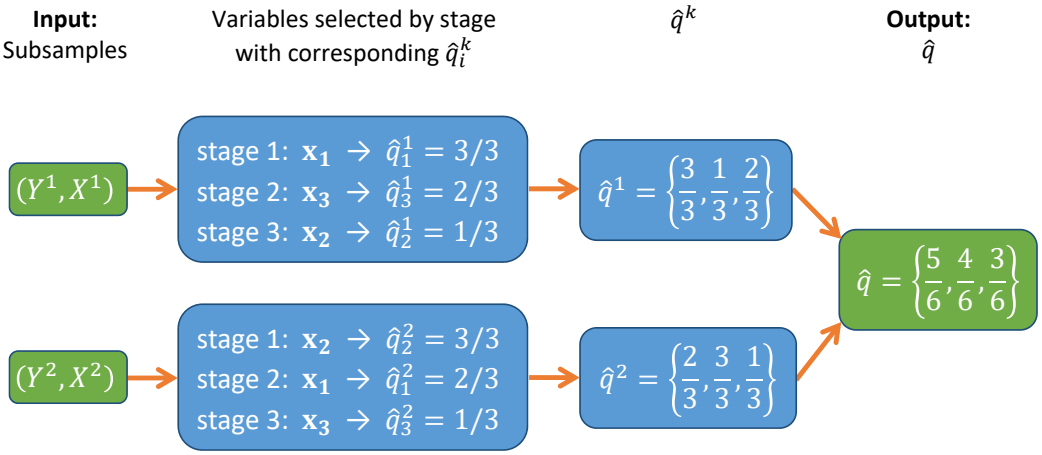


Figure 3: Computation of $\widehat{q}$ on 2 subsamples by least angle regression.

Zhang (2009, Theorem 2) implies that, on average, variables with the largest $\widehat{q}_i^k$ values are more likely to be informative. The $\widehat{q}_i^k$ may be sensitive in high-dimensional spaces to multicollinearity, sampling randomness, and noise. In these circumstances, a redundant variable may be included at an early stage in some $\left(Y^k, X^k\right)$ subsample. Algorithm 1 reduces the impact of sensitivity in the $\widehat{q}_i^k$ by computing $\widehat{q} := \frac{1}{K}\sum_{k=1}^{K} \widehat{q}^k$ and ranking the $\mathbf{x}_i$ according to $\widehat{q}_i$ (the i^{th} entry in $\widehat{q}$), to arrive at the average L_0 solution path. The average L_0 solution path is formally defined as follows.

Definition 2.2 (average L_0 solution path)**.** Define the **average L_0 solution path** of least angle regression on $\left\{\left(Y^k, X^k\right)\right\}_{k=1}^{K}$ to be the (decreasing) rank order of the $\mathbf{x}_i$ variables based on their corresponding $\widehat{q}_i$ values. For example, in Figure 3, the $\widehat{q}_i$ for $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ are, respectively, $\widehat{q}_1 = 5/6$, $\widehat{q}_2 = 4/6$ and $\widehat{q}_3 = 3/6$. Thus, the average L_0 solution path may be represented as an ordered set $\{\{\mathbf{x}_1\}, \{\mathbf{x}_1, \mathbf{x}_2\}, \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}\}$.

The solar algorithm is summarized in Algorithm 2. It is presented under the generic framework of forward regression and can easily be adapted to least angle regression, forward or backward selection algorithms. Under the general framework of forward selection by Zhang (2009), we show in Appendix A that the solar variable selection consistency and the variable ranking accuracy on the avergae L_0 path.

In Algorithm 2, variables are included into forward regression according to their rank order in the average L_0 solution path, represented by $\{Q(c)|c = 1, 0.98, \ldots, 0\}$ in Algorithm 2. We use $\widehat{q}$ from Algorithm 1 to generate a list of variables $Q(c) = \{\mathbf{x}_j \mid \widehat{q}_j \geqslant c, \forall j \leqslant p\}$. For any $c_1 > c_2$, $Q(c_1) \subset Q(c_2)$, implying a sequence of nested sets $\{Q(c)|c = 1, 0.98, \ldots, 0\}$. Each c denotes a stage of forward regression. For a given value of c, $Q(c)$ denotes the set of variables with $\|\beta_i\|_0 = 1$ on average and $Q(c) - Q(c - 0.02)$ is the set of variables with $\|\beta_i\|_0$ just turning to 1 at c. Therefore, $\{Q(c)|c = 1, 0.98, \ldots, 0\}$ is the average L_0 solution path of Definition 2.2. Variables that are more

Algorithm 2: Subsample-ordered least-angle regression (solar)

1 Randomly select 20% of the sample points as the validation set; denote the remaining points as the training set;
2 Estimate $\widehat{q}$ using Algorithm 1 on the training set and compute $Q(c) = \{\mathbf{x}_j \mid \widehat{q}_j \geqslant c, \forall j\}$ for all $c \in \{1, 0.98, \ldots, 0.02, 0\}$.
3 Run an OLS regression of each $Q(c)$ on Y using the training set and find c^*, the value of c that minimizes the validation error;
4 Compute the OLS coefficients of $Q(c^*)$ on Y using the whole sample.

likely to be informative have larger c values in $Q(c)$ and will be selected first by the solar algorithm.

2.2. Solar optimized by coordinate descent

The solar algorithm can easily be generalized to use coordinate descent. For lasso, least angle regression or coordinate descent generates a solution path parameterized by the β_i and the shrinkage parameter λ. Thus, to reprogram solar to use coordinate descent, we simply replace Algorithm 1 with Algorithm 3, which records the order of variable selection along the coordinate descent solution path.

Algorithm 3: average L_0 solution path estimation via coordinate descent

input : (Y, X).
1 generate K subsamples $\left\{\left(Y^k, X^k\right)\right\}_{k=1}^{K}$ by randomly remove $1/K$ of observations in (Y, X);
2 set $\widetilde{p} = \min\{n_{\text{sub}}, p\}$;
3 **for** *k := 1 to K, stepsize = 1* **do**
4 denote λ_s as the shrinkage parameter value that coordinate descent lasso selects s variables, $\forall s \in [0, \widetilde{p}]$;
5 run a pathwise coordinate descent for lasso on $\left(Y^k, X^k\right)$, $\forall \lambda \in \{\lambda_0, \lambda_1, \ldots, \lambda_{\widetilde{p}}, \}$
6 record the order of variable inclusion at each $\lambda \in \{\lambda_0, \lambda_1, \ldots, \lambda_{\widetilde{p}}, \}$;
7 define $\widehat{q}^k = \mathbf{0} \in \mathbb{R}^p$;
8 $\forall i, s \in \mathbf{N}^+$, if $\mathbf{x}_i$ is included at $\lambda = \lambda_s$ and excluded at λ_{s-1}, set $\widehat{q}_i^k = (\widetilde{p} + 1 - s)/\widetilde{p}$, where $\widehat{q}_i^k$ is the i^{th} entry of $\widehat{q}^k$;
9 **end**
10 $\widehat{q} := \frac{1}{K}\sum_{k=1}^{K} \widehat{q}^k$;
11 **return** $\widehat{q}$

Algorithm 3 serves the same purpose as Algorithm 1: to estimate the average L_0 path. Algorithm 3 uses λ to record the order that each variable enters the path. Consider the example in Figure 4. To re-parameterize the solution path, we denote λ_s to be the λ value that coordinate descent lasso includes s variables, $\forall s \in (0, \min\{n/2, p\}]$, giving a sequence of λ for grid search. In each subsample $\left(Y^k, X^k\right)$,

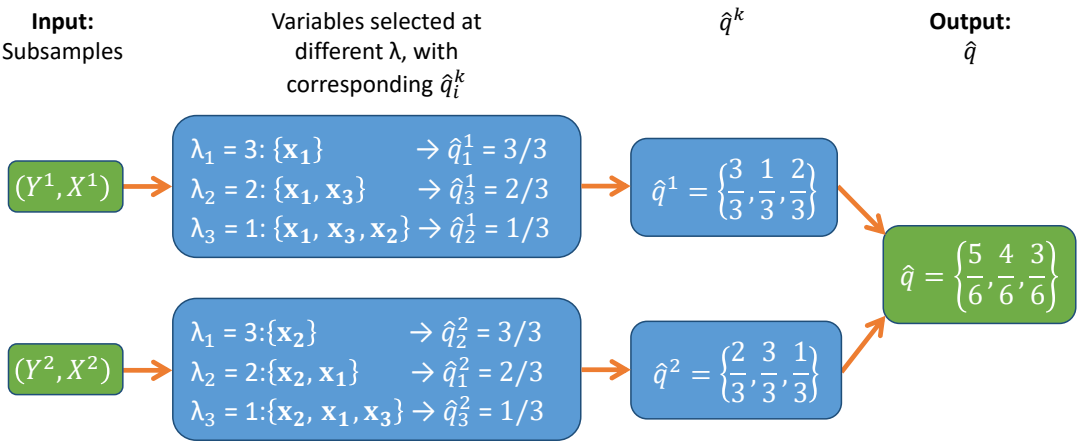


Figure 4: Computation of $\widehat{q}$ on 2 subsamples using coordinate descent.

we train a standard pathwise coordinate descent for lasso, allowing λ to increase stepwise within the grid $\{\lambda_1, \ldots, \lambda_{\min\{n/2,p\}}\}$, where $\lambda_1 \geqslant \ldots \geqslant \lambda_{\min\{n/2,p\}}$. In Figure 4, when $\lambda \leqslant \lambda_3$ at subsample (Y^1, X^1), all three variables are selected in the solution path, implying that $\widehat{q}_i^1 \geqslant 1/3$ for all variables. When λ increases to λ_2, only $\{\mathbf{x}_3, \mathbf{x}_1\}$ survive the harsher shrinkage, implying that they should be ranked higher than $\mathbf{x}_2$. As a result, $\widehat{q}_1^1, \widehat{q}_3^1 \geqslant 2/3$ and $\widehat{q}_2^1 = 1/3$. When λ reaches λ_3, only $\{\mathbf{x}_1\}$ remains, leaving $\widehat{q}_1^1 = 3/3$ and $\widehat{q}_3^1 = 2/3$. Applying the same method to each subsample produces the same $\widehat{q}$ as Algorithm 1.

2.3. Generalization to lasso variants

Because it is trained by least angle regression or coordinate descent, solar can easily be extended to several lasso variants:

- 'Grouped solar' is invoked by forcing specific variables to be simultaneously selected into the solution path;
- 'Adaptive solar' is obtained by weighting variable rankings in the average L_0 path according to their OLS coefficients;
- 'Solar elastic net' or 'fused solar' is derived by replacing the coordinate descent loss function in Algorithm 3 with the L_1-L_2 loss

$$\|Y - X\beta\|_2^2 + \lambda^{(1)} \|\beta\|_1 + \lambda^{(2)} \|\beta\|_2^2 \tag{2.1}$$

or fused loss

$$\|Y - X\beta\|_2^2 + \lambda^{(1)} \|\beta\|_1 + \lambda^{(2)} \sum_{j=2}^{p} |\beta_j - \beta_{j-1}|_1 . \tag{2.2}$$

Furthermore, many lasso enhancements (e.g., safe/strong rules, post-lasso hypothesis testing) may be applied to solar because they use the same optimization methods. Rather than competing with the lasso enhancements, solar supplements them by improving variable selection performance and computation speed in large-scale applications.

