On the use of information criteria for subset selection in least squares regressions

Sen Tian
and
Clifford M. Hurvich
and
Jeffrey S. Simonoff
Department of Technology, Operations, and Statistics,
Stern School of Business, New York University

Abstract

Best subset selection (BS) is a popular least squares (LS) based subset selection method for regression modeling. In order to select the optimal subset size k, one often applies an information criterion such as the Mallows' C_p. Ye (1998) and Efron (2004) demonstrated that with the effective degrees of freedom (edf) being plugged in place of the subset size, C_p-edf provides an unbiased estimate of the prediction error. This paper is largely motivated by the following challenges of applying BS in practice: (1) BS is NP-hard and its computational cost grows exponentially with the problem size; and (2) the edf for BS generally does not have an analytical expression. In this paper, we start from a restricted case (orthogonal design matrix X). We build a connection between BS and its Lagrangian version, and propose a heuristic degrees of freedom (hdf) for BS, which can be estimated via an analytically-based expression. Furthermore, we introduce AICc-edf and its feasible version AICc-hdf, which are motivated by trying to construct an unbiased estimator of the Kullback-Leibler divergence for BS. Finally, we return to a general X and propose a novel LS-based method, the best orthogonalized subset selection (BOSS) method. BOSS along with AICc-hdf works well in both simulations and real data analysis, with the computational effort of a single ordinary LS fit.

Keywords: Best subset regression, effective degrees of freedom, information criteria.

1 Introduction

Suppose that we have the following data generating process

$$\mathbf{y} = \mu + \epsilon, \tag{1}$$

where $\mathbf{y} \in \mathcal{R}^n$ is the response vector, $\mu \in \mathcal{R}^n$, is the fixed mean vector, and $\epsilon \in \mathcal{R}^n$ is the noise vector. The mean vector is estimated based on a fixed design matrix $\mathbf{X} \in \mathcal{R}^{n \times p}$. We assume the error $\epsilon \overset{iid}{\sim} \mathcal{N}(0, \sigma^2 I)$ and $n > p$.

1.1 Best subset selection

Best subset selection (BS) (Hocking and Leslie, 1967) seeks the set of predictors that best fit the data in terms of quadratic error for each given subset size (excluding the intercept) $k \in \{0, 1, \cdots, p\}$, i.e. it solves the following optimization problem:

$$\text{(constrained formulation)} \qquad \min_{\beta_0, \beta} \frac{1}{2}\|\mathbf{y} - \beta_0 \mathtt{1} - \mathbf{X}\beta\|_2^2 \quad \text{subject to} \quad \|\beta\|_0 \leq k, \tag{2}$$

where $\|\beta\|_0 = \sum_{i=1}^p \mathtt{1}(\beta_i \neq 0)$ is the number of non-zero coefficients in β. Note that to simplify the discussion, we assume that the intercept term $\beta_0 = 0$ throughout the paper, except in the real data examples, where all of the fitting procedures include an intercept.

BS is known to be an NP-hard problem (Natarajan, 1995) and its computational cost grows exponentially with the dimension p. Many attempts have been made to reduce the computational cost of the method. The most well-known one is the branch-and-bound algorithm 'leaps' (Furnival and Wilson, 1974) that solves (2) in seconds for p being up to around 30. More recently, Bertsimas et al. (2016) formulated (2) using a mixed integer operator (MIO), and largely reduced the computing overhead by using a well-developed optimization solver such as `GUROBI` or `CPLEX`. However, according to Hastie et al. (2017), MIO normally takes about 3 minutes to find the solution at a given size k for a problem with $n = 500$ and

$p = 100$. Considering the size of data that we are commonly dealt with nowadays, it's still not feasible in many situations, and solving (2) remains being a challenge for most real world applications.

In order to select the optimal tuning parameter, e.g. the subset size k in (2), one often applies an information criterion, which augments the training error with the effective degrees of freedom (edf). Efron (1986) defined the edf for a general fitting rule $\hat{\mu}$ as:

$$\text{edf}(\hat{\mu}) = \frac{1}{\sigma^2} \sum_{i=1}^{n} \text{cov}(\hat{\mu}_i, y_i). \tag{3}$$

It is easy to verify that edf for the linear regression of y upon X is the number of estimated coefficients p. However, Janson et al. (2015) showed in simulations that there can be a large discrepancy between the edf of BS at size k and itself. Similar evidence can be found in Tibshirani (2015), where the author quantifies the difference as the search degrees of freedom, which accommodates the amount of searching that BS performs in order to choose a best k-predictor subset. Unfortunately, edf of BS does not have an analytical expression except when X is orthogonal and $\mu = 0$ (Ye, 1998). Numerically, we can apply tools like data perturbation (Ye, 1998), bootstrap (Efron, 2004) or data randomization (Harris, 2016) to estimate edf, but all rely on tunings of some hyperparameters and can be computationally intensive.

This paper is motivated by the above challenges. We propose a heuristic degrees of freedom (hdf) for BS by assuming an orthogonal X. We further introduce a novel least squares (LS) based subset selection method, the best orthogonalized subset selection (BOSS), and we demonstrate that BOSS using an information criterion with hdf works well in practice for a general X with the computational cost of a single ordinary LS fit.

1.2 Optimism theorem and information criteria for BS

Information criteria are designed to provide an unbiased estimate of the testing error, and can be derived from the so-called optimism theorem. Denote Θ as an error measure, err as the

training error, Err as the testing error, y^0 as a new response vector with the same distribution but independent of the original y, and E_0 is the expectation taken over y^0. Efron (1986) defined the optimism as

$$\text{op} = \text{Err} - \text{err},$$

and introduced the optimism theorem,

$$E(\text{op}) = E(\text{Err}) - E(\text{err}).$$

A straightforward result from the optimism theorem is that

$$\widehat{\text{Err}} = \text{err} + E(\text{op})$$

is an unbiased estimator of $E(\text{Err})$, and is intended to balance the trade-off between model fit and model complexity. The challenge is to find $E(\text{op})$ for a given fitting rule $\hat{\mu}$ and error measure $\Theta(y, \hat{\mu})$.

When the error measure Θ is the squared error (SE), i.e. $\Theta(y_i, \hat{\mu}_i) = (y_i - \hat{\mu}_i)^2$, err_{SE} (denoted as the training error when Θ is SE) then becomes the residual sum of squares $\text{RSS} = \sum_{i=1}^n (y_i - \hat{\mu}_i)^2$, and the testing error $\text{Err}_{\text{SE}} = \sum_{i=1}^n E_0[\Theta(y_i^0, \hat{\mu}_i)]$. Ye (1998) and Efron (2004) proved that for a general fitting rule $\hat{\mu}$ such as BS, $E(\text{op}_{\text{SE}}) = 2\sigma^2 \cdot \text{edf}(\hat{\mu})$, and proposed

$$\text{C}_p\text{-edf} = \text{RSS} + 2\sigma^2 \cdot \text{edf} \tag{4}$$

as an information criterion. These authors also showed that the traditional C_p,

$$\text{C}_p\text{-ndf} = \text{RSS} + 2\sigma^2 \cdot \text{ndf},$$

can be greatly biased when applied for BS, since C_p-ndf (Mallows, 1973) was derived for a linear estimation rule $\hat{\mu} = Hy$ where H is independent of y, which is not the case for BS. Here ndf denotes the naive degrees of freedom, i.e. $\text{Tr}(H)$. A further major issue regarding applying C_p in practice is that it requires an estimate of σ^2.

Another commonly used error measure is the deviance (up to a constant)

$$\Theta = -2 \log f(y|\mu, \sigma^2),$$

where f is a pre-specified parametric model. Let $\hat{\mu}$ and $\hat{\sigma}^2$ be the maximum likelihood estimators obtained by maximizing $f(y|\mu, \sigma^2)$. We then have $\text{err}_{\text{KL}} = -2 \log f(y|\hat{\mu}, \hat{\sigma}^2)$ and $\text{Err}_{\text{KL}} = -2E_0 \left[\log f(y^0|\hat{\mu}, \hat{\sigma}^2)\right]$, where the latter is the Kullback-Leibler (KL) discrepancy. For a linear estimation procedure, assuming asymptotic normality (f not necessarily Gaussian) and the true model distribution is contained in the specified parametric model f, Konishi and Kitagawa (2008) proved that $E(\text{op}_{\text{KL}}) = 2 \cdot \text{ndf} + o(1)$, and AIC (Akaike, 1973),

$$-2 \log f(y|\hat{\mu}, \hat{\sigma}^2) + 2 \cdot \text{ndf},$$

is asymptotically $\widehat{\text{Err}}_{\text{KL}}$. If f follows a Gaussian distribution, as assumed in (1), AIC can be expressed as

$$\text{AIC-ndf} = n \log \left(\frac{\text{RSS}}{n} \right) + 2 \cdot \text{ndf}.$$

Hurvich and Tsai (1989) replaced the asymptotic $E(\text{op}_{\text{KL}})$ with its exact value, for Gaussian linear regression with an assumption that the predictors with non-zero true coefficients are included in the model, and used the corrected AIC

$$\text{AICc-ndf} = n \log \left(\frac{\text{RSS}}{n} \right) + n \frac{n + \text{ndf}}{n - \text{ndf} - 2}.$$

Neither AIC nor AICc has a penalty term depending upon σ^2, a clear advantage over C_p.

It remains a challenge to derive a KL-based information criterion for BS. Liao et al. (2018) estimated $E(\text{op}_{\text{KL}})$ via Monte Carlo simulations, but this relies on thousands of fits of the procedure, which is not computationally feasible for large datasets.

In this work, we propose AICc-edf

$$\text{AICc-edf} = n \log \left(\frac{\text{RSS}}{n} \right) + n \frac{n + \text{edf}}{n - \text{edf} - 2} \tag{5}$$

for this purpose. We demonstrate that (5) approximates the unbiased estimator $\widehat{\text{Err}}_{\text{KL}}$ well, and they both generally choose the same subset when used as selection rules. Furthermore, its feasible implementation AICc-hdf works reasonably well as a selection rule for BS with an orthogonal X and for our proposed method BOSS in general.

1.3 Structure of this paper

The rest of the paper is organized as follows. In section 2, by assuming an orthogonal X, we introduce the hdf for BS. We provide a theoretical justification in a restricted scenario, and numerical justifications in general. We provide numerical evidence that AICc-edf approximates $\widehat{\text{Err}}_{\text{KL}}$ well, and its feasible version AICc-hdf works well as a selection rule for BS. In Section 3, we consider a general X and propose the method BOSS. The performance of BOSS using AICc-hdf as the selection rule is studied in simulations and is compared with that of regularization methods. Real data examples are provided in section 4, and discussion is provided in section 5.

We make the code to reproduce the results in this paper as well as a R package **BOSS**[1] publicly available at `https://github.com/sentian/boss`.

2 AICc-hdf for BS with orthogonal X

2.1 A heuristic degrees of freedom for BS

The edf of BS has an analytical expression only when the true model is $\mu = 0$ (Ye, 1998). Tibshirani (2015) studied the Lagrangian formulation of BS (LBS) and provided an analytical expression for edf without any restrictions on μ. To distinguish between the two methods, we use $\text{df}_C(k)$ and $\text{df}_L(\lambda)$ to denote edf of BS at size k and edf of LBS at tuning parameter λ,

[1] At the time of submission of this paper, we have submitted the R package to *CRAN*.

respectively. In this section, we introduce a heuristic degrees of freedom (hdf) for BS that is built upon the connection between $\mathrm{df}_C(k)$ and $\mathrm{df}_L(\lambda)$.