3. Solar advantages over lasso variants, lasso rules, and variable screening

In this section, we use a series of examples to demonstrate the advantages of the solar algorithm for post-selection hypothesis testing, in the presence of complicated dependence structures, and in terms of its robustness to the *irrepresentable condition* (IRC).

3.1. Post-selection hypothesis testing

A major advantage of solar is its amenability to post-selection testing. Because the lasso tests (Lockhart et al., 2014; Taylor et al., 2014) are based on forward regression, they may be adapted to solar. More interestingly, it is straightforward to adapt the data-splitting tests (Wasserman and Roeder, 2009; Meinshausen et al., 2009) to solar. We illustrate this point using Example 1.

Example 1. Consider the DGP

$$Y = \mathbf{x}_0 + 2\mathbf{x}_1 + 3\mathbf{x}_2 + 4\mathbf{x}_3 + 5\mathbf{x}_4 + \sum_{j=5}^{p} 0 \cdot \mathbf{x}_j + e, \tag{3.1}$$

where $\mathbf{x}_i$, $i = 0, \ldots, p$, are standard Gaussian variables with pairwise correlations of 0.5, e is a standard Gaussian noise term, and $p/n = 100/100$.

Following Romano and DiCiccio (2019, Example 4.1) and DiCiccio et al. (2020), we conduct data-splitting tests by randomly separating the data into two portions of 50 observations. In the first round, one portion is used for solar or lasso selection and the other for testing. In the second round, the roles of the two portions are reversed. As a result, the p-values of any given variable are uncorrelated across the two rounds. Thus, we may apply Theorem 3.2 of Romano and DiCiccio (2019) and compute the average p-value across the two rounds to conduct a valid t-test for any selected covariate.

DiCiccio et al. (2020) stresses the importance of retaining residual degrees of freedom to ensure accurate tests. Because solar yields sparse and accurate variable selection (Section 4), it conserves residual degrees of freedom, improving the reliability of post-selection p-values. Figure 5 plots the average p-values for the informative variables $\{\mathbf{x}_0, \ldots, \mathbf{x}_4\}$ from post-solar and post-lasso data-splitting tests using 100 repetitions. While the solar and lasso p-values are less than 0.05 for the stronger signals $\{\mathbf{x}_1, \ldots, \mathbf{x}_4\}$, more than 25% of the lasso p-values exceed 0.05 for the weakest signal $\mathbf{x}_0$, implying non-trivial false non-rejection of H_0. By contrast, the solar p-value boxplot is very compact for $\mathbf{x}_0$, with only 5 out of 100 above 0.05. Hence, solar p-values are more reliable for detecting weak signals with small n and large p.

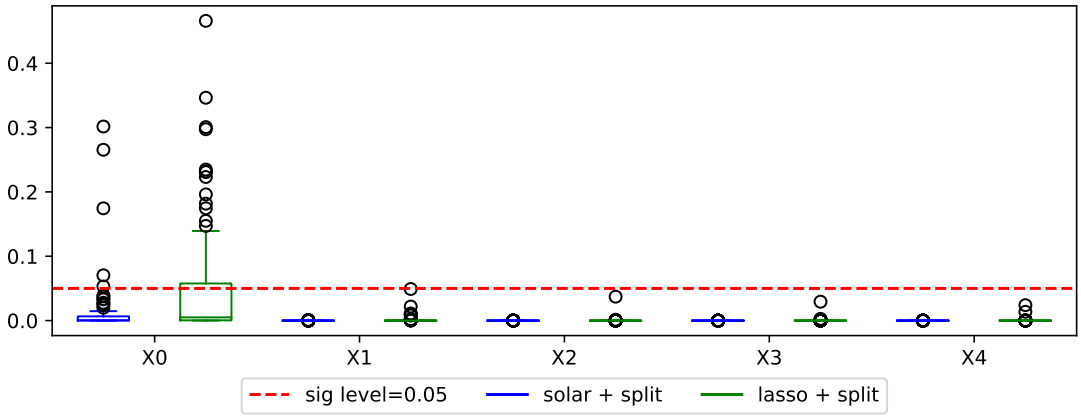


Figure 5: Average p-value boxplots for data-splitting t-tests with solar and lasso.

Moreover, the solar L_0 path may also assist with the formulation of H_0 for $p > n$. Because conserving residual degrees of freedom is so important, tests on the selection (omission) of redundant (informative) variables trigger decisions on which β_i to test. Zhang (2009, Theorem 2) shows that the earlier a variable enters the L_0 path, the more likely it is informative, implying that variables should be tested in rank order. Given the solar ranking is more robust than lasso to settings of the irrepresentable condition, sampling noise, multicollinearity, and other issues, it is likely to provide more reliable guidance on the order to test the β_i. ■

3.2. Complicated dependence structures

Another advantage of solar is that the average L_0 solution path is more robust to outliers, multicollinearity, and noise in high-dimensional spaces. Thus, solar is likely to be more reliable than other variable selection methods under complicated dependence structures. We illustrate the point with the following two (Bayesian network) examples.

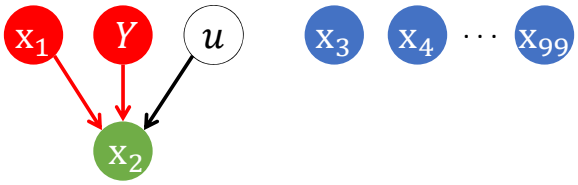


Figure 6: Y is unconditionally uncorrelated with an informative $\mathbf{x}_1$.

The first example is a common empirical regression problem: *informative variables* that are *unconditionally uncorrelated to* Y in the DGP. In Figure 6, $\mathbf{x}_1$ and $\mathbf{x}_2$ are informative for Y, while $\mathbf{x}_1$ and Y are independent. For example, in biostatistics, concussion ($\mathbf{x}_1$) or a brain tumor (Y) may cause headaches ($\mathbf{x}_2$), implying that concussion history is when attempting to diagnose a brain tumor. In this setting, Example 2a shows that solar is more reliable than post-lasso rules and variable screening.

Example 2a. In Figure 6, there are 100 variables and $\mathbf{x}_2$ is (causally) generated by its parents $\{\mathbf{x}_1, Y\}$ as follows,

$$\mathbf{x}_2 = \alpha_1 \mathbf{x}_1 + \alpha_2 Y + u, \tag{3.2}$$

where $\mathbf{x}_1$ is unconditionally uncorrelated with Y, $\mathbf{x}_1$ and Y are both unconditionally and conditionally uncorrelated with the redundant variables $\{\mathbf{x}_3, \ldots, \mathbf{x}_{99}\}$, $\{\alpha_1, \alpha_2\}$ are population regression coefficients, and u is a Gaussian noise term. If Y is chosen to be the response variable, the population regression equation is

$$Y = -\frac{\alpha_1}{\alpha_2}\mathbf{x}_1 + \frac{1}{\alpha_2}\mathbf{x}_2 - \frac{1}{\alpha_2}u. \tag{3.3}$$

Note that $\mathbf{x}_1$ and $\mathbf{x}_2$ are both informative variables for Y. However, since $\mathbf{x}_1$ is unconditionally uncorrelated with Y in the population, some post-lasso rules [such as the strong rule (Tibshirani et al., 2012) and the safe rule (Ghaoui et al., 2010)] may be prone to rejecting $\mathbf{x}_1$. For a given value of the shrinkage parameter λ in grid search, the base strong rule and the safe rule for lasso to reject a selected variable, respectively, satisfies (3.4) and (3.5):

$$\left|\mathbf{x}_i^T Y\right| < \lambda - \|\mathbf{x}_i\|_2 \|Y\|_2 \tfrac{\lambda_{max}-\lambda}{\lambda_{max}}; \tag{3.4}$$

$$\left|\mathbf{x}_i^T Y\right| < 2\lambda - \lambda_{max}, \tag{3.5}$$

where the $\mathbf{x}_i$ are standardized and λ_{max} is the value of the shrinkage parameter that rejects all the variables. Both rules are based on the unconditional covariance between $\mathbf{x}_i$ and Y. For a given value of λ (typically selected by CV), lasso will likely select $\mathbf{x}_1$ and $\mathbf{x}_2$ along with redundant variables from $\{\mathbf{x}_3, \ldots, \mathbf{x}_{99}\}$ [because the DGP does not violate the IRC]. Since $\text{corr}(\mathbf{x}_1, Y) = \text{corr}(\mathbf{x}_3, Y) = \cdots = \text{corr}(\mathbf{x}_{99}, Y) = 0$ in the population, the sample value of $\left|\mathbf{x}_1^T Y\right|$ will be approximately as small as the $\left|\mathbf{x}_i^T Y\right|$ of any redundant variable. Put another way, $\mathbf{x}_1$ cannot be distinguished from the redundant variables by the value of $\left|\mathbf{x}_i^T Y\right|$. To ensure $\mathbf{x}_1$ is not rejected by (3.4) or (3.5), both $\lambda - \|\mathbf{x}_1\|_2 \|Y\|_2 \frac{\lambda_{max}-\lambda}{\lambda_{max}}$ and $2\lambda - \lambda_{max}$ must be smaller than $\left|\mathbf{x}_1^T Y\right|$. However, this will lead to two problems. First, decreasing the right-hand side of (3.4) and (3.5) will reduce the value of λ, implying that lasso will select more redundant variables. Second, since $\left|\mathbf{x}_1^T Y\right|$ will be approximately as small as the $\left|\mathbf{x}_i^T Y\right|$ of any redundant variable selected by lasso, not rejecting $\mathbf{x}_1$ (by reducing both right-hand side terms) may result in (3.4) and (3.5) retaining redundant variables.

Variable screening methods (Fan and Lv, 2008) may also be prone to selecting redundant variables. Screening ranks variables decreasingly based on the absolute values of their unconditional correlations to Y, selecting the top w variables (with w selected by CV, bootstrap, or BIC). Since $\text{corr}(\mathbf{x}_2, Y) \neq 0$ in the population, screening will rank $\mathbf{x}_2$ highly. However, it may not rank $\mathbf{x}_1$ highly because $\text{corr}(\mathbf{x}_1, Y) = 0$ in the population. Thus, some redundant variables may be ranked between $\mathbf{x}_2$ and $\mathbf{x}_1$, implying that if both $\mathbf{x}_1$ and $\mathbf{x}_2$ are selected, screening will select redundant variables.

The average L_0 solution path will not suffer the same problems. For convenience, assume $-\alpha_1/\alpha_2 > 0$ and $p/n = 100/200$ or smaller. For least angle regression, as $\|\beta_2\|_1$ increases at stage 1 (i.e., as $\mathbf{x}_2$ is 'partialled out' of Y), the unconditional correlation between $Y - \beta_2\mathbf{x}_2$ and $\mathbf{x}_1$ will increase above 0 significantly while the marginal correlation between $Y - \beta_2\mathbf{x}_2$ and any redundant variable will remain approximately 0. Thus, in the L_0 solution path and, hence, the average L_0 solution path, $\mathbf{x}_1$ will be included immediately after $\mathbf{x}_2$ is included. ■

Fan and Lv (2008) and Barut et al. (2016) propose two solutions for the problems with variable screening in situations like Example 2a. However,

- the first approach (Barut et al., 2016, Section 2.2 and 3) assumes the identity of $\mathbf{x}_2$ is known, which is unlikely to be realistic in practical applications. [In Bayesian networks or probabilistic graph modelling, $\mathbf{x}_2$ is known as a *collider*; Barut et al. (2016) refer to $\mathbf{x}_2$ as a *hidden signature* variable and denote it by X_c];
- the second approach (Barut et al., 2016, Section 1 and 2.2) suggests randomly trying out several variables to be colliders. The logic is straightforward: randomly trying out a wrong variable (like $\mathbf{x}_2$) to be a collider is harmless because conditioning on that variable will not make $corr(Y, \mathbf{x}_1) \neq 0$, nor will it cause the selection of a redundant variable. Moreover, by repeatedly randomly trying out variables, there is a non-zero probability the correct collider will eventually be uncovered, producing a statistically significant $corr(Y, \mathbf{x}_1) \neq 0$. However, using multiple trials may be inefficient and computationally expensive, especially with high-dimensional data. To improve high-dimensional efficiency, Barut et al. (2016) suggests trying out several variables simultaneously. However, if $corr(Y, \mathbf{x}_1) \neq 0$ were discovered after trying out, say, $\{\mathbf{x}_2, \text{other variables}\}$, it would still be necessary to decide which of $\{\mathbf{x}_2, \text{other variables}\}$ are redundant, meaning variable selection is not completed.

The second example illustrates another common problem in empirical regression: *redundant variables* that are *unconditionally correlated to* Y in the DGP. In Figure 7, the problem occurs because $\mathbf{x}_3$ and Y are determined by common variables. For example, house rent (Y) and food expenditure ($\mathbf{x}_3$) are both determined by income ($\mathbf{x}_1$) and saving ($\mathbf{x}_2$), yet $\mathbf{x}_3$ is redundant if $\mathbf{x}_1$ and $\mathbf{x}_2$ are used to predict Y. In this setting, Example 2b illustrates that the strong rule, base rule, and variable screening methods may struggle to reject the redundant $\mathbf{x}_3$ even when IRC is satisfied. By contrast, solar will be less prone to selecting redundant variables.

Example 2b. Figure 7 depicts the following confounding structure,

$$\begin{cases} \mathbf{x}_3 = \frac{1}{3}\mathbf{x}_1 + \frac{1}{3}\mathbf{x}_2 + \frac{\sqrt{7}}{3}u, \\ Y = \frac{7}{10}\mathbf{x}_1 + \frac{2}{10}\mathbf{x}_2 + \frac{\sqrt{47}}{10}e, \end{cases} \tag{3.6}$$

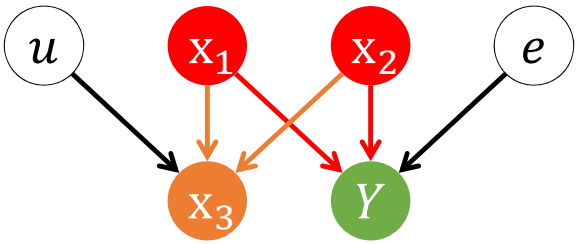


Figure 7: Y is unconditionally correlated with a redundant $\mathbf{x}_3$.

where $\mathbf{x}_1$ and $\mathbf{x}_2$ cause both Y and $\mathbf{x}_3$, implying that $\mathbf{x}_3$ is unconditionally correlated to Y; $\mathbf{x}_1$, $\mathbf{x}_2$, u and e are independent; $\mathbf{x}_3$ is independent from e; Y is independent from u; and all variables are standardized.

For large n, when the sample correlations are close to their population values, the sample marginal correlations to Y are:

$$\begin{aligned} &\text{corr}\,(\mathbf{x}_1, Y) = 0.7, \\ &\text{corr}\,(\mathbf{x}_3, Y) = \text{corr}\left(\frac{1}{3}\mathbf{x}_1 + \frac{1}{3}\mathbf{x}_2, \frac{7}{10}\mathbf{x}_1 + \frac{2}{10}\mathbf{x}_2\right) = 0.3, \\ &\text{corr}\,(\mathbf{x}_2, Y) = 0.2. \end{aligned} \tag{3.7}$$

Because $\mathbf{x}_2$ ranks below $\mathbf{x}_1$ and $\mathbf{x}_3$ in terms of marginal correlations to Y, the variable screening method must select all 3 variables—including the redundant $\mathbf{x}_3$—to avoid omitting $\mathbf{x}_2$. The base strong rule and safe rule may also have difficulty rejecting $\mathbf{x}_3$. Since $\text{corr}\,(\mathbf{x}_3, Y) > \text{corr}\,(\mathbf{x}_2, Y)$, if lasso selects $\mathbf{x}_3$ and $\mathbf{x}_2$ and the strong (or safe) rule is used to reject $\mathbf{x}_3$, $\mathbf{x}_2$ will also be rejected.

Forward regression, solar, and lasso will not make the same error. Because (3.6) does not violate the IRC, variable-selection consistency of forward regression, lars, and lasso is assured from the theoretical results of Zhang (2009) and Zhao and Yu (2006). In forward regression, $\mathbf{x}_1$ will be included at the first stage. After controlling for $\mathbf{x}_1$, the partial correlations (for large n) of both $\mathbf{x}_2$ and $\mathbf{x}_3$ with Y are:

$$\begin{aligned} &\text{corr}\,(\mathbf{x}_2, Y|\mathbf{x}_1) = \text{corr}\left(\mathbf{x}_2, \frac{2}{10}\mathbf{x}_2\right) = 0.2, \\ &\text{corr}\,(\mathbf{x}_3, Y|\mathbf{x}_1) = \text{corr}\left(\frac{1}{3}\mathbf{x}_1 + \frac{1}{3}\mathbf{x}_2, \frac{2}{10}\mathbf{x}_2\right) = 0.0667. \end{aligned} \tag{3.8}$$

Because $\text{corr}(\mathbf{x}_2, Y|\mathbf{x}_1) > \text{corr}(\mathbf{x}_3, Y|\mathbf{x}_1)$, forward regression will include $\mathbf{x}_2$ not $\mathbf{x}_3$ at the second stage. After controlling for both $\mathbf{x}_1$ and $\mathbf{x}_2$, the remaining variation in Y is due to e, which $\mathbf{x}_3$ cannot explain. Thus, CV or BIC will terminate forward regression after the second stage and $\mathbf{x}_3$ will not be selected. Similarly, because solar relies on the average L_0 path, it will include $\mathbf{x}_1$ and $\mathbf{x}_2$ but not $\mathbf{x}_3$. ■

Essentially, the strong rule, safe rule, and variable screening struggle in Examples 2a and 2b because they rely on unconditional correlations to Y, whereas informative variables in regression analysis

are defined in terms of conditional correlations. In many scenarios, unconditional and conditional correlations are aligned. However, when they are not, variable selection based conditional correlation is better placed to select the informative variables.

Fan and Lv (2008) propose redeeming variable screening on Y by first selecting variables with high unconditional correlations to Y and then running a lasso of the residuals on the dropped variables. By contrast, solar completes variable selection in a single pass of conditional correlation ranking, reducing computational costs. Moreover, the Fan and Lv (2008) approach does not solve Example 2b type problems. At the first step, variables with high unconditional correlations to Y will be selected, including the redundant $\mathbf{x}_3$. Selecting redundant variables will be more serious when Y has multiple $\mathbf{x}_3$-like siblings and in complicated dependence structures where multicollinearity results in inaccurate estimates of the coefficients and standard errors in finite samples. In short, solar is likely to be more computationally efficient and better at variable selection in settings with complicated dependence structures.

3.3. Robustness to the IRC

Solar is more robust to different settings of the IRC than the lasso. The IRC is considered to be sufficient and almost necessary for accurate lasso variable selection (Zhang, 2009). Here, we ignore lasso rules and variable screening since, as discussed above, their selection accuracy may be compromised by a reliance on unconditional correlations to Y. We define the IRC as in Zhang (2009).

Definition 3.1 (IRC)**.** Given $F \subset \{1, \ldots, p\}$, define X_F to be the $n \times |F|$ matrix with only the full set of informative variables. Define

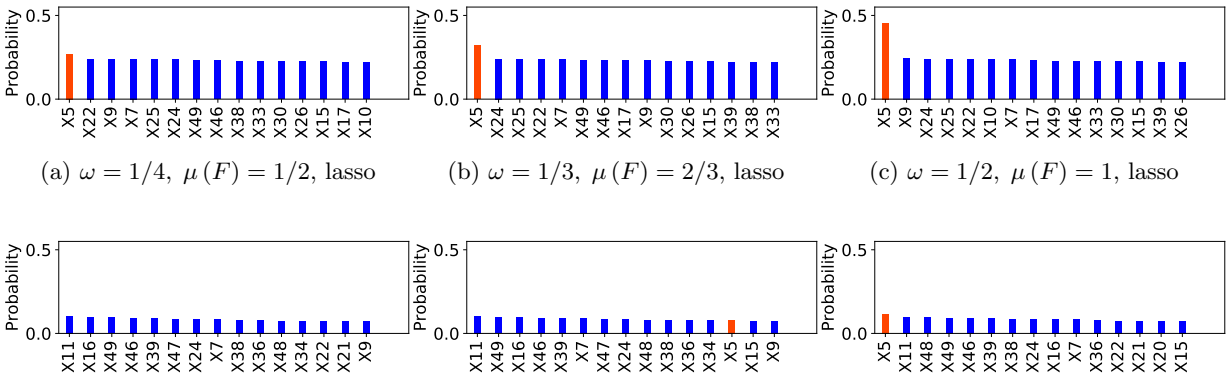
$$\mu(F) = \max \left\{ \left\| \left((X_F)^T X_F \right)^{-1} (X_F)^T \mathbf{x}_j \right\|_1 \mid \forall j \notin F \right\}.$$

Given a constant $1 \geqslant \eta > 0$, the *strong* irrepresentable condition is satisfied if $\mu(F) \leqslant 1 - \eta$ and the *weak* irrepresentable condition is satisfied if $\mu(F) < 1$.■

Example 3. Modify the DGP in Example 2b to match the Zhao and Yu (2006) simulations. Thus, $n = 200$, $p = 50$, and $\{\mathbf{x}_0, \ldots, \mathbf{x}_4, \mathbf{x}_6, \ldots, \mathbf{x}_{50}\}$ are generated from a zero-mean, unit-variance multivariate Gaussian distribution, where all the correlation coefficients are 0.5. The DGP of Y and $\mathbf{x}_5$ is

$$\begin{cases} \mathbf{x}_5 = \omega \mathbf{x}_0 + \omega \mathbf{x}_1 + \gamma \cdot \sqrt{1 - 2\omega^2} \\ Y = 2\mathbf{x}_0 + 3\mathbf{x}_1 + 4\mathbf{x}_2 + 5\mathbf{x}_3 + 6\mathbf{x}_4 + e \end{cases} \tag{3.9}$$

where $\omega \in \mathbb{R}$, while γ and e are both standard Gaussian noise terms, independent from each other and all the other variables. Compared with Example 2b, this DGP increases the challenge of accurate selection by increasing the number of redundant variables from 1 to 46, $\{\mathbf{x}_5, \ldots, \mathbf{x}_{50}\}$. This DGP also makes it straightforward to control the IRC through ω, which affects the value of $\mu(F)$.