2.1.1 Lagrangian BS and its edf

At each regularization parameter $\lambda \geq 0$, LBS solves

$$\min_{\beta} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_0. \tag{6}$$

Both LBS (6) and BS (2) are LS regressions of y upon a certain subset of X. With orthogonal X, both problems have analytical solutions: $\hat{\beta}_i(\lambda) = z_i \mathbb{1}_{(|z_i| \geq \sqrt{2\lambda})}$ for (6) and $\hat{\beta}_i(k) = z_i \mathbb{1}_{(|z_i| \geq |z_{(k)}|)}$ for (2), where $z = X^T y$ and $z_{(k)}$ is the k-th largest coefficient in absolute values. These two problems are not equivalent, and there is no clear one-to-one correspondence between λ in (6) and k in (2). Indeed, for each λ there exists a k such that $\hat{\beta}(\lambda) = \hat{\beta}(k)$ where $\hat{\beta}(\lambda)$ is the solution of (6) at λ and $\hat{\beta}(k)$ is the solution of (2) at k, but the reverse does not necessarily hold, since there will be multiple λ corresponding to the same solution $\hat{\beta}(k)$. Moreover, with a general X, solving (6) does not guarantee recovery of the entire solution path given by solving (2) at $k = 0, \cdots, p$.

By assuming an orthogonal X, Tibshirani (2015) derived an expression for $\mathrm{df}_L(\lambda)$ based on definition (3),

$$\mathrm{df}_L(\lambda) = E(k_L(\lambda)) + \frac{\sqrt{2\lambda}}{\sigma} \sum_{i=1}^{p} \left[\phi\left(\frac{\sqrt{2\lambda} - (X^T\mu)_i}{\sigma} \right) + \phi\left(\frac{-\sqrt{2\lambda} - (X^T\mu)_i}{\sigma} \right) \right], \tag{7}$$

where the expected subset size is given as

$$E(k_L(\lambda)) = \sum_{i=1}^{p} \left[1 - \Phi\left(\frac{\sqrt{2\lambda} - (X^T\mu)_i}{\sigma} \right) + \Phi\left(\frac{-\sqrt{2\lambda} - (X^T\mu)_i}{\sigma} \right) \right]. \tag{8}$$

2.1.2 hdf for BS

Given the similarity of problems (2) and (6), we would like to approximate $\mathrm{df}_C(k)$ with $\mathrm{df}_L(\lambda)$. One implementation of this could be proceeded as follows. Note that $\mathrm{df}_C(k)$ is a

discrete function of $k = 0, \cdots, p$ while $\mathrm{df}_L(\lambda)$ is a continuous function of a real variable $\lambda \geq 0$. We propose an hdf that uses $\mathrm{df}_L(\lambda)$ for a particular value of λ depending on k as a proxy for $\mathrm{df}_C(k)$. Based on (8), λ and $E(k_L(\lambda))$ have a clear one-to-one correspondence, which implies that we can find a unique $\lambda_k^\star$ such that $E(k_L(\lambda_k^\star)) = k$ for each $k = 0, \cdots, p$. The value of hdf is $\mathrm{df}_L(\lambda_k^\star)$ obtained by substituting $\lambda_k^\star$ into (7). The implementation process is summarized in Algorithm 1. Note that hdf requires estimates of μ and σ, and we use the estimates from the LS regression on all predictors.

Algorithm 1 The heuristic df (hdf) for BS at size k

Input: X (orthogonal), σ and μ. For a given subset size k,

1. Based on (8), calculate λ such that $E(k_L(\lambda)) = k$, and call it $\lambda_k^\star$.
2. Based on (7), calculate $df(\lambda_k^\star)$, and call it hdf(k).

Repeat the above steps for all $k = 0, \cdots, p$, yielding hdf at each subset size k.
In place of μ and σ in (7) and (8), we use OLS estimates based on the full model, i.e. $\hat{\mu} = XX^T y$, $\hat{\sigma}^2 = \|y - \hat{\mu}\|_2^2/(n-p)$.

2.1.3 Theoretical justification of hdf under a null true model

Assume $\mu = 0$, with X still being orthogonal. In such a restricted scenario, $\mathrm{df}_C(k)$ has an analytical expression, which allows us to provide some theoretical justification for hdf(k). We start by introducing some notations, and present the main result in Theorem 1 and its Corollary. The detailed proofs are given in Appendix A.

Denote $\tilde{X}_{(i)}$ as the i-th largest order statistic in an i.i.d sample of size p from a χ_1^2 distribution. Ye (1998) showed that

$$\mathrm{df}_C(k) = E\left(\sum_{i=1}^{k} \tilde{X}_{(i)}\right).$$

Let $\tilde{H}(s) = -\tilde{Q}(1-s)$ where $\tilde{Q}$ is the quantile function of a χ^2_1 distribution, and $s \in (0,1)$. For $0 \le s \le t \le 1$, the truncated variance function is defined as

$$\tilde{\sigma}^2(s,t) = \int_s^t \int_s^t (u \wedge v - uv) d\tilde{H}(u) d\tilde{H}(v),$$

where $u \wedge v = \min(u,v)$. Denote $\tilde{Y}_p = \tilde{\sigma}_p^{-1}(\sum_{i=1}^k \tilde{X}_{(i)} - \tilde{\mu}_p)$, where

$$\tilde{\sigma}_p = \sqrt{p} \cdot \tilde{\sigma}(1/p, k/p),$$

and

$$\tilde{\mu}_p = -p \int_{1/p}^{k/p} \tilde{H}(u) du - \tilde{H}(\frac{1}{p}).$$

Theorem 1. *Assume X is orthogonal and the true model is null ($\mu = 0$). As $p \to \infty$, $k \to \infty$ with $k = \lfloor px \rfloor$, we have*

$$\frac{1}{2p} hdf(k) = \frac{1}{2p} df_C(k) - \frac{\tilde{\sigma}_p}{2p} E(\tilde{Y}_p) + O\left(\frac{\log(p)}{p}\right), \tag{9}$$

where $x \in (0,1)$ is a constant and $\lfloor \cdot \rfloor$ denotes the greatest integer function.

Corollary 1.1. *If $\limsup |E(\tilde{Y}_p)| < \infty$, we further have*

$$\frac{df_C(k)}{hdf(k)} \to 1. \tag{10}$$

Remark: If $\tilde{Y}_p$ is uniformly integrable, then $E(\tilde{Y}_p) \to 0$, and hence the result of Corollary 1.1 holds.

It can be seen that Corollary 1.1 holds given the assumptions, since both $\mathrm{hdf}(k)$ and $\mathrm{df}_C(k)$ become infinite while $E(\tilde{Y}_p)$ and the remainder term remain bounded. The Corollary suggests that for large k and large p, the ratio of $\mathrm{df}_C(k)$ over $\mathrm{hdf}(k)$ shall be close to 1. We next explore empirically the relative behavior of the two dfs for a fixed p with an increasing k.

2.1.4 Numerical justification of hdf

Figure 1 shows the comparison via simulations. We fit BS on 1000 realizations of the response generated by fixing X. The edf is calculated based on definition 3 using the sample covariances, while hdf is an average over 1000 replications. We see that in the null case, using hdf to approximate edf becomes better as k approaches to p, providing a finite-sample justification of Corollary 1.1.

Besides the null model, we consider a sparse model (Orth-Sparse-Ex1) with $p_0 = 6$ true predictors (those with non-zero coefficients), and a dense model (Orth-Dense) where all predictors have non-zero coefficients. We also consider two signal-to-noise (SNR) ratios with 'hsnr' and 'lsnr' representing high and low SNR respectively, and the SNR is defined as $\text{Var}(x^T\beta)/\sigma^2$. The details of the setups for Orth-Sparse-Ex1 and Orth-Dense models can be found in section 2.3.1. Similar to the null case, we see that hdf approaches edf as k gets close to p, i.e. the statement of Corollary 1.1 holds in these scenarios as well. An exception may be the dense model with high SNR where the two dfs do not agree, but we can still see the gap between them becomes smaller as k approaches p. Furthermore, we see that hdf generally approximates edf well, where the difference is more pronounced when BS underfits, e.g. a spare true model with $k < p_0 = 6$.

2.2 A KL-based IC for BS

2.2.1 AICc-hdf as a feasible implementation of AICc-edf

We have shown that hdf generally approximates edf well, and it agrees with edf as k approaches p. By replacing edf with hdf in (5), we have a feasible selection rule AICc-hdf. Figure 2 compares AICc-edf with AICc-hdf. Similarly to the comparison of the degrees of freedom values, we see AICc-hdf agrees with AICc-edf as k approaches p. Even at the places where we see differences between the degrees of freedom values, e.g. a sparse true model

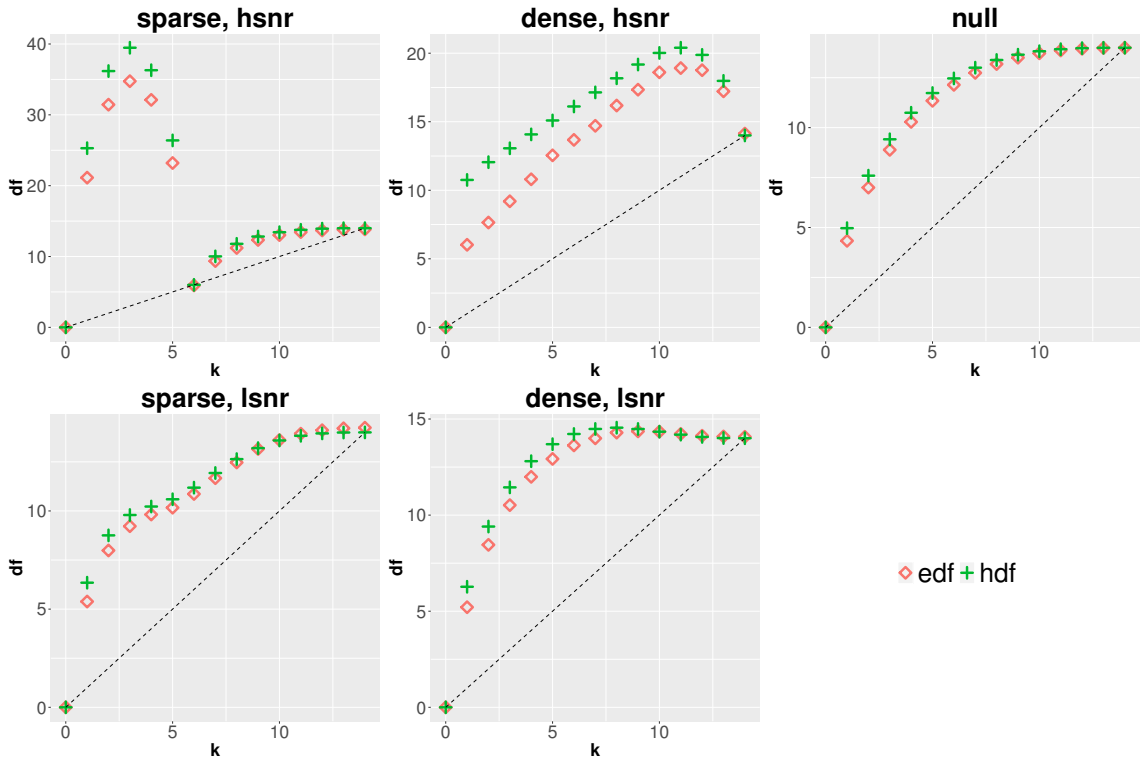

Figure 1: hdf(k) vs df$_C(k)$ (edf of constrained BS). The black dashed line is the 45-degree line. Here X is orthogonal with $n = 200$ and $p = 14$. Three types of the true model and two SNR are considered. We assume the knowledge of μ and σ in calculating the dfs.

with high SNR and $k < p_0 = 6$, the differences are compensated by the model fit and we see AICc-hdf is very close to AICc-edf.