(a) $\omega = 1/4,\ \mu(F) = 1/2$, lasso (b) $\omega = 1/3,\ \mu(F) = 2/3$, lasso (c) $\omega = 1/2,\ \mu(F) = 1$, lasso

(d) $\omega = 1/4,\ \mu(F) = 1/2$, solar (e) $\omega = 1/3,\ \mu(F) = 2/3$, solar (f) $\omega = 1/2,\ \mu(F) = 1$, solar

Figure 8: Probability of including redundant variables (top 15) in simulation 2 ($\mathbf{x}_5$ in orange).

In (3.9), the IRC only affects the redundant $\mathbf{x}_5$. Hence, we focus on the probability of incorrectly selecting $\mathbf{x}_5$ in 200 repetitions. By setting ω to either 1/4, 1/3, or 1/2, the population value of $\mu(F)$ changes, respectively, to 1/2, 2/3, or 1, gradually increasing the difficulty of rejecting the redundant $\mathbf{x}_5$.

Figure 8 displays the simulation results. When $\mu(F) = 1/2$, lasso wrongly includes $\mathbf{x}_5$ with probability 0.25. By contrast, $\mathbf{x}_5$ is not among the top 15 variables selected by solar, implying a probability less than 0.1. When $\mu(F)$ increases to 2/3, the probability lasso includes $\mathbf{x}_5$ increases to around 0.3. When $\mu(F)$ increases to 1 in the population and strong IRC is violated, the probability lasso includes $\mathbf{x}_5$ rises to almost 0.5. By contrast, the probability solar includes $\mathbf{x}_5$ is below 0.1 even when $\mu(F) = 1$. The results illustrate that solar is more robust to different settings of the IRC. ■

4. Solar advantages over subsample variable selection

In this section, we shift our focus to simulation. We demonstrate that: (i) solar offers significant improvements over lasso-type algorithms in terms of variable selection sparsity and accuracy; (ii) replacing lasso with solar in subsample selection drastically reduces the computation load, measured by runtime. We choose the simulation settings so that, as far as possible, the comparisons are fair, representative, and generalizable. Our overall goal is to enable *ceteris paribus* comparisons between solar and state-of-the-art lasso algorithms.

4.1. Simulation competitors

We consider a subset of lasso-type algorithms for comparison to solar. Firstly, some lasso modifications (e.g., fused lasso, grouped lasso) are designed to solve specific empirical problems that are not relevant to our paper. Secondly, it may be difficult to determine how much some variants outperform

lasso.[1] Since both solar and lasso may be evaluated via least angle regression and coordinate descent, many other lasso modifications can be directly applied to solar, as discussed in Section 2.3. We do not consider information criteria for shrinkage parameter tuning. Pedregosa et al. (2011) points out that information criteria are over-optimistic and require a proper estimation of the degrees of freedom for the solution. Moreover, information criteria are derived asymptotically and tend to break down when the problem is badly conditioned (e.g., $p > n$).[2]

Solar competes with K-fold, cross-validated lasso (denoted 'lasso' for short). Based on the Friedman et al. (2001) simulations that show $K = 10$ balances the bias-variance trade-off in CV error minimization, we choose the number of CV folds and the number of subsamples generated in Algorithm 1 to be 10. Least-angle regression and coordinate descent yield almost identical selection results for solar and lasso. Hence, we combine lars and coordinate descent results for solar, ignore the coordinate descent lasso (available in the supplementary file), and report only the runtime comparison between least angle regression and coordinate descent.

We also include bootstrap selection algorithms (aka bootstrap ensembles) in the comparisons. A bootstrap ensemble repeats lasso multiple times across bootstrap subsamples to produce a set of averaged (or accumulated) selection results. Given the similarities among lasso bootstrap ensembles, we choose the Bach (2008) bootstrap lasso (*bolasso*) to be the competitor to solar. Bach (2008) proposes two bolasso algorithms: bolasso-H and bolasso-S; both are competitors in the simulations. Bolasso-H selects only variables that are selected in all bootstrap subsamples, i.e., the subsample selection frequency threshold, $f = 1$. Bolasso-S selects variables that are selected in 90% of the bootstrap subsamples ($f = 0.9$). Bach (2008) finds that bolasso selection and prediction performance improves with the number of subsamples. To ensure a rigorous challenge for solar, we set the number of bootstrap subsamples in bolasso to 256, the maximum in the Bach (2008) simulations.

We also consider a bootstrap solar ensemble *(bsolar)*, which executes solar on each bootstrap subsample and computes the selection frequency for each variable across all bootstrap subsamples. To ensure that any performance difference is due to replacing lasso with solar in the ensemble system, bolasso and bsolar use the same subsample selection frequency threshold. Thus, we evaluate 2 versions of bsolar: bsolar-H ($f = 1$) and bsolar-S ($f = 0.9$). We use the notation bsolar-mH and bsolar-mS, where m is the number of subsamples used to compute the selection frequency.

[1] For example, while Jia and Yu (2010) show numerically that elastic net has slightly better variable-selection accuracy than lasso, they also find that "when the lasso does not select the true model, it is more likely that the elastic net does not select the true model either" (a point we verify in Section 5). While simulations in Zou (2006) show that adaptive lasso outperforms lasso when $p/n < 1$, it requires first computing the OLS estimates of all $\mathbf{x}_i$ coefficients, which is difficult when $p/n > 1$.

[2] See https://scikit-learn.org/stable/modules/linear_model#lasso.html for details.

4.2. Simulation settings

The DGP for the simulations is as follows. The p covariates in $X \in \mathbb{R}^{n \times p}$ are generated from a zero-mean, multivariate Gaussian distribution, with all off-diagonal elements in the covariance matrix equal to 0.5. The first 5 variables in X are informative; the remaining $p - 5$ variables are redundant. The response variable $Y \in \mathbb{R}^{n \times 1}$ is:

$$Y = 2\mathbf{x}_0 + 3\mathbf{x}_1 + 4\mathbf{x}_2 + 5\mathbf{x}_3 + 6\mathbf{x}_4 + e, \tag{4.1}$$

where $e \in \mathbb{R}^{n \times 1}$ is a standard Gaussian noise term. All data points are independently and identically distributed. Each $\mathbf{x}_i$, $i = 1, \ldots, p$, is independent from the noise term e, which is standard Gaussian. Simulations are repeated 200 times with fixed Python random generators across simulations.

We vary the data dimensions p/n as follows. In the first block of simulations, p/n approaches 0 from above, corresponding to the classical $p < n$ setting. In the second block, p/n approaches 1 from above, corresponding to high dimension settings. In the third block, $p/n = 2$ as $\log(p)/n$ slowly approaches 0, corresponding to ultrahigh dimension settings, i.e., where $(p - n) \to \infty$.

We compare the performance of solar and lasso in terms of sparsity and accuracy of variable selection and on the runtime. Sparsity is measured by the mean number of selected variables. Discovery accuracy is measured by the mean number of *informative* selected variables. Purge accuracy is measured by the mean number of *redundant* selected variables (equal to sparsity minus discovery accuracy). Runtime is measured by mean CPU time. The raw simulation results are available in the supplementary file.

4.3. Programming languages, parallelization, and hardware

To ensure a credible comparison between solar and the lasso competitors, we choose the hardware and software settings to maximize the computation speed of lasso. We show that, even under the ideal computation environment for lasso, solar exhibits a substantial runtime advantage.

To maximize computation speed, we use `Numpy`, `Scipy`, and `Cython`—all well-known for performance and speed—to outsource all numerical and matrix operations to the Intel Math Kernel Library, currently the fastest and most accurate C++/Fortran library for CPU numerical operations.

To reduce the possibility of CPU and RAM bottlenecks in parallel computing of lasso and bootstrap lasso, we code in Python rather than R. Donoho (2017) claims: "R has the well-known reputation of being less scalable than Python to large problem sizes". Given the simulations repeat solar, lasso, and bootstrap lasso many times to arrive at representative performance measures, choosing Python over R mitigates the impact of hardware limitations. Computations are executed with an Intel Xeon W-3245 CPU with 3.2GHz base frequency, 10-processor parallelization, and 64GB RAM, further reducing the possibility of CPU-RAM bottlenecks.

To guarantee the programming quality of the lasso implementation, we source lasso and bootstrap lasso from the Sci-kit learn library (Pedregosa et al., 2011) of efficient machine-learning tools.[3] Used widely in research and industry, Sci-kit learn also uses `Numpy`, `Scipy`, and `Cython` to delegate all numerical and matrix operations to Fortran/C++.

Lastly, to optimize computation and avoid large overheads, we implement 10-processor parallelization. Because each realization of solar and CV-lasso requires 10 repetitions of lars or coordinate descent, optimization must be carried out sequentially, which means that each realization of solar or lasso must wait for the preceding realization to finish. Thus, we design a parallel architecture to assign one realization per CPU core. The design is optimized for a CPU with at least 10 cores and may generate an overhead with fewer cores. For example, in each realization, an 8-core CPU would assign two cores to training the second repetition of coordinate descent, with the other 6 cores left idle until the 10 repetitions were completed.

4.4. Comparison of sparsity and accuracy

Table 1 summarizes average selection performance.[4] While all the competitors always include the 5 informative variables, solar outperforms lasso in terms of sparsity in every p/n scenario, implying superior ability to limit the selection of redundant variables. Notably, lasso sparsity deteriorates as $p/n \to 1$, while solar sparsity improves over the same range. While the sparsity of all the competitors deteriorates as $\log(p)/n \to 0$, solar maintains a clear advantage over lasso.

Table 1: Simulation results for sparsity and accuracy.

	$p/n \to 0$			$p/n \to 1$			$\log(p)/n \to 0$		
	$\frac{100}{100}$	$\frac{100}{150}$	$\frac{100}{200}$	$\frac{150}{100}$	$\frac{200}{150}$	$\frac{250}{200}$	$\frac{400}{200}$	$\frac{800}{400}$	$\frac{1200}{600}$
mean number of selected variables									
lasso	19.73	19.84	19.54	22.30	23.57	26.56	28.92	33.88	37.96
solar	9.86	8.66	8.50	11.34	9.8	8.2	10.54	13.28	15.52
bolasso-S	5.46	6.09	6.60	5.46	6.09	6.6	5.58	6.63	7.67
bolasso-H	5	5.02	5.01	5	5.02	5.01	5	5.01	5.02
bsolar-3S/3H	5.44	5.18	5.22	5.44	5.18	5.22	5.25	5.86	6.09
bsolar-5S/5H	5.14	5.07	5.1	5.14	5.07	5.1	5.08	5.28	5.46
bsolar-10S	5.12	5.04	5.04	5.12	5.04	5.04	5.05	5.24	5.39
bsolar-10H	5.06	5.01	5	5.06	5.01	5	5.01	5.09	5.17
mean number of selected informative variables									
lasso	5	5	5	5	5	5	5	5	5
solar	5	5	5	5	5	5	5	5	5
bolasso-S/H	5	5	5	5	5	5	5	5	5
bsolar-3S/3H/5S/5H/10S/10H	5	5	5	5	5	5	5	5	5

[3]Detail is available at `https://scikit-learn.org/stable/`.

[4]Detailed histograms are available in the supplementary file.

Table 1 also confirms the advantage of solar over lasso in bootstrap ensembles. Bolasso-S stands out with the poorest sparsity while the others perform almost identically. Recall that bolasso requires 256 subsample lasso repetitions compared with bsolar-3/5/10, which take only 3, 5, and 10 subsample solar repetitions, respectively. Hence, bsolar reduces subsample repetitions by 96% relative to bolasso. As we show in Section 4.6, solar and lasso have identical computation loads. If time complexity decreases linearly, the 96% reduction in subsample repetitions implies at least a 96% reduction in computation time for solar relative to bolasso.

4.5. Explanation of the efficiency discrepancy between bolasso-bsolar

The efficiency of bsolar is due to its unique, intrinsic multi-layer variable ranking scheme. While bsolar and bolasso both generate bootstrap subsamples, bsolar uses a different bootstrap variable selection procedure. Specifically,

- solar executes Algorithm 1 or 3 on each bootstrap subsample and ranks variables using the average L_0 path, which we call the *internal ranking*. The internal ranking identifies the strongest signals on each bootstrap subsample.
- bsolar collects the internal ranking results to produce an overall ranking, which we call the *external ranking*. The external ranking identifies the strongest signals on the majority of bootstrap subsamples.

Compared to the usual (one-layer) ranking methods (Fan and Lv, 2008; Hall et al., 2009; Hall and Miller, 2009; Li et al., 2012a,b), our multi-layer method balances efficiency, robustness, and stability. First, one-layer methods rank variables on the whole sample. By contrast, the internal ranking uses the average L_0 path, which, as discussed in Section 2.1, improves robustness to multicollinearity, noise, and sample size. Second, one-layer methods select variables immediately after ranking; our method performs a second external ranking that focuses on the signals that are persistently strong regardless of sampling variation. Third, as shown in Section 3.2, internal ranking avoids issues caused by complicated dependence structures that other (unconditional) ranking methods cannot.

Furthermore, research shows the clear efficiency advantage of one-layer variable ranking methods, especially with large p and n. Hence, by embedding ranking into bootstrap variable selection, bsolar reduces the number of bootstrap repetitions required by bolasso. Moreover, as shown in Table 2, bsolar produces a shorter and more accurate list of subsample variable selection frequencies.

Table 2 illustrates numerically how efficient multi-layer ranking is. Table 2a breaks down the subsample selection frequency list from 256 subsamples for one bolasso realization with $p/n = 100/200$. Due to the length of the list, we report only subsample selection frequencies ≥ 0.69. With only one layer of ranking, bolasso is unable to separate informative from redundant variables even with 256

Table 2: Subsample variable selection frequencies for bolasso and bsolar-10.

(a) bolasso

frequency	variables
$\geqslant 1.00$	$\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0$
$\geqslant 0.88$	$\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_{28}$
$\geqslant 0.84$	$\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_{28}, \mathbf{x}_{71}$
$\geqslant 0.76$	$\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_{28}, \mathbf{x}_{71}, \mathbf{x}_{91}$
$\geqslant 0.70$	$\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_{28}, \mathbf{x}_{71}, \mathbf{x}_{91}, \mathbf{x}_{94}$
$\geqslant 0.69$	$\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_{28}, \mathbf{x}_{71}, \mathbf{x}_{91}, \mathbf{x}_{94}, \mathbf{x}_{70}, \mathbf{x}_{40}$
$\vdots$	$\vdots$

(b) bsolar-10

frequency	variables
$\geqslant 1.00$	$\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0$
$\geqslant 0.10$	$\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1, \mathbf{x}_0, \mathbf{x}_{91}, \mathbf{x}_{71}$
$= 0$	all other variables

subsample repetitions. The frequency discrepancy for bolasso between the highest-ranking redundant ($\mathbf{x}_{28}$) and the lowest-ranking informative variable ($\mathbf{x}_0$) is only 0.12. By contrast, Table 2b shows bsolar-10 returns a much shorter list with a frequency discrepancy between the highest-ranking redundant ($\mathbf{x}_{91}$) and the lowest-ranking informative variable ($\mathbf{x}_0$) of 0.9. To increase the discrepancy between the lowest ranked informative and highest ranked redundant variables for bolasso, Bach (2008) suggests raising the number of subsample repetitions. However, increasing repetitions will raise the bolasso computation load in high-dimensional spaces, increasing the advantage of bsolar.

4.6. Computation load comparison

Since the computation load for lars or coordinate descent on a given sample is fixed, we may use the number of lars or coordinate descents to approximate the computation load for solar and lasso. For comparison, we compute solar with K subsamples and lasso with K-fold cross-validation. As shown in Algorithm 1 and 3, solar computes one lars or coordinate descent on each subsample (X^k, Y^k), which implies K lars or coordinate descents to compute $\widehat{q}$ and one more pass to compute c^* for variable selection. Lasso also requires computing K lars or coordinate descents to optimize the tuning parameter and, given the optimal tuning parameter, one more pass on the full sample to select variables. Thus, solar and lasso have the same computation load.

Given equal computation loads for lasso and solar, differences between bolasso and bsolar are due primarily to the number of subsample repetitions (SR). Solar and lasso have a computation load of 1 SR, bolasso has a load of 256 SR, and bsolar-3/5/10 has a load of 3/5/10 SR.

Table 3 shows the average runtime for the simulations. Bsolar-3 has a much shorter runtime than bolasso. The runtime differences are even more pronounced when p and n increase. The 256 subsample repetitions render the bolasso selection algorithms computationally infeasible even with moderate p and n. By contrast, bsolar-3 requires only 30 realizations of lars or coordinate descent. Due to a lighter computational load and CPU usage, bsolar-3 allows the CPU to work at a higher frequency

Table 3: Simulation results for computation load (mean runtime in seconds).

	$p/n \to 0$			$p/n \to 1$			$\log(p)/n \to 0$		
	$\frac{100}{100}$	$\frac{100}{150}$	$\frac{100}{200}$	$\frac{150}{100}$	$\frac{200}{150}$	$\frac{250}{200}$	$\frac{400}{200}$	$\frac{800}{400}$	$\frac{1200}{600}$
bsolar-3	0.11	0.16	0.15	0.10	0.15	0.22	0.41	0.74	1.48
bolasso (lars, 256 SR)	9.52	12.49	10.61	10.01	13.92	19.72	23.10	184.59	502.56
bolasso (cd, 256 SR)	13.49	60.51	60.35	13.92	16.85	20.17	27.73	100.58	308.12

than bolasso, decreasing the runtime for each lars realization.

4.6.1. Comparison with previous lasso computation research

The bolasso findings is consistent with Tibshirani (2015), further confirming with previous research on lasso. Given the same convergence criteria (tolerance for optimization and maximum number of iterations), number of folds for CV ($K = 10$), and number of λs in the grid search (100), the time complexity of lasso is mostly determined by n, p, and pairwise correlations among the covariates ($corr$). For the purposes of comparison, we consider a Gaussian regression with $p/n = 1000/100$ and $corr = 0.5$.

- Friedman et al. (2010, Table 1) shows that, on a 2-core Intel Xeon CPU with 2.8GHz frequency, the average runtime is 0.07 seconds for one pathwise coordinate descent realization (with covariance pre-computed for updating).[5] Their package is coded in R and all numerical computations are executed in Fortran/C++.

- Using an Intel Xeon W-3245 CPU with 3.2GHz frequency and 16 cores, the average runtime for the coordinate descent bolasso package is 41.92 seconds (with covariance pre-computed automatically), accounting for 256 realizations of 10-fold, cross-validated lasso (namely 2,816 pathwise coordinate descent realizations). Thus, the average runtime is 0.014 seconds per pathwise coordinate descent.

Thus, with similar CPU frequency and 14 additional cores, our implementation produces an average speedup of $0.07/0.014 = 5$ times over Friedman et al. (2010) for each pathwise coordinate descent repetition.

Given the CPU core number and frequencies, our speedup almost reaches the theoretical maximum. To confirm this point, we calculate the theoretical maximum speedup from 2 to 16 cores using

[5]Our 2-core R replications are around 0.01 seconds slower than Friedman et al. (2010) on a CPU with 3.2GHz frequency. Because we lack configuration details on the interpreters and compilers Friedman et al. (2010) used for R, openBLAS, Fortran, and C++, we use their average runtime of 0.07 seconds for comparison.

Amdahl's law. Friedman et al. (2010) and our code apply the same desgin to cross-validated lasso. Both require 10 pathwise coordinate descent repetitions to optimize λ and, based on the results of the first 10, one extra to compute β. Thus, approximately 11% of the total computations (including 1 of the 11 pathwise coordinate descent repetitions, data generation, matrix manipulation, I/O, code interpretation to C++/Fortran, etc.) is not parallelizable. With a given computation load (n and p), according to Amdahl's law the maximum speedup is:

$$\frac{1}{\rho + (1-\rho)/s} = \frac{1}{0.11 + (1-0.11)/(16/2)} \approx 4.5, \tag{4.2}$$

where ρ is the proportion of computation that is not parallelizable and s is the computation speedup for the parallelizable proportion (e.g., the core number multiple). Given that our CPU base frequency is also higher than Friedman et al. (2010, Table 1) (3.2GHz over 2.8GHz), we adjust the maximum speedup by the frequency multiple (3.2/2.8), resulting in a final maximum speedup of $4.5 \times 3.2/2.8 \approx 5.2$, or 4% faster than our speedup. Thus, our coordinate descent bolasso package achieves almost 96% of the maximum possible speedup, implying little room for further parallelization improvements.