2.2.2 Numerical justification of AICc-edf

AICc-edf is motivated by trying to construct an unbiased estimator of the prediction error $E(\text{Err}_{\text{KL}})$. The expected KL-based optimism for BS is given as

$$E(\text{op}_{\text{KL}}) = E\left(n\frac{n\sigma^2 + \|\mu - X\hat{\beta}(k)\|_2^2}{\|y - X\hat{\beta}(k)\|_2^2}\right). \tag{11}$$

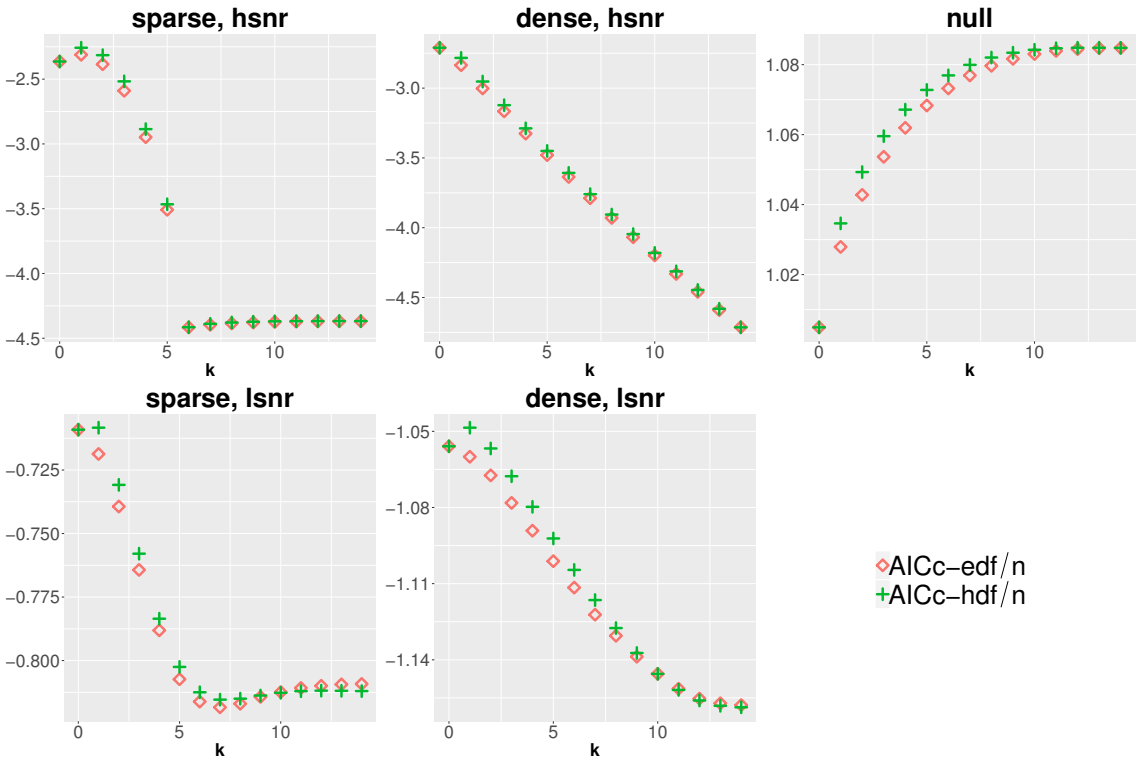

Figure 2: AICc-edf vs AICc-hdf. The model fit is an average over 1000 replications. The setups of the true models are the same as in Figure 1.

The derivation is presented in Appendix B. Note that (11) holds for a general X. Augmenting $E(\text{op}_{\text{KL}})$ with the training error err_{KL}, we have an unbiased estimator of the prediction error $\widehat{\text{Err}}_{\text{KL}}$.

Figure 3 compares AICc-edf and its feasible implementation AICc-hdf with $\widehat{\text{Err}}_{\text{KL}}$. We can see that AICc-edf generally tracks $\widehat{\text{Err}}_{\text{KL}}$ reasonably well. In fact, they agree with each other in the null case and a sparse true model with high SNR. Noticeable discrepancies can be observed in a sparse true model with high SNR and $k < p_0 = 6$. This is the place where the set of true predictors is not entirely included in the model. The derivations of the classic AIC-ndf and AICc-ndf are based on an assumption that the true predictors are included in the model. In the place where such assumption is violated, AICc-ndf will no longer be guaranteed

to be unbiased, and the conjecture can be made here for AICc-edf. More importantly, both AICc-edf and AICc-hdf yield the same average optimal size $k^\star$ as $\widehat{\text{Err}}_{\text{KL}}$ across all scenarios, when they are applied as selection rules.

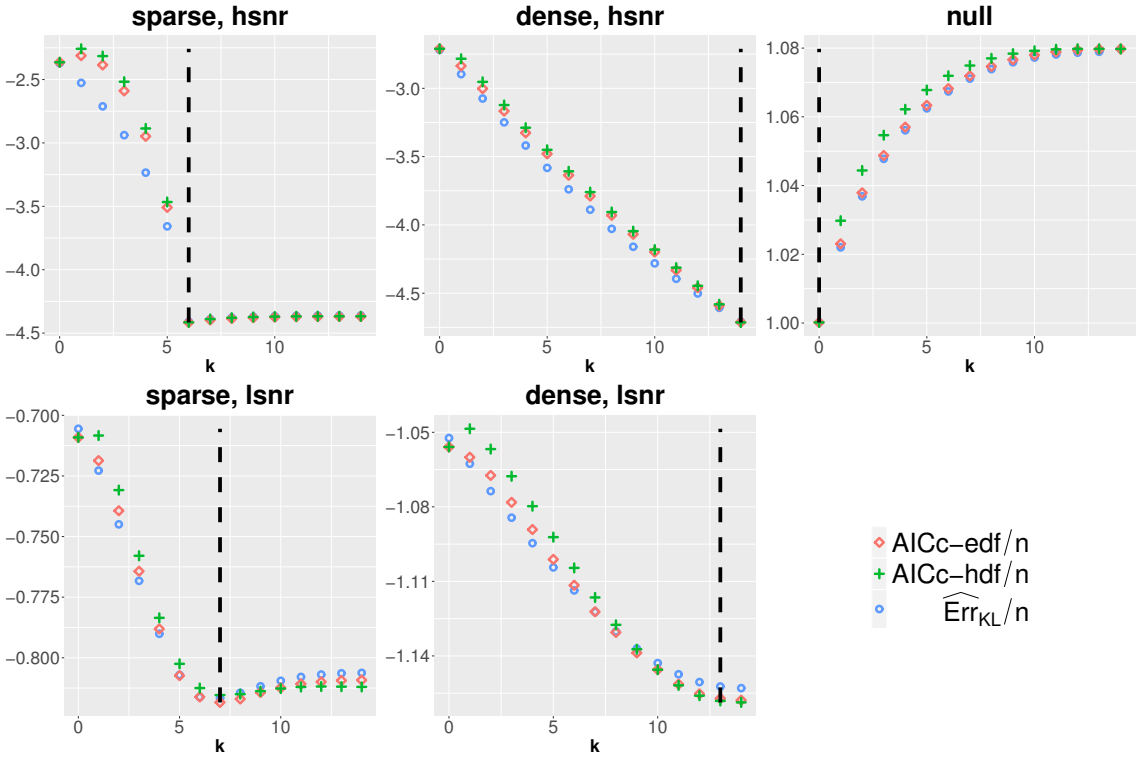

Figure 3: AICc-edf, AICc-hdf and $\widehat{\text{Err}}_{\text{KL}}$. The optimal subset size $k^\star$ is selected by minimizing a criterion. All three criteria in the figure lead to the same average of $k^\star$ over 1000 replications, as denoted by the black dashed vertical lines. The setups of the true models are the same as in Figure 1.

2.3 The performance of AICc-hdf as a selection rule for BS

We now study the performance of AICc-hdf as a selection rule for BS. The only assumption we make in the simulations is X being orthogonal. Both μ and σ are unknown, as would be the case in practice. We start by showing that AICc-hdf can perform well for BS comparing

with other selection rules. We then compare the performance of BS-AICc-hdf to that of regularization methods.

2.3.1 Simulation set-up

We consider a trigonometric configuration of X that is studied by Hurvich and Tsai (1991), where $X = (x^1, x^2)$ is an n by p matrix with components defined by

$$x^1_{tj} = \sin\left(\frac{2\pi j}{n}t\right),$$

and

$$x^2_{tj} = \cos\left(\frac{2\pi j}{n}t\right),$$

for $j = 1, \cdots, p/2$ and $t = 0, \cdots, n-1$. The columns of X are then standardized to have l_2 norm 1, to make them orthonormal. By fixing X, the responses are generated by (1), where $\mu = X\beta$. The error ϵ is also shifted to have mean 0, hence the intercept will be zero.

We consider the following configurations of the experiment:

- Sample size: $n \in \{200, 2000\}$.
- Number of predictors: $p \in \{14, 30, 60, 180\}$.
- Signal-to-noise ratio: SNR $\in \{0.2, 1.5, 7\}$ (low, medium and high). The average oracle R^2 (linear regression on the set of true predictors) corresponding to these three SNR values are around 20%, 50% and 90%, respectively.
- Coefficient vector β (Orth in the following denotes for orthogonal X):
  - Orth-Sparse-Ex1: $\beta = [1_6, 0_{p-6}]^T$
  - Orth-Sparse-Ex2: $\beta = [1, -1, 5, -5, 10, -10, 0_{p-6}]^T$
  - Orth-Dense (Taddy, 2017): $\beta_j = (-1)^j \exp(-\frac{j}{\kappa})$, $j = 1, \cdots, p$. $\kappa = 10$

In total, there are 72 different scenarios in the experiment. The full set of simulation results is presented in the supplemental material. In each scenario, 1000 replications of the response y are generated. A fitting procedure $\hat{\mu}$, is evaluated via the average RMSE, where

$$\text{RMSE}(\hat{\mu}) = \sqrt{\frac{1}{n}\|\hat{\mu} - X\beta\|_2^2}. \tag{12}$$

To make the scales easier to compare, we construct two relative metrics: % worse than the best possible BS, and relative efficiency, which are defined as follows:

- **% worse than best possible BS**

$$= 100 \times \left(\frac{\text{averge RMSE of a fitting procedure } \hat{\mu}}{\text{average RMSE of the best possible BS}} - 1 \right) \%, \tag{13}$$

  where the best possible BS here means that on a single fit, choosing the optimal subset size $k^\star$ with the minimum RMSE among all $p+1$ candidates, as if an oracle tells us the true model.