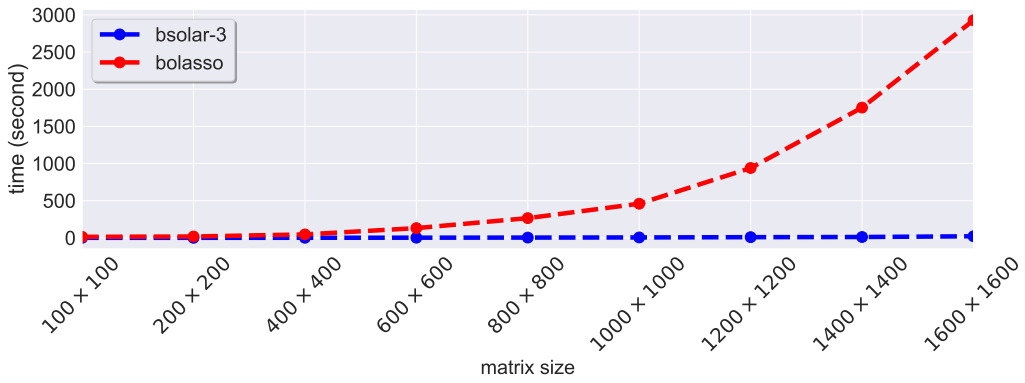


Figure 9: Average runtime (per pathwise coordinate descent) comparison for different X matrix sizes.

Nonetheless, bolasso still requires a large computation time with $p/n = 1200/600$ due to the 256 CV-lasso repetitions, once again confirming the bsolar computation advantage. Figure 9 plots average runtime against the size of the X matrix. As matrix size increases, bsolar runtime increases linearly while bolasso rises exponentially, confirming the bsolar-3 advantage for high-dimensional data.

5. Real-world data: Sydney house price prediction

To demonstrate that the improvements from solar are empirically feasible, we apply solar to real-world data. The real-world data reflect both the $p/n \to 0$ scenarios as well as the challenging IRC settings, complicated dependence structures, and grouping effects typical of data in the social sciences.

The database is assembled from multiple sources. The primary source comprises real estate market transaction data for 11,974 Sydney, Australia, houses sold in 2010, including price and house attribute information (GIS coordinates, property address, bedrooms, bathrooms, car spaces, etc.). Each property is GIS-matched with: 2011 census data by Statistical Area Level 1 (the smallest census area in Australia, comprising at most 200 people or 60 households); 2010 and 2011 crime data by suburb; 2010 geo-spatial information on topology, climate, pollution, and aircraft noise; Google Maps data; 2009 primary and secondary school data; and 2010 Sydney traffic and public transport data (bus routes, train stations, and ferry wharfs). We predict house price with a linear model.

Using an ensemble of Bayes network learning algorithms for data cleaning, we reject variables with both very low conditional and unconditional correlations to house price. The remaining variables are listed in the first column of Table 4.[6] The 57 variables fall into 5 broad categories: house attributes, distance to key locations (public transport, shopping, etc.), neighbourhood socioeconomic data, localized administrative and crime data, and local school quality. Pairwise correlations among all 57 covariates indicate, not surprisingly, severe multicollinearity and grouping effects, implying a harsh IRC setting.[7] Thus, heuristically increasing the value of the tuning parameter in lasso-type estimators (e.g., using the one-sd or the 'elbow' rule) is unlikely to be useful since it may trigger further grouping effects and the random dropping of variables.

Table 4 shows the selection comparison across the elastic net, lasso, and solar. With all variables in linear form, both lasso and elastic net lose sparsity, likely due to the complicated dependence structures and severe multicollinearity in the data, accordant with Jia and Yu (2010). By contrast, solar returns a much sparser model, with only 9 variables selected from 57. Very similar results are found with the variables in log form, hinting that solar possesses superior selection sparsity and robustness to a change in functional form. More importantly, solar variable selection outperforms the lasso-type estimators in terms of the balance between sparsity and prediction power. While pruning 25-48 variables from the elastic net and lasso selections, the post-selection regression R^2 for solar falls by just 3-5%.

6. Conclusion

In this paper we propose a new algorithm for high-dimensional data called solar (subsample-ordered least-angle regression). We show that solar yields substantial improvements over lasso in terms of the sparsity, stability, accuracy, and robustness of variable selection. We also illustrate analogous improvements from solar ensembles relative to lasso ensembles.

Detection of weak signals is a potential weakness evident in solar, although relative to the lasso

[6] Due to the 200GB size of the database, we include only the data for these variables in the supplementary file.

[7] Correlations and IRC are also reported in supplementary files.

competitor the difference is very slight. Nonetheless, we are working on an extension to solar, the double-bootstrap solar (DBsolar), which, if early results are any indication, promises to enable solar accurately to detect variables with weak signals.

References

Bach, F.R., 2008. Bolasso: model consistent lasso estimation through the bootstrap, in: Proceedings of the 25th international conference on Machine learning, ACM. pp. 33–40.

Barber, R.F., Candès, E.J., 2019. A knockoff filter for high-dimensional selective inference. The Annals of Statistics 47, 2504–2537.

Barut, E., Fan, J., Verhasselt, A., 2016. Conditional sure independence screening. Journal of the American Statistical Association 111, 1266–1277.

DiCiccio, C.J., DiCiccio, T.J., Romano, J.P., 2020. Exact tests via multiple data splitting. Statistics & Probability Letters 166, 108865.

Donoho, D., 2017. 50 years of data science. Journal of Computational and Graphical Statistics 26, 745–766.

Draper, N., Smith, H., 1966. Applied regression analysis new york: Wiley .

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least angle regression. Annals of Statistics 32, 407–499.

Efroymson, M., 1966. Stepwise regression–a backward and forward look, in: Eastern Regional Meetings of the Institute of Mathematical Statistics, pp. 27–29.

Fan, J., Lv, J., 2008. Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70, 849–911.

Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., et al., 2007. Pathwise coordinate optimization. Annals of applied statistics 1, 302–332.

Friedman, J., Hastie, T., Tibshirani, R., 2001. The elements of statistical learning. volume 1 of *Springer Series in Statistics*. Springer-Verlag New York.

Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software 33, 1.

Ghaoui, L.E., Viallon, V., Rabbani, T., 2010. Safe feature elimination for the lasso and sparse supervised learning problems. arXiv preprint arXiv:1009.4219 .

Hall, P., Miller, H., 2009. Using generalized correlation to effect variable selection in very high dimensional problems. Journal of Computational and Graphical Statistics 18, 533–550.

Hall, P., Miller, H., et al., 2009. Using the bootstrap to quantify the authority of an empirical ranking. The Annals of Statistics 37, 3929–3959.

Ing, C.K., Lai, T.L., 2011. A stepwise regression method and consistent model selection for high-dimensional sparse linear models. Statistica Sinica , 1473–1513.

Jia, J., Yu, B., 2010. On model selection consistency of the elastic net when $p >> n$. Statistica Sinica , 595–611.

Li, G., Peng, H., Zhang, J., Zhu, L., et al., 2012a. Robust rank correlation based screening. The Annals of Statistics 40, 1846–1877.

Li, R., Zhong, W., Zhu, L., 2012b. Feature screening via distance correlation learning. Journal of the American Statistical Association 107, 1129–1139.

Lockhart, R., Taylor, J., Tibshirani, R.J., Tibshirani, R., 2014. A significance test for the lasso. Annals of Statistics 42, 413–468.

Meinshausen, N., Bühlmann, P., 2010. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72, 417–473.

Meinshausen, N., Meier, L., Bühlmann, P., 2009. P-values for high-dimensional regression. Journal of the American Statistical Association 104, 1671–1681.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Romano, J.P., DiCiccio, C., 2019. Multiple data splitting for testing. Department of Statistics, Stanford University.

Taylor, J., Lockhart, R., Tibshirani, R.J., Tibshirani, R., 2014. Exact post-selection inference for forward stepwise and least angle regression. arXiv preprint arXiv:1401.3889 7, 2.

Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., Tibshirani, R.J., 2012. Strong rules for discarding predictors in lasso-type problems. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 74, 245–266.

Tibshirani, R.J., 2015. A general framework for fast stagewise algorithms. J. Mach. Learn. Res. 16, 2543–2588.

Tropp, J.A., 2004. Greed is good: Algorithmic results for sparse approximation. IEEE Transactions on Information theory 50, 2231–2242.

Wainwright, M.J., 2009. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso). IEEE transactions on information theory 55, 2183–2202.

Wang, J., Zhou, J., Liu, J., Wonka, P., Ye, J., 2014. A safe screening rule for sparse logistic regression, in: Advances in neural information processing systems, pp. 1053–1061.

Wasserman, L., Roeder, K., 2009. High dimensional variable selection. Annals of statistics 37, 2178.

Weisberg, S., 2004. Discussion following "Least angle regression," by B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Annals of Statistics 32, 490–494.

Xu, G., Huang, J.Z., et al., 2012. Asymptotic optimality and efficient computation of the leave-subject-out cross-validation. The Annals of Statistics 40, 3003–3030.

Yuan, M., Lin, Y., 2007. On the non-negative garrotte estimator. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69, 143–161.

Zeng, Y., Yang, T., Breheny, P., 2017. Efficient feature screening for lasso-type problems via hybrid safe-strong rules. arXiv preprint arXiv:1704.08742 .

Zhang, T., 2009. On the consistency of feature selection using greedy least squares regression. Journal of Machine Learning Research 10, 555–568.

Zhao, P., Yu, B., 2006. On model selection consistency of Lasso. Journal of Machine Learning Research 7, 2541–2563.

Zou, H., 2006. The adaptive lasso and its oracle properties. Journal of the American statistical association 101, 1418–1429.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical methodology) 67, 301–320.