- **Relative efficiency:** For a collection of fitting procedures, the relative efficiency for a particular procedure j, is defined as

$$\frac{\min_l \text{ average RMSE of fitting procedure } l}{\text{average RMSE of fitting procedure } j}. \tag{14}$$

  The relative efficiency is a measure between 0 and 1. Higher value indicates better performance. Besides the fitting procedures specified, we include the null and full OLS in the calculation of relative efficiency.

We also present the sparsistency (number of true positives) and number of extra predictors (number of false positives).

2.3.2 AICc-hdf and other selection rules for BS

hdf also makes C_p-hdf a feasible selection rule for BS. By analogy to C_p and AICc, we can also define BIC-edf as

$$\text{BIC-edf} = n \log\left(\frac{\text{RSS}}{n}\right) + \log(n) \cdot \text{edf}, \tag{15}$$

and its feasible version BIC-hdf, where the original BIC (or BIC-ndf in our notation) was introduced in Schwarz et al. (1978). We also consider a numerical estimation of edf that is based on the parametric bootstrap, and we denote it as bdf. The detailed implementation of bdf and the benefit of parametric bootstrap is discussed in Efron (2004). In our experiment, we use 100 bootstrapped samples. Besides the information criterion, we further include 10-fold cross-validation (CV) for comparison.

A selected set of results is shown in Table 1 and 2. A brief summary is as follows:

- Using information criteria in the naive way (with ndf) can be dangerous, especially when p is large and SNR is high. For example, it significantly overfits and can be almost 400 times worse in terms of RMSE than using hdf in AICc for $n = 200$, high SNR and $p = 180$ in Orth-Sparse-Ex1. Increasing the sample size n does not improve IC-ndf, and the overfiting persists.

- AICc-hdf generally does not lose much efficiency and performs similarly in terms of RMSE, in comparison to the infeasible AICc-edf. Increasing the sample size n or SNR improves the performance of both AICc-edf and AICc-hdf.

- AICc-hdf performs very similarly to AICc-bdf. Since bdf is calculated based on 100 bootstrapped samples, it is roughly 100 times more intensive than hdf in terms computation.

- AICc-hdf is generally better than 10-fold CV, except when n is small and SNR is low. The most noticeable difference in the efficiency between two rules is in Orth-Sparse-Ex1, $n = 200$, low SNR and $p = 30$, where AICc-hdf has relative efficiency 0.87 compared to 0.93 for CV. However, when n is large or SNR is high, the advantage of applying AICc-hdf can be significant. For example, in the scenarios other than $n = 200$ and low SNR for Orth-Sparse-Ex1, AICc-hdf has efficiency above 0.95, while it is only around 0.8 for CV. In fact, unlike AICc-hdf, CV is relatively insensitive to the change of SNR or

sample size. The improvement in performance by increasing n or SNR does not seem to apply to CV. Note that 10-fold CV is roughly 10 times heavier in terms of computation than AICc-hdf.

- C_p-edf performs similarly to AICc-edf. In contrast, when we consider the feasible implementations (ndf/hdf/bdf), i.e. when σ is estimated by full OLS, C_p can suffer when p is close to n, such as when $n = 200$ and $p = 180$. Under a sparse true model BIC-hdf performs slightly better than AICc-hdf except when SNR is low and $n = 200$, where BIC is considerably worse. Under a dense true model BIC-hdf is always outperformed by AICc-hdf.

For the reasons presented above, we conclude that AICc-hdf is the best feasible selection rule for BS, among all that have been considered.

2.3.3 How does BS perform compared to regularization methods?

We have seen that AICc-hdf can be an effective selection rule for BS. In this section, we compare BS with some popular regularization methods, including LASSO (Tibshirani, 1996), SparseNet (Mazumder et al., 2011), Gamma LASSO (Taddy, 2017) and relaxed LASSO (Meinshausen, 2007). We use `R` packages **glmnet** (Friedman et al., 2010), **sparsenet** (Mazumder et al., 2011), **gamlr** (Taddy, 2017) and **relaxo** (Meinshausen, 2012), to fit them, respectively, which are all available on *CRAN*.

As to the selection rule, we use AICc-hdf for BS, AICc-ndf for LASSO, and 10-fold CV for the rest. In addition to these selectors, we have also considered 10-fold CV for LASSO. We find (in the supplement) that 10-fold CV performs similarly to AICc-ndf for LASSO. In fact, the use of AICc for LASSO has been explored in Flynn et al. (2013), where the authors proved that AICc-ndf is asymptotically efficient while performing similarly to CV. We further notice (in the supplement) that SparseNet performs largely similarly to, but slightly better

than, relaxed LASSO and Gamma LASSO, and hence only the result for SparseNet will be presented here.

A selected set of results is presented in Table 3. A brief summary is as follows:

- With a relatively small sample size $n = 200$ and a sparse true model, BS performs the best when SNR is high, LASSO is best in low SNR, and SparseNet has performance in between of the other two methods. LASSO has the property of 'over-select and shrink', in order to retain less bias on the large non-zero estimates. In a high SNR, this property can result in disastrous performance, especially when p is close to n. For example, in Orth-Sparse-Ex1, high SNR and $p = 180$, the relative efficiency of LASSO is only 0.19 and it significantly overfits. However, this property can be beneficial when SNR is low, as a method like BS has higher chance to miss the true predictors (less sparsistency).

- With $n = 200$ and a dense true model, the methods perform similarly when SNR is high, while LASSO is better in low SNR.

- With a relatively large sample size $n = 2000$, BS becomes the best in almost all scenarios. The only exception is when the true model is dense and SNR is low, where BS is very close to the best. In fact, all three methods benifit from increasing n, since we can see larger sparsistency and fewer extra variables. Given that, it seems that BS profits the most according to the boost of its relative performance in low SNR.

Given the spirit of the summary above, it's important to point out the relevant work of Hastie et al. (2017), where the authors provide a comprehensive set of simulation comparisons on BS, LASSO and relaxed LASSO. The authors concluded that BS performs the best in high SNR, LASSO is the best in low SNR while relaxed LASSO is in between. This coincides with the results here when sample size is relatively small $n = 200$, given the similarity in the performance of relaxed LASSO and SparseNet. However, we find BS to be the best for large sample size n even when the SNR is low (note that Hastie et al. (2017) did not examine any

sample sizes greater than $n = 500$). Moreover, it should be noted that Hastie et al. (2017) focus on the best possible performance of each method by applying a separate validation set drawn from the true model, rather than on feasible selection, as is considered in this study.

2.4 A discussion on the use of information criteria in LBS

Since the edf of LBS has an analytical expression and LBS can recover the solution path of BS, one may ask why not just use LBS with a selection rule such as C_p-edf, which is well-defined for any general fitting procedure.

Recall that with an orthogonal X, the estimated coefficients for LBS (6) are $\hat{\beta}_i(\lambda) = z_i \mathbb{1}_{(|z_i| \geq \sqrt{2\lambda})}$, and they are $\hat{\beta}_i(k) = z_i \mathbb{1}_{(|z_i| \geq |z_{(k)}|)}$ for BS (2), where $z = X^T y$ and $z_{(k)}$ is the k-th largest coefficient in absolute value. Given a realization (X, y), we generate a sequence of λ where $\sqrt{2\lambda_i} = |z_{(i)}|$ $(i = 1, \cdots, p)$. Between each subsequent λ_i and λ_{i+1}, we add another 9 equally spaced values in the log scale.

Table 4 shows that LBS is outperformed by BS in almost all scenarios. We use C_p-edf as the selection rule for both methods, where edf of BS is estimated via simulations and edf of LBS is calculated using formulas (7) and (8). We see that (1) LBS always chooses more predictors than BS; (2) LBS deteriorates as p gets larger; and (3) increasing the sample size n does not help LBS. An explanation of why LBS overfits is as follows. We know that for at a subset size k, there're 10 different λ's by construction result in the same solution $\hat{\beta}(\lambda)$, and they lead to different edf and further different values of C_p-edf. Figure 4 presents the C_p-edf of both methods in a specific realization. And we can see that there's a large variation in the C_p values of LBS at $k = 7$, which results in choosing the 7-predictor model. However, there's no variation in the C_p values of BS at a given size k, and in this realization, it selects the 6-predictor model.

Table 1: The performance of AICc-hdf. The true model setup is Orth-Sparse-Ex1. The columns involving 'edf' refer to infeasible selection rules since edf is estimated as if the true model is known, while other columns correspond to feasible rules.

			C_p		AICc		BIC		CV
			edf	ndf/hdf/bdf	edf	ndf/hdf/bdf	edf	ndf/hdf/bdf	
			% worse than the best possible BS						
n=200	hsnr	p=30	4	84/5/7	2	83/3/5	0	27/0/0	22
		p=180	1	345/37/39	0	398/1/2	0	214/0/0	24
	lsnr	p=30	20	24/35/33	20	24/38/35	69	24/69/68	28
		p=180	16	109/36/35	18	132/22/22	25	51/25/25	18
n=2000	hsnr	p=30	3	86/4/6	3	86/3/6	0	10/0/0	24
		p=180	1	339/1/2	0	342/1/2	0	61/0/0	22
	lsnr	p=30	3	85/5/6	3	85/5/6	0	9/0/0	23
		p=180	0	339/4/3	0	342/3/3	0	61/1/1	26
			Relative efficiency						
n=200	hsnr	p=30	0.96	0.54/0.95/0.93	0.98	0.54/0.97/0.96	1	0.79/1/1	0.82
		p=180	0.99	0.22/0.73/0.72	1	0.2/0.99/0.98	1	0.32/1/1	0.81
	lsnr	p=30	1	0.97/0.89/0.9	1	0.97/0.87/0.89	0.71	0.97/0.71/0.71	0.93
		p=180	1	0.55/0.85/0.85	0.98	0.5/0.95/0.95	0.93	0.77/0.93/0.93	0.98
n=2000	hsnr	p=30	0.97	0.54/0.97/0.94	0.97	0.54/0.97/0.95	1	0.91/1/1	0.8
		p=180	0.99	0.23/0.99/0.98	1	0.23/0.99/0.98	1	0.62/1/1	0.82
	lsnr	p=30	0.97	0.54/0.95/0.94	0.97	0.54/0.95/0.94	1	0.92/1/1	0.81
		p=180	1	0.23/0.96/0.97	1	0.23/0.97/0.97	1	0.62/0.99/0.99	0.79
			Sparsistency (number of extra variables)						
n=200	hsnr	p=30	6(0.1)	6(3.7)/6(0.2)/6(0.2)	6(0)	6(3.6)/6(0)/6(0.1)	6(0)	6(0.6)/6(0)/6(0)	6(0.6)
		p=180	6(0)	6(32.4)/6(7.5)/6(7.4)	6(0)	6(48.6)/6(0)/6(0)	6(0)	6(10)/6(0)/6(0)	6(0.5)
	lsnr	p=30	4.6(2.1)	5.3(3.8)/4.2(4.8)/4.2(3.8)	4.3(1.3)	5.3(3.8)/3.3(2.2)/3.5(1.8)	0.1(0)	3.7(0.6)/0.1(0)/0.2(0)	3.9(1.9)
		p=180	1.9(0.5)	5.3(32.2)/1.8(11.3)/1.9(10.3)	1.1(0.1)	5.6(48.8)/0.5(0)/0.6(0)	0(0)	4.3(8.4)/0(0)/0(0)	1.8(0.7)
n=2000	hsnr	p=30	6(0.1)	6(3.8)/6(0.1)/6(0.2)	6(0.1)	6(3.8)/6(0.1)/6(0.1)	6(0)	6(0.1)/6(0)/6(0)	6(0.7)
		p=180	6(0)	6(27.5)/6(0)/6(0)	6(0)	6(28.2)/6(0)/6(0)	6(0)	6(1.1)/6(0)/6(0)	6(0.4)
	lsnr	p=30	6(0.1)	6(3.8)/6(0.1)/6(0.1)	6(0)	6(3.8)/6(0.1)/6(0.1)	6(0)	6(0.1)/6(0)/6(0)	6(0.6)
		p=180	6(0)	6(27.6)/6(0)/6(0)	6(0)	6(28.3)/6(0)/6(0)	6(0)	6(1.1)/6(0)/6(0)	6(0.5)

Table 2: The performance of AICc-hdf. The true model setup is Orth-Dense. Details of the columns can be referred to the caption in Table 1.

			C_p		AICc		BIC		CV
			edf	ndf/hdf/bdf	edf	ndf/hdf/bdf	edf	ndf/hdf/bdf	
			% worse than the best possible BS						
n=200	hsnr	p=30	1	11/1/2	1	13/2/2	1	28/3/5	6
		p=180	8	44/20/20	9	51/18/19	17	25/38/41	14
	lsnr	p=30	15	11/16/15	20	11/21/20	27	16/27/27	16
		p=180	8	86/22/23	7	103/8/8	7	41/7/7	10
n=2000	hsnr	p=30	0	1/0/0	0	1/0/0	0	18/0/1	1
		p=180	6	34/8/8	6	34/8/8	21	6/35/37	10
	lsnr	p=30	2	11/3/3	3	11/3/3	45	40/35/44	10
		p=180	7	48/10/10	8	49/10/10	24	8/46/49	13
			Relative efficiency						
n=200	hsnr	p=30	1	0.91/1/0.99	1	0.9/1/0.99	1	0.79/0.98/0.96	0.95
		p=180	1	0.75/0.9/0.9	0.99	0.71/0.91/0.91	0.92	0.86/0.78/0.76	0.95
	lsnr	p=30	0.95	0.99/0.95/0.95	0.92	1/0.91/0.92	0.87	0.95/0.87/0.87	0.95
		p=180	0.99	0.58/0.88/0.88	1	0.53/1/1	1	0.76/1/1	0.98
n=2000	hsnr	p=30	1	0.99/1/1	1	0.99/1/1	1	0.85/1/0.99	0.99
		p=180	1	0.79/0.98/0.98	1	0.79/0.98/0.98	0.87	0.99/0.79/0.78	0.96
	lsnr	p=30	1	0.92/0.99/0.99	1	0.92/0.99/0.99	0.71	0.73/0.76/0.71	0.93
		p=180	1	0.72/0.98/0.98	1	0.72/0.98/0.98	0.87	1/0.73/0.72	0.95
			Number of variables						
n=200	hsnr	p=30	30	24.7/29.6/29	30	24.2/29.5/28.8	30	20.9/28.8/27.7	26.7
		p=180	20.3	52.7/35.9/34.6	18.1	62/16.2/16.2	16.1	34.9/13.7/13.5	18.8
	lsnr	p=30	12.9	10.7/15.6/13.7	7.2	10.5/8.6/7.9	0	4.1/0/0.1	8
		p=180	0.8	39/13.8/13.4	0.2	55/0.2/0.3	0	12.4/0/0	1.6
n=2000	hsnr	p=30	30	29.8/30/29.9	30	29.8/30/29.9	30	28.6/30/29.9	29.8
		p=180	32.2	58.8/32.5/32.4	31.9	58.7/31.6/31.6	26.6	31.3/25.1/24.9	32.4
	lsnr	p=30	28.5	19.9/28/27	28.5	19.9/28/26.9	12.9	12.5/16.9/14.5	22.3
		p=180	14.1	43.7/14.1/14.1	13.8	44.1/13.4/13.4	9.1	13.4/7/6.8	14.4

Table 3: The performance of BS compared to regularization methods. The selection rules are AICc-hdf for BS, AICc-ndf for LASSO and 10-fold CV for SparseNet, respectively.

			Orth-Sparse-Ex1			Orth-Sparse-Ex2			Orth-Dense		
			BS	LASSO	SparseNet	BS	LASSO	SparseNet	BS	LASSO	SparseNet
			% worse than the best possible BS								
n=200	hsnr	p=30	3	70	17	26	32	21	2	1	5
		p=180	1	139	26	11	54	17	18	15	7
	lsnr	p=30	38	7	17	33	21	25	21	-6	0
		p=180	22	5	9	37	30	26	8	0	3
n=2000	hsnr	p=30	3	70	17	7	71	17	0	2	1
		p=180	1	129	18	9	125	18	8	16	4
	lsnr	p=30	5	70	16	13	45	16	3	0	7
		p=180	3	129	19	7	86	17	10	8	5
			Relative efficiency								
n=200	hsnr	p=30	1	0.39	0.74	0.95	0.84	1	0.99	1	0.93
		p=180	1	0.19	0.59	1	0.51	0.87	0.82	0.87	1
	lsnr	p=30	0.6	1	0.84	0.8	1	0.93	0.62	1	0.88
		p=180	0.76	1	0.93	0.83	0.96	1	0.88	1	0.94
n=2000	hsnr	p=30	1	0.4	0.78	1	0.43	0.84	1	0.96	0.97
		p=180	1	0.21	0.71	1	0.25	0.84	0.93	0.8	1
	lsnr	p=30	1	0.41	0.81	1	0.64	0.95	0.95	1	0.88
		p=180	1	0.22	0.72	1	0.35	0.82	0.91	0.94	1
			Sparsistency (number of extra variables)								
n=200	hsnr	p=30	6(0)	6(8.3)	6(2)	4.4(0.2)	5.8(8.4)	5.1(3)	29.5	29.4	27.4
		p=180	6(0)	6(23.7)	6(5.2)	4.1(0)	5.4(20.1)	4.7(6.3)	16.2	67.4	31.8
	lsnr	p=30	3.3(2.2)	5.4(7.1)	4.7(5.6)	2.5(0.7)	4(5.6)	3(3)	8.6	15.1	13.1
		p=180	0.5(0)	4(15.8)	3.4(12.5)	1.2(0.1)	3(13.3)	2.4(7.9)	0.2	16.4	13.1
n=2000	hsnr	p=30	6(0.1)	6(8.6)	6(2)	6(0.2)	6(8.4)	6(1.8)	30	30	29.8
		p=180	6(0)	6(19)	6(3.4)	6(0.1)	6(18.7)	6(2.5)	31.6	89.2	45
	lsnr	p=30	6(0.1)	6(8.4)	6(1.7)	4.2(0.4)	5.1(7.1)	4.3(1.9)	28	27.6	24
		p=180	6(0)	6(19.3)	6(2.5)	4(0.1)	4.6(15.4)	4.1(2.7)	13.4	53.3	27.2

Table 4: The performance of BS vs LBS. The selection rule for both methods is C_p-edf. We assume the knowledge of μ and σ.

			Orth-Sparse-Ex1		Orth-Sparse-Ex2		Dense	
			BS	LBS	BS	LBS	BS	LBS
			% worse than the best possible BS					
n=200	hsnr	p=30	4	29	22	27	1	1
		p=180	1	51	11	28	8	11
	lsnr	p=30	20	23	22	30	15	14
		p=180	16	21	16	32	8	15
n=2000	hsnr	p=30	3	32	4	31	0	0
		p=180	1	50	1	52	6	9
	lsnr	p=30	3	30	7	27	2	2
		p=180	0	51	1	48	7	12
			Relative efficiency					
n=200	hsnr	p=30	1	0.64	1	0.91	1	1
		p=180	1	0.39	1	0.71	1	0.94
	lsnr	p=30	1	0.96	1	0.87	0.91	0.93
		p=180	1	0.9	1	0.76	0.98	0.85
n=2000	hsnr	p=30	1	0.6	1	0.62	1	1
		p=180	1	0.4	1	0.39	1	0.95
	lsnr	p=30	1	0.61	1	0.69	0.99	1
		p=180	1	0.39	1	0.41	1	0.92
			Sparsistency (number of extra variables)					
n=200	hsnr	p=30	6(0.1)	6(0.8)	4.8(0.4)	5(1)	30	30
		p=180	6(0)	6(1.1)	4.2(0)	4.6(1)	20.3	22.4
	lsnr	p=30	4.6(2.1)	4.6(2.2)	2.7(0.5)	3(1.1)	12.9	9.2
		p=180	1.9(0.5)	2.8(1.9)	1.9(0.3)	2.2(1.3)	0.8	3.7
n=2000	hsnr	p=30	6(0.1)	6(0.9)	6(0.1)	6(0.8)	30	30
		p=180	6(0)	6(1)	6(0)	6(1.2)	32.2	34.2
	lsnr	p=30	6(0.1)	6(0.8)	4.1(0.1)	4.3(0.8)	28.5	29.3
		p=180	6(0)	6(1.1)	4(0)	4.1(1.2)	14.1	16.3

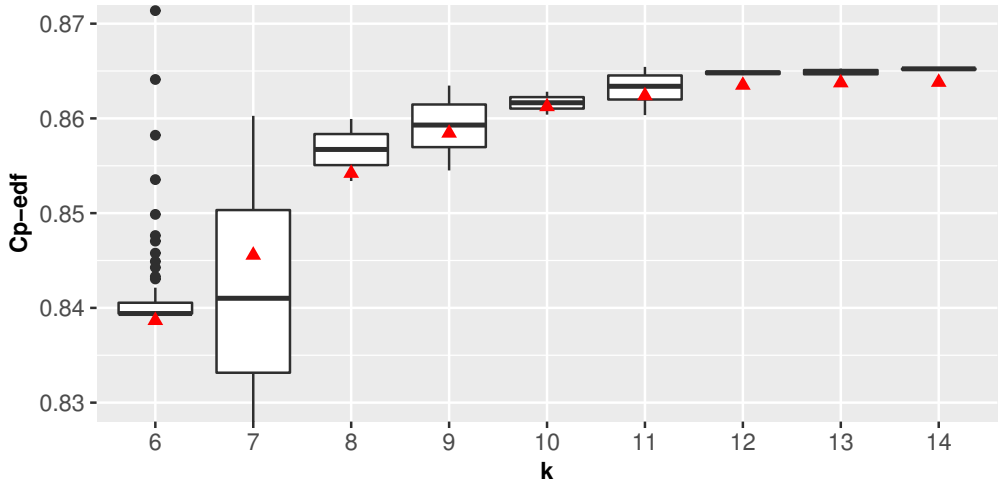

Figure 4: C_p-edf for a specific realization. The boxes are for LBS since multiple λ correspond to the same solution, while the red triangles are for BS. The true model is Orth-Sparse-Ex1 with $n = 200$, $p = 14$, $p_0 = 6$ and high SNR.