Table 4: Variable selection results for linear and log house price models.

		elastic net		lasso		solar	
Variable	Description	linear	log	linear	log	linear	log
Bedrooms	property, number of bedrooms	✓	✓	✓	✓	✓	✓
Baths	property, number of bathrooms	✓	✓	✓	✓	✓	✓
Parking	property, number of parking spaces	✓	✓	✓	✓	✓	✓
AreaSize	property, land size	✓	✓	✓	✓		
Airport	distance, nearest airport	✓	✓	✓	✓		
Beach	distance, nearest beach	✓	✓	✓	✓	✓	✓
Boundary	distance, nearest suburb boundary	✓	✓	✓	✓		
Cemetery	distance, nearest cemetery	✓		✓			
Child care	distance, nearest child-care centre	✓	✓	✓	✓		✓
Club	distance, nearest club	✓	✓	✓	✓		
Community facility	distance, nearest community facility	✓	✓				
Gaol	distance, nearest gaol	✓	✓			✓	✓
Golf course	distance, nearest golf course	✓	✓	✓	✓		
High	distance, nearest high school	✓	✓	✓	✓		
Hospital	distance, nearest general hospital	✓	✓		✓		
Library	distance, nearest library	✓		✓			
Medical	distance, nearest medical centre	✓	✓		✓		
Museum	distance, nearest museum	✓	✓	✓	✓		
Park	distance, nearest park	✓	✓	✓			
PO	distance, nearest post office	✓	✓		✓		
Police	distance, nearest police station	✓	✓	✓	✓		
Pre-school	distance, nearest preschool	✓	✓	✓	✓		
Primary	distance, nearest primary school	✓	✓	✓	✓		
Primary High	distance, nearest primary-high school	✓	✓	✓	✓		
Rubbish	distance, nearest rubbish incinerator	✓	✓	✓			
Sewage	distance, nearest sewage treatment	✓					
SportsCenter	distance, nearest sports centre	✓	✓	✓	✓		
SportsCourtField	distance, nearest sports court/field	✓		✓	✓		
Station	distance, nearest train station	✓		✓			
Swimming	distance, nearest swimming pool	✓	✓	✓	✓		
Tertiary	distance, nearest tertiary school	✓	✓	✓	✓		
Mortgage	SA1, mean mortgage repayment (log)	✓	✓	✓	✓	✓	✓
Rent	SA1, mean rent (log)	✓	✓	✓	✓	✓	✓
Income	SA1, mean family income (log)	✓	✓	✓	✓	✓	✓
Income (personal)	SA1, mean personal income (log)	✓					
Household size	SA1, mean household size	✓	✓	✓	✓		
Household density	SA1, mean persons to bedroom ratio	✓	✓	✓	✓		
Age	SA1, mean age	✓	✓	✓	✓		✓
English spoken	SA1, percent English at home	✓		✓			
Australian born	SA1, percent Australian-born	✓		✓			
Suburb area	suburb area	✓		✓	✓		
Population	suburb population	✓	✓		✓		
TVO2010	suburb total violent offences, 2010	✓					
TPO2010	suburb total property offences, 2010	✓	✓		✓		
TVO2009	suburb total violent offences, 2009	✓	✓	✓			
TPO2009	suburb total property offences, 2009	✓	✓				
ICSEA	local school, socio-educational advantage	✓	✓	✓	✓	✓	✓
ReadingY3	local school, year 3 mean reading score	✓	✓	✓	✓		
WritingY3	local school, year 3 mean writing score	✓	✓	✓	✓		
SpellingY3	local school, year 3 mean spelling score	✓	✓	✓			
GrammarY3	local school, year 3 mean grammar score	✓		✓			
NumeracyY3	local school, year 3 mean numeracy score	✓	✓	✓	✓		
ReadingY5	local school, year 5 mean reading score	✓					
WritingY5	local school, year 5 mean writing score	✓	✓	✓			
SpellingY5	local school, year 5 mean spelling score	✓	✓	✓			
GrammarY5	local school, year 5 mean grammar score	✓	✓	✓			
NumeracyY5	local school, year 5 mean numeracy score	✓					
	Number of variables selected	57	45	44	36	9	11
	post-selection OLS R^2	0.55	0.76	0.55	0.76	0.50	0.73
	Sample size			11,974			

Appendix A L_0 path ranking accuracy and variable selection consistency

For generality, we derive the solar theoretical properties under the general framework of forward selection by Zhang (2009). Our proof method is summarized as follows. For various settings and assumptions, Tropp (2004), Yuan and Lin (2007), Wainwright (2009), Zhang (2009), and Ing and Lai (2011) have shown: (i) forward selection is variable selection consistent in different modes, and (ii) informative variables are ranked at earlier stages of the solution path than redundant variables. Since the L_0 path on each solar subsample is essentially the re-parameterized forward selection path, (i) and (ii) can be applied directly to the L_0 path on each solar subsample. As a result, we can build a probabilistic lower bound to show that, on average, (i) and (ii) also hold for the average L_0 path and the variable selection result on the average L_0 path (including solar variable selection). While we follow the methods of Tropp (2004), Wainwright (2009), and Zhang (2009), due to their alignment and direct comparison to the L_0 and L_2 consistency of lasso-type estimators, our proof method is compatible to other forward selection results.

We adopt the notation from Zhang (2009) as follows:

Definition A.1. Consider the regression model $Y = X\beta + \mathbf{e}$,

1. The regression coefficients of the data generating process (DGP) are $\overline{\beta} = \left[\overline{\beta}_1, \ldots, \overline{\beta}_{p_1}, \mathbf{0}\right]^T \in \mathbb{R}^{p\times 1}$, where the first p_1 entries are not 0.
2. $Y = [y_1, \ldots, y_n]^T \in \mathbb{R}^{n\times 1}$ is the response variable.
3. The data matrix is $X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] \in \mathbb{R}^{n\times p}$ with columns $\mathbf{x}_j \in \mathbb{R}^{n\times 1}$, $\forall j = 1, \ldots, p$ and rows $X_{i,\bullet} \in \mathbb{R}^{1\times p}$, $\forall i = 1, \ldots, n$. $\mathbf{e} \in \mathbb{R}^{n\times 1}$ is the stochastic noise.
4. The support, $\forall \beta \in \mathbb{R}^{p\times 1}$, is $\text{supp}(\beta) = \{j : \beta_j \neq 0\}$
5. Given $X \in \mathbb{R}^{n\times p}$ and $F \subset \{1, \ldots, p\}$,
$$\widehat{\beta}_X(F, Y) = \text{argmin}_{\beta\in\mathbb{R}^d} \frac{1}{n}\|X\beta - Y\|_2^2 \quad \text{subject to} \quad \text{supp}(\beta) \subset F.$$
That is, $\widehat{\beta}_X(F, Y)$ is the least squares solution with coefficients restricted to F.
6. $|F|$ is the cardinality of F and $\overline{F} - F$ is the difference of sets $\overline{F}$ and F.
7. X_F is the $n \times |F|$ matrix, whose columns are $\mathbf{x}_j \in [\mathbf{x}_0, \ldots \mathbf{x}_p]$ with $j \in F$ arranged in the ascending order.
8. To introduce the irrepresentable condition and sparse eigenvalue condition, define
$$\mu_X(F) = \max_{j\in F} \left\| \left(X_F^T X_F\right)^{-1} X_F^T \mathbf{x}_j \right\|_1$$
and
$$\rho_X(F) = \inf\left\{ \frac{1}{n}\|X\beta\|_2^2 / \|\beta\|_2^2 : \text{supp}(\beta) \subset F \right\}$$

9. (the ϵ stopping rule) The forward selection algorithm we dicuss is also referred to as orthogonal matching pursuit (OMP) and presented at Zhang (2009, Figure 1). Its stopping rule is denoted by ϵ. Prior to stage l, forward selection finds the unselected variable

$$\mathbf{x}^{(l)} = \text{argmax}_{\mathbf{x}_j} \quad \left|\mathbf{x}_j^T u^{(l-1)}\right|, \quad \text{for all unselected} \quad \mathbf{x}_j,$$

where $u^{(l-1)}$ is the forward regression residual of stage $l-1$. If the maximum value $\geqslant \epsilon$, the forward selection loop will select $\mathbf{x}^{(l)}$, compute $u^{(l)}$ and move on to stage $l+1$; otherwise, the forward selection loop will stop and report the regression coefficients and the selected variables on and before stage $l-1$.

We also adopt the Zhang (2009) assumptions:

[A1] each $\mathbf{x}_i$ is normalized such that $\|\mathbf{x}_j\|_2^2/n = 1,\ \forall j = 1, \ldots, p$;

[A2] the $\overline{\beta}$ is sparse : $\exists \overline{\beta} \in \mathbb{R}^{p\times 1}$ with $\overline{F} = \text{supp}\left(\overline{\beta}\right)$ such that $\mathbf{E}(Y) = X\overline{\beta} = \left[X_{1,\cdot}\,\overline{\beta},\ \ldots,\ X_{n,\cdot}\,\overline{\beta}\right]^T$;

[A3] the irrepresentable condition (Tropp, 2004): $\mu_X\left(\overline{F}\right) < 1$; the sparse eigenvalue condition (Wainwright, 2009): $\rho_X\left(\overline{F}\right) > 0$.

[A4] $\exists \sigma > 0$ such that $Y = [Y_1, \ldots, Y_n]$ are independent (but not necessarily identically distributed) sub-Gaussians with $\mathbb{E}(Y_i) = X_{i,\cdot}\,\overline{\beta}$ and $\mathbb{E}_{Y_i}\left(e^{t(Y_i - \mathbb{E}(Y_i))}\right) \leqslant e^{\sigma^2 t^2/2}$, $\forall t \in \mathbb{R}$ and $\forall i \in \{1, \ldots, n\}$.

A4 implies that Y can be either unbounded or bounded. **A2** and **A4** imply that the regression noise in DGP $[e_1, \ldots, e_n]^T$ are independent sub-Gaussians.

The proof for variable selection accuracy and variable ranking accuracy is specified in the following steps.

Step 1 : reparametrize ϵ stopping rule via $\widehat{q}^k$

We firstly introduce the theoretical result of Zhang (2009, Theorem 1), which shows that, for a general forward selection on a finite sample, the probability of omitting informative variables are under control.

Theorem A.1. *(Zhang, 2009) Consider the forward selection algorithm with Assumption 1 satisfied. Given any $\eta \in (0,1)$, with probability larger than $1-\eta$, if the ϵ stopping criterion stops forward selection at stage l, satisfying*

$$\epsilon > \frac{1}{1-\mu_X\left(\overline{F}\right)}\sigma\sqrt{2\ln(4p/\eta)} \tag{A.1}$$

and

$$\min_{j\in\overline{F}}\left|\overline{\beta}_j\right| \geqslant \frac{3\epsilon}{\rho_X\left(\overline{F}\right)\cdot\sqrt{n}}, \tag{A.2}$$

then when the procedure stops at stage l,

$$\overline{F} = F^{(l-1)},$$

where $F^{(l-1)}$ is the set of variable selected at stage $l-1$. ■

As shown in the Definition A.1, Zhang (2009) executes each forward selection stage based on ϵ. By constrast, we reparametrize forward selection stages (in Algorithm 1) and its stopping rule (in Algorithm 2) using different values of $\widehat{q}$. Given the assumption $\epsilon > \frac{1}{1-\mu_X(\overline{F})}\sigma\sqrt{2\ln(4p/\eta)}$, we can find an equivalent stopping criterion based on $\widehat{q}$ as follows,

- Assume that, on subsample $\left(Y^k, X^k\right)$, $\epsilon > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(4p/\eta)}$ stops the forward selection at stage l^k;
- based on line 6 of Algorithm 1, all the variables selected before stage l^k must have their $\widehat{q}_i^k$ values $> \left(\widetilde{p}+1-l^k\right)/\widetilde{p}$, where $\widetilde{p}$ is defined at line 2.
- hence, on k$^{\text{th}}$ subsample, the stopping rule

  $$\text{forward selection stops at stage } l^k$$

  is equivlanet to the stopping rule

  $$\text{the forward selection only selects the variables } \left\{\mathbf{x}_j : \widehat{q}_j^k > \left(\widetilde{p}+1-l^k\right)/\widetilde{p}\right\}.$$

  where $\widetilde{p} = \min\{n(K-1)/K, p\}$

Zhang (2009) also assume that, when the forward selection stops at stage l^k on subsample $\left(Y^k, X^k\right)$,

$$\epsilon > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(4p/\eta)},\ \exists\eta \in (0,1).$$

Such assumption plays a key role in theoretical analysis for the true signal recovery of forward selection or OMP. Since solar does not visualize ϵ, we need to reparametrize such assumption before applying the same analysis method to solar algorithm. Specifically, let's denote

$$\omega^{(l^k)} = \left|\left(\mathbf{x}^{(l^k)}\right)^T u^{(l^k)}\right|, \tag{A.3}$$

where $u^{(l^k)}$ is the regression residual of stage l^k and $\mathbf{x}^{(l^k)}$ is the variable selected at stage l^k. Solar selects variable on the L_0 and average L_0 paths based on $\widehat{q}^k$ and $\widehat{q}$. Hence, we can analysis the variable selection decision of solar by examining the absolute comovement between $u^{(l^k)}$ and $\mathbf{x}^{(l^k)}$ on the L_0 path, which is identical to the ϵ assumption. This can be shown as follows.

- Assume ϵ stopping rule stops forward selection at stage l^k and $\epsilon > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(2p/\eta)}$. We must have
$$\omega^{(l^k-1)} \geqslant \epsilon > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(2p/\eta)} \tag{A.4}$$
Hence, stopping forward selection based on $\epsilon > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(2p/\eta)}$ implies
$$\omega^{(l^k-1)} \geqslant \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(2p/\eta)}.$$
- If $\omega^{(\cdot)} \geqslant \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(2p/\eta)}$ is not violated until stage l^k and $\epsilon > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(2p/\eta)}$, it implies that $\omega^{(l^k)} < \epsilon$, triggering the ϵ stopping rule at stage l^k. Since forward selection does not stop at stage $l^k - 1$, we still have $\omega^{(l^k-1)} \geqslant \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(2p/\eta)}$

Hence, "the $\epsilon > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(2p/\eta)}$ stopping rule stops forward selection at stage l^k" is equivalent to "the last variable that forward selection selects before stop has $\omega > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(2p/\eta)}$". As such, we can reparametrize the stopping rule on $\left(Y^k, X^k\right)$ based on $\widehat{q}_i^k$ as follows

Definition A.2. (the assumptions on ϵ and $\widetilde{q}^k$ stopping rules)

- We refer to the Zhang (2009) assumption on ϵ stopping rule as *the ϵ stopping assumption*
$$\begin{aligned} &\text{when forward selection stops at stage } l^k \text{ on subsample } \left(Y^k, X^k\right), \\ &\exists \eta \in (0,1) \text{ such that } \epsilon > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(4p/\eta)}. \end{aligned} \tag{A.5}$$
- we refer to the following equivalent assumption as *the $\widetilde{q}^k$ stopping assumption* for solar:
$$\begin{aligned} &\text{when forward selection only select } \left\{\mathbf{x}_j : \widehat{q}_j^k > \frac{\widetilde{p}+1-l^k}{\widetilde{p}}\right\} \text{ on subsample } \left(Y^k, X^k\right), \\ &\exists \eta \in (0,1) \text{ such that } \omega^{(l^k)} = \left|\left(\mathbf{x}^{(l^k)}\right)^T u^{(l^k)}\right| > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(4p/\eta)}. \end{aligned} \tag{A.6}$$

Based on Definition A.2, we can reparametrize Lemma 1 into Lemma 2.

Lemma A.1. *Consider the forward selection algorithm on the kth subsample $\left(Y^k, X^k\right)$ with Assumption 1 satisfied. With probability larger than $1-\eta$, if eqn (A.6) is satisfied and*
$$\min_{j\in\overline{F}} \left|\overline{\beta}_j\right| \geqslant \frac{3\omega}{\rho_X\left(\overline{F}\right)\cdot\sqrt{n(K-1)/K}},$$
then
$$\overline{F} = \left\{\mathbf{x}_j : \widehat{q}_i^k > \widetilde{q}^k\right\}.$$
where $\widetilde{q}^k = \left(\widetilde{p}+1-l^k\right)/\widetilde{p}$.

Proof. Lemma A.1 is a straightforward result when replacing the ϵ stopping rule with $\widehat{q}^k$ stopping rule. Note that we replace n in eqn (A.2) with $n(K-1)/K$ since each subsample randomly drops $1/K$ of original sample points at Algorithm 1. □

Step 2 : averaging performance

Since we assume the nonzero $\overline{\beta}_i$ are the first p_1 components of $\overline{\beta}$, Lemma A.1 can be rewritten as

Lemma A.2. *Consider the forward selection algorithm on the kth subsample* $\left(Y^k, X^k\right)$ *with Assumption 1 satisfied. With probability less than* η*, if eqn (A.6) is satisfied and*

$$\min_{j\in\overline{F}} \left|\overline{\beta}_j\right| \geqslant \frac{3\omega}{\rho_X\left(\overline{F}\right)\cdot\sqrt{n\left(K-1\right)/K}},$$

then

$$\begin{cases} \widehat{q}_j^k \leqslant \widetilde{q}^k, \ \forall j \leqslant p_1 \\ \widehat{q}_j^k > \widetilde{q}^k, \ \forall j > p_1. \end{cases}$$

where $\widetilde{q}^k = \left(\widetilde{p}+1-l^k\right)/\widetilde{p}$.

To accomodate the multiple subsamples in average L_0 path, we modify the $\widehat{q}$ stopping rule as follows

Definition A.3. (the assumption for the $\widetilde{q}$ stopping rule) : we refer to the following rule as *the* $\widetilde{q}$ *stopping assumption* for average L_0 path:

$$\text{when forward selection only select } \left\{\mathbf{x}_j : \widehat{q}_j^k > \left(\widetilde{p}+1-l^k\right)/\widetilde{p}\right\} \text{ on subsample } \left(Y^k, X^k\right),$$
$$\exists \eta \in (0, 1/K) \text{ such that } \omega = \left|\left(\mathbf{x}^{(l^k-1)}\right)^T u^{(l^k-1)}\right| > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln\left(\frac{4p}{K\eta}\right)}. \quad \text{(A.7)}$$

Based on the $\widetilde{q}$ stopping rule, we can bound the probability and generate the Lemma A.3.

Lemma A.3. *Consider the forward selection algorithm on the average* L_0 *path with Assumption 1 satisfied. With probability less than* η*, if eqn (A.7) is satisfied and*

$$\min_{j\in\overline{F}} \left|\overline{\beta}_j\right| \geqslant \frac{3\omega}{\rho_X\left(\overline{F}\right)\cdot\sqrt{n\left(K-1\right)/K}},$$

then

$$\begin{cases} \frac{1}{K}\sum \widehat{q}_i^k = \widehat{q}_i > \widetilde{q}^*, \forall i \leqslant p_1 \\ \frac{1}{K}\sum \widehat{q}_i^k = \widehat{q}_i \leqslant \widetilde{q}^*, \forall i > p_1 \end{cases}$$

where $\widetilde{q}^* = \frac{1}{n}\sum_k^K \widetilde{q}^k/K$ *and* $\widetilde{q}^k = \left(\widetilde{p}+1-l^k\right)/\widetilde{p}$.

Proof. The proof is a direct result from Lemma 2. If we apply the $\widetilde{q}$ stopping rule, Lemma 2 implies that

$$Pr\left\{\widehat{q}_i^k \leqslant \widetilde{q}^k, \forall i \leqslant p_1 \text{ and } \widehat{q}_j^k > \widetilde{q}^k, \forall j > p_1\right\} \leqslant \eta/K \quad \text{(A.8)}$$

Since, for mutiple event A_i, $Pr\{\cap_i A_i\} \leqslant \sum_i Pr\{A_i\}$, we have

$$Pr\left\{\sum_{k=1}^{K}\widehat{q}_i^k \leqslant \sum_{k=1}^{K}\widetilde{q}^k, \forall i \leqslant p_1 \text{ and } \sum_{k=1}^{K}\widehat{q}_j^k > \sum_{k=1}^{K}\widetilde{q}^k, \forall j > p_1\right\} \leqslant \eta. \tag{A.9}$$

Since, $\widehat{q}_i = \frac{1}{K}\sum \widehat{q}_i^k$ and $\widetilde{q}^* = \frac{1}{K}\sum \widetilde{q}^k$, we have

$$Pr\{\widehat{q}_i \leqslant \widetilde{q}^*, \forall i \leqslant p_1 \text{ and } \widehat{q}_j > \widetilde{q}^*, \forall j > p_1\} \leqslant \eta. \tag{A.10}$$

□

Step 3 : variable selection consistency

Based on Lemma A.3, we can obtain Theorem A.2 for variable selection consistency as a straighforward result.

Theorem A.2. *Consider the forward selection algorithm on the average L_0 path with Assumption 1 satisfied, noise σ independent of n. Assume that the strong irrepresentable condition holds. For each problem of sample size n, denote $F(n)$ as the index set of selected variables when forward selection stops with $\omega \geqslant n^{s/2}$, $\forall s \in (0,1]$, and $\overline{F}(n)$ as the correpsonding index set of informative variables. We have*

$$Pr\left(\ F(n) \neq \overline{F}(n)\ \right) \leqslant \exp\left(-\frac{n^s}{\log(n)}\right)$$

if

$$p(n) \leqslant \exp\left(\frac{n^s}{\log(n)}\right),$$

and

$$\min_{j\in\overline{F}}|\overline{\beta}_j| \geqslant \frac{3n^{(s-1)/2}}{\rho_X\left(\overline{F}(n)\right)}$$

where $p(n)$ is the total dimension of variable as n increases.

Proof. When n is sufficiently large, the assumption

$$\omega^{(l^k)} = \left|\left(\mathbf{x}^{(l^k)}\right)^T u^{(l^k)}\right| > \frac{1}{1-\mu_X(F)}\sigma\sqrt{2\ln(4p/\eta)}$$

and

$$\min_{j\in\overline{F}}|\overline{\beta}_j| \geqslant \frac{3\omega}{\rho_X\left(\overline{F}\right)\cdot\sqrt{n(K-1)/K}}$$

hold with $\eta = \exp(-n^s/\log(n))$. Thus, Theorem A.2 is a direct result from Lemma A.2 and A.3. □