3 Best orthogonalized subset selection (BOSS)

With a general X, BS is not computationally feasible for large problems. In this section, we propose a LS-based subset selection method BOSS, that has cost on the same order of a multiple regression on all the predictors.

3.1 The method and its computational cost

The detailed implementation process of BOSS is described in Algorithm 2. The main steps can be summarized as follows: (1) order and orthogonalize the predictors, (2) perform BS on the set of orthogonal predictors, (3) transform the coefficients back to the original space, and (4) use a selection rule such as AICc-hdf to choose the optimal subset. The computation of BOSS has an overall cost of $O(np^2)$. Note that BOSS is based on things that have already

established in section 2.

As can be seen as follows, the computation of BOSS has an overall cost of $O(np^2)$, the same cost as OLS on all p predictors and LASSO. At step k, we have a set of ordered predictors $X_{S_{k-1}}$ and its orthogonal basis $Q_{S_{k-1}}$). Then from the remaining $p-k+1$ predictors, we choose the one that has the largest correlation with y conditioning on $Q_{S_{k-1}}$, that is the correlation between y and the residual from regressing a candidate predictor on $Q_{S_{k-1}}$. The regression part costs $O(n)$ since we maintain the regression result, e.g. estimated coefficients and residual, in the previous steps, and only need to perform a simple linear regression upon the predictor joined in step $k-1$, i.e. the last column in $Q_{S_{k-1}}$. Repeating the above step for all $p-k+1$ predictors costs $O(n(p-k+1))$. We then update the QR decomposition, by adding the chosen predictor as a new column, which costs $O(n(p-k))$ via the modified Gram-Schmidt algorithm as discussed in Hammarling and Lucas (2008). Therefore, we end up with an ordered set of predictors X_{S_p} and its corresponding QR decomposition Q_{S_p} and R_{S_p}. We regress y upon Q_{S_p} which costs $O(np)$, and denote the coefficient vector as z. BOSS, then performs BS on Q_{S_p}, which is indeed ranking of predictors based on their magnitudes of corresponding element in z, and the cost is $O(p\log(p))$. Once we have the solution path of BOSS, we then apply AICc-hdf to choose the optimal subset size (denoted as k_Q), where hdf is calculated via Algorithm 1 by inputting Q_{S_p}. The entire BOSS-AICc-hdf procedure costs $O(np^2)$.

The ordering of predictors is essential in terms of both getting a sparse solution and a better predictive performance. Consider a sparse true model with only two uncorrelated predictors $X=[X_1,X_2]$, $\beta=[0,1]^T$ and a high SNR. Based on the evidence we see from the previous section, the best model in such a scenario is LS regression on X_2. Without the ordering step, the orthogonal basis is $Q=[Q_1,Q_2]$ s.t. $X=QR$, i.e. the predictors are orthogonalized in their physical orders. The 1-predictor model ($k_Q=1$) of BS can either be Q_1 or Q_2, which when transformed back to the space of X do not correspond to LS regression upon X_2. The

Algorithm 2 Best Orthogonalized Subset Selection (BOSS)

1. Standardize y and the columns of X to have mean 0, and denote the means as $\bar{X}$ and $\bar{y}$.

   Order and orthogonalize the predictors:

2. From the p predictors, select the one that has the largest marginal correlation with the response y, and denote it as X_{S_1}. Standardize X_{S_1} to have unit l_2 norm and denote it as Q_{S_1}. Calculate R_{S_1} such that $X_{S_1} = Q_{S_1} R_{S_1}$. Let $S = \{1, \cdots, p\}$. Initialize vectors $\text{resid}_j = X_j$ where $j = 1, \cdots, p$.

3. For $k = 2, \cdots, p$:

   a. For each of the $p - k + 1$ predictors X_j in $X_{S \setminus S_{k-1}}$, calculate its partial correlations with the response y conditioning on $Q_{S_{k-1}}$.

      a1. Regress X_j on $Q_{S_{k-1} \setminus S_{k-2}}$ ($S_{k-2} = \emptyset$ if $k = 2$), and denote the estimated coefficient as r. Update $\text{resid}_j = \text{resid}_j - r Q_{S_{k-1} \setminus S_{k-2}}$.

      a2. Calculate the correlation between y and resid_j.

   b. Select the predictor that has the largest partial correlation in magnitude, augment S_{k-1} with this predictor and call it S_k.

   c. Update $Q_{S_{k-1}}$ and $R_{S_{k-1}}$ given the newly added column $X_{S_k \setminus S_{k-1}}$, and call them Q_{S_k} and R_{S_k}. The update is based on the modified Gram-Schmidt algorithm as discussed in Hammarling and Lucas (2008).

   BS on the orthogonalized predictors Q_{S_p}:

4. Calculate $\tilde{\gamma}_j(k_Q) = z_j \mathbb{1}(|z_j| \geq |z_{(k_Q)}|)$, i.e. the j-th component of coefficient vector at subset size k_Q, where $z = Q_{S_p}^T y$ and $z_{(k_Q)}$ is the k-th largest entry in absolute values. Let $\tilde{\Gamma} = [\tilde{\gamma}(0)\tilde{\gamma}(1) \cdots \tilde{\gamma}(p)]$.

   Transform back to the original space:

5. Project $\tilde{\Gamma}$, a $p \times (p+1)$ matrix, to the original space of X_{S_p}, i.e. back solving $R\tilde{B} = \tilde{\Gamma}$, and re-order the rows of $\tilde{B}$ to their correspondences in X, i.e. $\hat{B} = O\tilde{B}$ where O represents the ordering matrix s.t. $X_{S_p} = XO$. The intercept vector is $\hat{B}_0 = \bar{y}\mathbb{1} - \hat{B}^T \bar{X}$.

   Select the optimal subset:

6. Select the optimal subset $k_Q^\star$ using AICc-hdf, where hdf is calculated via Algorithm 1, by inputting (Q_{S_p}, y). The inclusion of an intercept term indicates that $\text{hdf}(k_Q)$ shall increase by 1.

former corresponds to LS estimates upon X_1, while the latter is a linear combination of LS estimates upon X and LS estimates upon X_1; the former leads to a completely wrong model while the latter results in non-zero coefficients on both predictors. Therefore, the ordering step is crucial to both sparsity as well as predictive performance. Note that we show the coefficients of BOSS can be expressed as a linear combination of LS coefficients on subsets of X in Theorem 2 and the proof can be found in Appendix C.

BS on the set of orthogonalized predictors, gives the chance for BOSS to 'look back' at the predictors that are already stepped in. One may notice that BOSS is similar to forward stepwise regression (FS), which was first introduced in Efroymson (1960). FS orders and orthogonalizes the predictors in the same way as BOSS. It then takes a nested subsets $Q_{S_1}, Q_{S_2}, \cdots, Q_{S_p}$ as the candidate subsets and performs LS regression upon them. Therefore, once a predictor is stepped in, it remains in the subsets of every following steps of FS. As we will show in the next section, under certain circumstances, FS can easily overfit, since noise predictors (those with $\beta_j = 0$) step in at early steps. However, BOSS re-visits the predictors that have already stepped in and allows them to be dropped, resulting in a better predictive performance than FS.

Theorem 2 (Coefficients of BOSS are a linear combination of LS coefficients on subsets of X)**.** *Suppose X has full column rank and the columns are already ordered. $X = QR$ where Q is a $n \times p$ matrix with orthonormal columns and R is a $p \times p$ upper-triangular matrix. Let $S_k = \{j_1, j_2, \cdots, j_k\}$ denote the support (position of predictors) of the best k-predictor model given by BS upon (Q, y), and use $\hat{\gamma}(k)$ (p by 1) to denote the BS coefficients. The corresponding coefficients in the X space, i.e. $\hat{\beta}(k)$ s.t. $R\hat{\beta}(k) = \hat{\gamma}(k)$, can be expressed as*

$$\hat{\beta}(k) = \sum_{j \in S_k} \hat{\alpha}^{(j)} - \hat{\alpha}^{(j-1)},$$

where the first j entries in $\hat{\alpha}^{(j)}$ (p by 1) are LS coefficients of regressing y upon $[X_1, X_2, \cdots, X_j]$ (the first j columns in X), and the rest $p - j$ entries are zero.

3.2 Numerical justification of using hdf for BOSS

The hdf is designed for BS on a set of orthogonal predictors. However, before performing BS on the orthogonal basis, BOSS first orders the predictors. This raises a question that whether hdf is reasonable to use in the selection rules for BOSS.

Figure 5 compares C_p-edf and C_p-hdf for BOSS under various true models. We prefer C_p for this comparison since it is defined for any general fitting procedure. The details of setups for the sparse and dense models can be referred to section 3.3.1 where they correspond to Sparse-Ex3 and Dense designs, respectively. The correlation between predictors is $\rho = 0.5$. We see that C_p-hdf approximates C_p-edf well except when the SNR is low, where we see discrepancies at early steps. Moreover, both criteria lead to the same average optimal subset size.

3.3 The performance of BOSS

We now study the performance of BOSS via simulations. We first show that BOSS can provide a better solution path than FS, and we further compare BOSS with regularization methods.

3.3.1 Simulation setups

We consider a similar setup as in section 2.3.1, but with a general X, where $x_i \sim \mathcal{N}(0, \Sigma)$, $i = 1, \cdots, n$ are independent realizations from a p-dimensional multivariate normal distribution with mean zero and covariance matrix $\Sigma = (\sigma_{ij})$.

The correlation structure and true coefficient vector β include the following scenarios:

- Sparse-Ex1: **All the predictors (both signal and noise) are correlated.** We take $\sigma_{i,j} = \rho^{|i-j|}$ for $i, j \in \{1, \cdots, p\} \times \{1, \cdots, p\}$. As to β, we have $\beta_j = 1$ for p_0 equispaced values and 0 everywhere else.

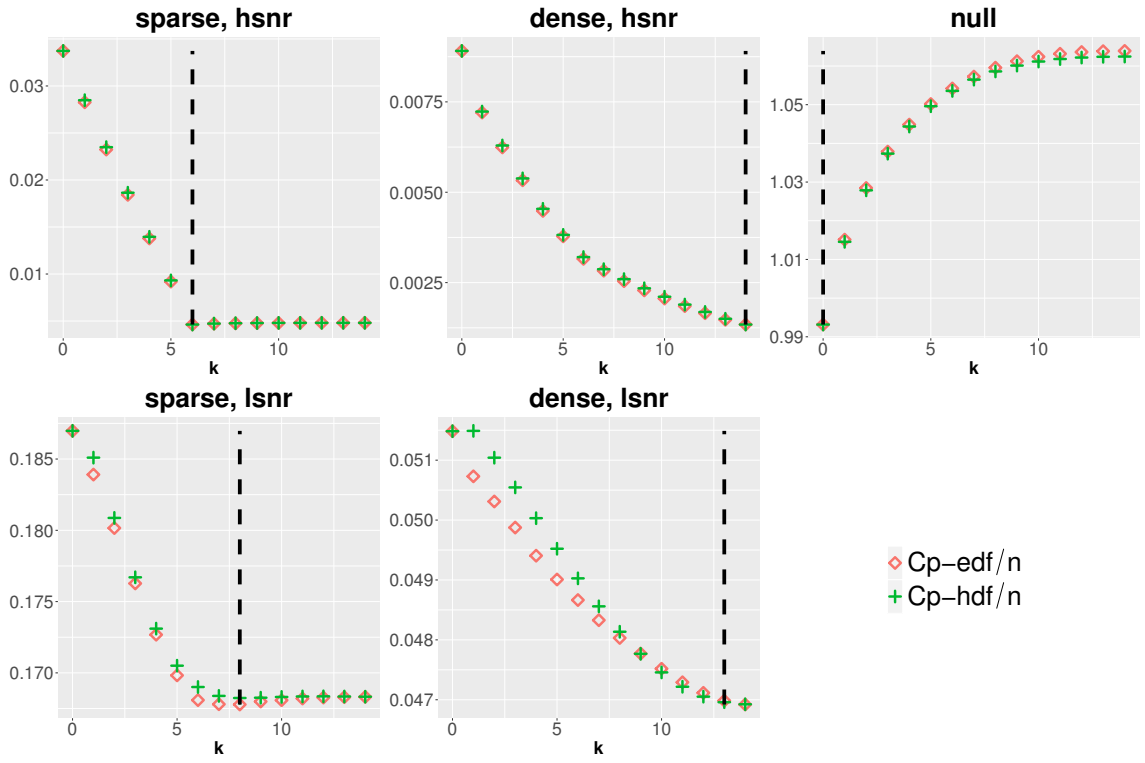

Figure 5: C_p-edf vs C_p-hdf for BOSS. Here X is general with $n = 200$, $p = 14$. The optimal subset size $k_Q^\star$ is selected by minimizing a criterion. Both criteria result in the same average $k_Q^\star$ over 1000 replications (round to integer) as denoted by the dashed vertical lines. We assume the knowledge of μ and σ.

- Sparse-Ex2: **Signal predictors are pairwise correlated with opposite effects.** We take $\sigma_{i,j} = \sigma_{j,i} = \rho$ for $1 \le i < j \le p_0$. Other off-diagonal elements in Σ are zero. For the true coefficient vector, we have $\beta_{2j-1} = 1$ and $\beta_{2j} = -1$ for $1 \le j \le p_0/2$, and all other $\beta_j = 0$ for $j = p_0 + 1, \cdots, p$.

- Sparse-Ex3: **Signal predictors are pairwise correlated with noise predictors.** We take $\sigma_{i,j} = \sigma_{j,i} = \rho$ for $1 \le i \le p_0$ and $j = p_0 + i$. Other off-diagonal elements in Σ are zero. $\beta = [1_{p_0}, 0_{p-p_0}]^T$.

- Sparse-Ex4: **Same correlation structure as Sparse-Ex2, but with varying strengths of coefficients.** We have $\beta_j = -\beta_{j+1}$ where $j = 2k+1$ and $k = 0, 1, \cdots, p_0/2 - 1$. Suppose that $\beta' = [1, 5, 10]$, then $\beta_j = \beta'_k$ where $k = j(\text{mod}3)$.

- Dense: **Same correlation structure as Ex1, but with diminishing strengths of coefficients**. The true coefficient vector has: $\beta_j = (-1)^j \exp(-\frac{j}{\kappa})$, $j = 1, \cdots, p$, and here $\kappa = 10$.

The setup of Sparse-Ex1 is very common in the literature, such as in Bertsimas et al. (2016) and Hastie et al. (2017). All of the predictors are correlated (when $\rho \neq 0$) where the strength of correlation depends on the physical positions of variables. Sparse-Ex2 is designed such that the pair of correlated predictors, e.g. (X_1, X_2), leads to a good fit (high R^2), while either single one of them contribute little to the fitted R^2. Sparse-Ex4 is similar to Sparse-Ex2, but has varying strengths of coefficients for the true predictors. In Sparse-Ex3, signal predictors are only correlated with the noise ones. Finally, the dense setup is built on the dense example in section 2.3.1, by having correlated predictors.

For the sparse examples, we take $p_0 = 6$. We consider three values of the correlation parameter, $\rho \in [0, 0.5, 0.9]$. Other configuration options, including n, p, and SNR, are the same as in Section 2.3.1. This implies a total of 360 different combinations of configuration options. For each configuration, 1000 replications are estimated and we present the same evaluation measures as introduced in Section 2.3.1. The full set of results can be found in the supplemental material.

3.3.2 The solution paths of BOSS and FS

Unlike FS whose candidate subsets are nested, BOSS performs an extra step of BS upon Q_{S_p}, which raises the question of whether the extra step brings any benefit. We set aside the selection rule for now, and focus on the solution paths of the two methods.

Figure 6 shows two examples of the average RMSE along the solution paths of BS, FS and BOSS. When the true model is Sparse-Ex3, all three methods provide almost the same solution path. However, for Sparse-Ex4, we see a clear advantage of BOSS over FS in early steps up until about the 8th step. Recall that in Sparse-Ex4, there are $p_0 = 6$ predictors with $\beta_j \neq 0$ that are pairwise correlated with opposite effects, where each pair say (X_1, X_2) together leads to a high R^2 but each single one of them (X_1 or X_2) contributes little. Since the subsets along the solution path of FS are nested, if a noise predictor steps in during early steps, which is very likely when the correlation between them is high, it remains in the subsets of every following steps. Hence the subset containing both X_1 and X_2 may appear in a late stage. However, BOSS takes ordered predictors provided by FS, and re-orders them by performing BS upon their orthogonal basis, which gives a greater chance for (X_1, X_2) to appear early in the solution path of BOSS and potentially results in a better predictive performance than FS. Furthermore, in this example, we notice that BOSS provides a better solution path than BS until step 5, and the two methods give similar performances in further steps.

3.3.3 The performance of BOSS compared to other methods

We now consider feasible implementations of methods. We looked at results using AICc-hdf, C_p-hdf and 10-fold CV for BOSS, and AICc-hdf was the best (see supplemental material), so that is what we will use here. For BS and FS, we will use 10-fold CV. Similar to our discussion in Section 2.3.3, we find that (see supplemental material) AICc-ndf performs similarly to 10-fold CV for LASSO, and that is what we will use for LASSO. For other regularization methods, the selection rule will be 10-fold CV. According to our results (see supplemental material), SparseNet is slightly better than relaxed LASSO and Gamma LASSO, and therefore we only present the result for SparseNet here.

A selected set of simulation results is presented in Table 5. Note that for BS, we only have results for $p \leq 30$, since it is fitted using the 'leaps' algorithm and p being around 30 is the

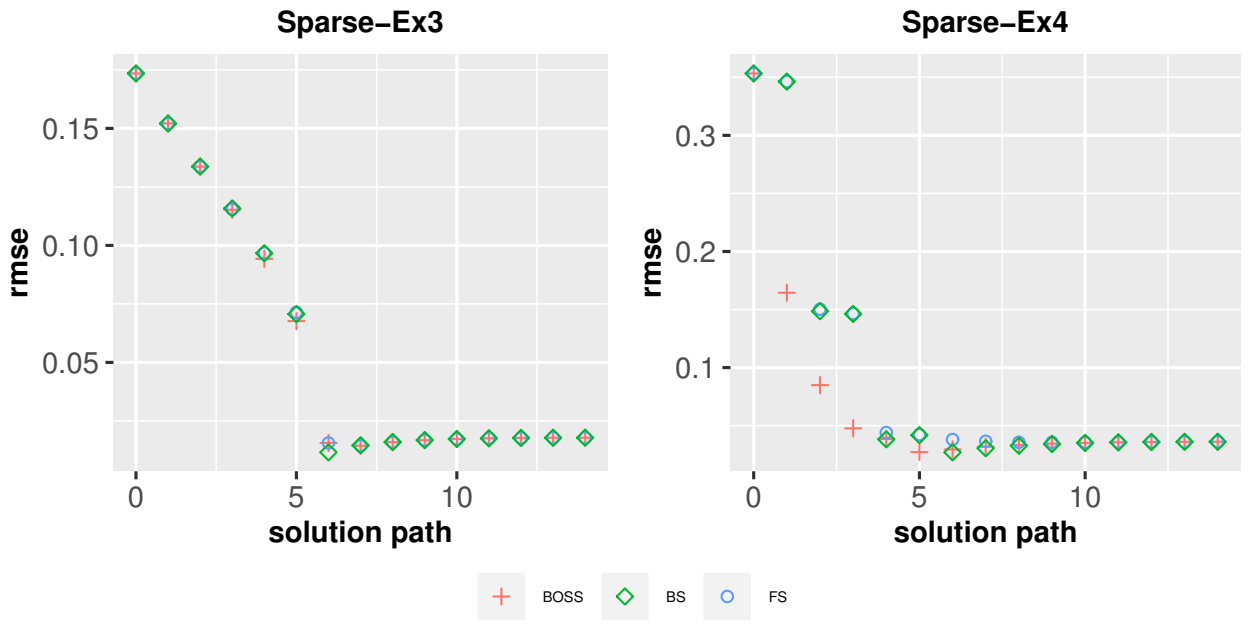

Figure 6: RMSE along the solution path, average over 1000 replications. For both BS and FS, the solution path is denoted by k while for BOSS, it's denoted by k_Q, i.e. the subset size in the orthogonalized space. In both scenarios, we have $n = 200$, $p = 14$, $\rho = 0.9$ and high SNR.

ad-hoc limit. Here is a brief summary of the results:

- BOSS is overall the best method for large sample size $n = 2000$. The only exception is when the true model is Sparse-Ex2 or Sparse-Ex4, and the correlation $\rho = 0.9$.
- Compared to FS and BS, BOSS is generally better. The advantage is obvious when the true model is sparse, with the only exception being $n = 200$ and low SNR, where we see similar performances given by the three methods. Also note BOSS uses AICc-hdf as the selection rule, which only requires fitting the procedure once and is much more efficient than BS and FS that rely on 10-fold CV.
- Compared to the regularization methods, with $n = 200$ and the true model being Sparse-Ex1 or Sparse-Ex3, we see that BOSS is the best when SNR is high, LASSO is the best

when SNR is low and SparseNet is in between. With $n = 2000$, BOSS is almost always the best even when SNR is low, followed by SparseNet, and LASSO is generally the worst. The findings for Sparse-Ex1 and Sparse-Ex3 generally apply to Sparse-Ex2 and Sparse-Ex4. An exception is when SNR is low and $n = 200$, where LASSO is no longer dominant. We see both BOSS and SparseNet perform similarly to LASSO, with no clear winner in this regime. Furthermore, with a Dense true model, all three methods perform similarly and no clear winner emerges. These findings correspond to our discussion in Section 2.3.3, where we compare the performance of BS with regularization methods under an orthogonal X.

- In terms of support recovery in the sparse true models, all of the methods can recover the true predictors (those with $\beta_j \neq 0$) when SNR is high or the sample size n is large. LS-based methods (BOSS, FS, BS) hardly include any noise predictors (those with $\beta_j = 0$). However, SparseNet and LASSO generally overfit, with the latter being worse in that regard. In the low SNR and small n scenario, LASSO and SparseNet are better in terms of recovering true predictors, but it comes with a price of including more false positives.

4 Real data analysis

In this section, we implement BOSS on five real world datasets. We consider four datasets from the StatLib library[2], which is maintained at Carnegie Mellon University. The 'Housing' data are often used in comparisons of different regression methods. The aim is to predict the housing values in the suburbs of Boston based on 13 predictors, including crime rate, property tax rate, pupil-teacher ratio, etc. The 'Hitters' data contains the 1987 annual salary for MLB players. For each player, it records 19 different performance metrics happening in 1986, such as number of times at bat, number of hits, etc, and the task is to predict the salary

[2]http://lib.stat.cmu.edu/datasets/

based on these statistics. The 'Auto' data is driven by prediction of the miles per gallon of vehicles based on features like the horsepower, weight, etc. The 'College' data contains various statistics for a large number of US colleges from the 1995 issue of 'US News and World Report', and we use these statistics to predict the number of applications received. We also consider a dataset from the Machine Learning Repository[3], that is maintained by UC Irvine. The 'ForestFire' data is provided by Cortez and Morais (2007) and the aim is to use meteorological and other data to predict the burned area of forest fires that happened in the northeast region of Portugal. The authors considered several machine learning algorithms, e.g. support vector regression, and concluded that the best prediction in terms of RMSE is the naive mean vector.

In real data analysis, one almost always would consider an intercept term. The way that BOSS handles the intercept term is discussed in Algorithm 2. To be more specific, we first center both X and y, and fit BOSS-AICc-hdf without an intercept to get $\hat{\beta}$. Then we calculate the intercept by $\hat{\beta}_0 = \bar{y} - \bar{X}^T\hat{\beta}$, which can be easily shown to be equivalent to fitting an intercept in every subset considered by BOSS.

We compare the performance of BOSS with LS-based methods BS and FS, and with regularization methods LASSO and SparseNet. All of the methods are fitted with an intercept term. Note that for the Forest Fires dataset, we fit BS via MIO (Bertsimas et al., 2016) using the `R` package **bestsubset** (Hastie et al., 2017), where we restrict subset size $k = 0, \cdots, 10$, with 3 minutes as the time budget to find an optimal solution at each k, as suggested by the authors. For all of the other datasets, BS is fitted using the **leaps** package. To measure the performance of each method, we apply the leave-one-out (LOO) testing procedure, in which we fit the method on all observations expect one, test the performance on that particular observation, and repeat the procedure for all n observations.

Table 6 presents the average RMSE, the average number of predictors and average running

[3]https://archive.ics.uci.edu/ml

time for various subset selection methods given by LOO testing. We see that BOSS has the fastest running time for each dataset. It is also the best in terms of predictive performance in all datasets except the 'Housing' data where LASSO is the best for that dataset and its RMSE is 0.5% lower than that of BOSS.

5 Conclusion and future work

In this paper, we introduce a heuristic degrees of freedom (hdf) for best subset regression based on an orthogonal X. We further propose the KL-based information criterion AICc-edf and its feasible implementation AICc-hdf. We demonstrate that they are approximate the unbiased estimator of KL-based expected predictive error $\widehat{\text{Err}}$ well. More importantly, they result in the same choice of subset as $\widehat{\text{Err}}$ when they are used as selection rules for BS. Furthermore, we propose an LS-based subset selection method BOSS. BOSS together with the selection rule AICc-hdf has computational cost on the same order of OLS. Finally, we show in simulations and real data examples that BOSS can be the best method in terms of speed as well as predictive performance.

Throughout the paper, we focus on the case where $n > p$ and potential future work is to extend our results to $n \leq p$. Further, an interesting question is whether AICc-hdf can be applied as a selection rule for BS and FS when X is general.

Table 5: The performance of BOSS compared to other methods. Selection rules are AICc-hdf for BOSS, AICc-ndf for LASSO and 10-fold CV for the rest methods in the table, respectively.

				Sprse-Ex3					Sparse-Ex4					Dense				
				BOSS	FS	BS	LASSO	SparseNet	BOSS	FS	BS	LASSO	SparseNet	BOSS	FS	BS	LASSO	SparseNet
				% worse than the best possible BOSS					% worse than the best possible BOSS					% worse than the best possible BOSS				
n=200	hsnr	$\rho = 0.5$	p=30	2	24	23	70	17	19	28	23	49	23	1	6	9	1	6
			p=180	1	21	NA	134	20	4	15	NA	82	15	16	17	NA	39	8
		$\rho = 0.9$	p=30	3	30	22	71	18	21	56	21	73	30	2	7	9	2	7
			p=180	3	28	NA	126	19	7	66	NA	122	-9	18	12	NA	41	15
	lsnr	$\rho = 0.5$	p=30	30	25	25	0	9	37	35	24	34	24	18	16	15	4	10
			p=180	11	12	NA	-4	4	32	34	NA	33	21	5	8	NA	2	5
		$\rho = 0.9$	p=30	31	25	25	0	9	35	78	19	70	44	16	12	14	10	14
			p=180	17	16	NA	-4	3	16	34	NA	31	34	12	8	NA	12	9
n=2000	hsnr	$\rho = 0.5$	p=30	3	23	21	73	16	7	22	22	85	16	0	1	1	2	0
			p=180	1	22	NA	129	18	6	21	NA	174	17	8	12	NA	33	8
		$\rho = 0.9$	p=30	3	23	22	76	17	32	32	16	107	13	0	1	1	3	1
			p=180	1	22	NA	135	17	15	89	NA	227	8	9	8	NA	38	15
	lsnr	$\rho = 0.5$	p=30	5	24	23	72	18	14	24	24	62	18	2	9	11	2	8
			p=180	4	23	NA	129	17	8	19	NA	125	15	11	17	NA	31	10
		$\rho = 0.9$	p=30	7	24	23	55	11	29	41	17	84	13	3	9	11	2	9
			p=180	4	18	NA	94	5	14	106	NA	178	15	12	11	NA	38	17
				Relative efficiency					Relative efficiency					Relative efficiency				
n=200	hsnr	$\rho = 0.5$	p=30	1	0.66	0.67	0.39	0.76	1	0.85	0.9	0.64	0.93	1	0.9	0.86	0.99	0.91
			p=180	1	0.64	NA	0.2	0.63	1	0.79	NA	0.33	0.81	0.87	0.85	NA	0.61	1
		$\rho = 0.9$	p=30	1	0.62	0.69	0.39	0.75	0.98	0.62	1	0.51	0.89	1	0.9	0.87	0.99	0.9
			p=180	1	0.61	NA	0.23	0.71	0.7	0.29	NA	0.17	1	0.9	1	NA	0.63	0.95
	lsnr	$\rho = 0.5$	p=30	0.58	0.64	0.64	1	0.83	0.81	0.84	0.99	0.89	1	0.79	0.81	0.82	1	0.9
			p=180	0.76	0.74	NA	1	0.87	0.86	0.82	NA	0.87	1	0.94	0.88	NA	1	0.93
		$\rho = 0.9$	p=30	0.58	0.64	0.63	1	0.82	0.78	0.48	1	0.53	0.71	0.88	0.95	0.93	1	0.94
			p=180	0.68	0.68	NA	1	0.86	1	0.77	NA	0.82	0.78	0.92	1	NA	0.95	0.99
n=2000	hsnr	$\rho = 0.5$	p=30	1	0.69	0.69	0.38	0.78	1	0.76	0.76	0.37	0.86	1	0.99	0.99	0.95	0.99
			p=180	1	0.63	NA	0.21	0.71	1	0.73	NA	0.17	0.82	1	0.92	NA	0.66	1
		$\rho = 0.9$	p=30	1	0.68	0.69	0.37	0.77	0.76	0.72	0.92	0.33	1	1	0.98	0.98	0.95	0.99
			p=180	1	0.63	NA	0.2	0.73	0.94	0.3	NA	0.12	1	1	1	NA	0.62	0.89
	lsnr	$\rho = 0.5$	p=30	1	0.71	0.72	0.41	0.8	1	0.85	0.86	0.53	0.95	0.99	0.86	0.84	1	0.89
			p=180	1	0.67	NA	0.23	0.76	1	0.78	NA	0.24	0.87	0.98	0.88	NA	0.71	1
		$\rho = 0.9$	p=30	1	0.75	0.76	0.54	0.93	0.78	0.65	0.92	0.4	1	0.99	0.88	0.85	1	0.88
			p=180	1	0.76	NA	0.33	0.96	1	0.28	NA	0.18	0.97	0.98	1	NA	0.65	0.91
				Sparsistency (number of extra variables)					Sparsistency (number of extra variables)					Sparsistency (number of extra variables)				
n=200	hsnr	$\rho = 0.5$	p=30	6(0)	6(0.7)	6(0.7)	6(7.8)	6(2)	4.4(0.2)	4.8(1)	5(1)	5.7(10.1)	4.8(2.9)	29.6	26.3	25.1	29.2	27
			p=180	6(0)	6(0.4)	NA	6(16.8)	6(3.3)	4(0)	4.1(0.4)	NA	5.1(20)	4.3(4.6)	16.6	18.1	NA	61	29.4
		$\rho = 0.9$	p=30	6(0.1)	6(0.9)	6(0.7)	6(8.4)	6(2.2)	5.1(2.8)	4.9(4.3)	5(0.8)	5.8(17.6)	4.5(3.8)	29.3	24.6	23.1	28.9	25.8
			p=180	6(0.1)	6(0.5)	NA	6(15.9)	6(3.7)	4.2(2.4)	4.3(7.8)	NA	4.6(44.3)	4.1(4.2)	15.2	17.2	NA	64.7	36.1
	lsnr	$\rho = 0.5$	p=30	3(2.2)	3.6(2.5)	3.6(2.5)	5.2(6.7)	4.6(6.4)	2.2(1.1)	2.5(1.6)	2.6(1.1)	3.5(6.6)	2.8(3.7)	6	6.7	6.7	12.3	9.5
			p=180	0.3(0.1)	1(0.7)	NA	3(9.7)	2.5(9.6)	1(0.2)	1.4(0.9)	NA	2.2(10.9)	2(6.6)	0.2	0.9	NA	7.5	5.6
		$\rho = 0.9$	p=30	2.5(2.6)	2.9(3.1)	2.8(3)	4.3(7.4)	3.9(7)	2.5(3.7)	2.2(4.7)	2.7(1.1)	2.9(9.9)	2.8(8.2)	4.3	4.4	4.5	8.9	6.5
			p=180	0.4(0.2)	1.1(1)	NA	3.3(11.7)	2.8(11.7)	0.6(1.5)	0.2(0.6)	NA	0.3(4.7)	0.5(8.9)	0.9	1.5	NA	6.4	3.6
n=2000	hsnr	$\rho = 0.5$	p=30	6(0.1)	6(0.6)	6(0.6)	6(8.2)	6(1.9)	6(0.2)	6(0.6)	6(0.6)	6(10.7)	6(1.8)	30	30	30	30	30
			p=180	6(0)	6(0.4)	NA	6(21.3)	6(3.2)	6(0.1)	6(0.4)	NA	6(32)	6(2.5)	34.3	32.3	NA	109.5	41.3
		$\rho = 0.9$	p=30	6(0.1)	6(0.6)	6(0.6)	6(8.7)	6(2)	6(0.4)	6(1.3)	6(0.6)	6(17.5)	6(2.5)	30	29.9	29.9	30	29.9
			p=180	6(0)	6(0.4)	NA	6(22.8)	6(3.2)	6(1.4)	5.9(3.8)	NA	6(72)	6(9.2)	35.8	30.2	NA	114.7	48.8
	lsnr	$\rho = 0.5$	p=30	6(0.1)	6(0.6)	6(0.6)	6(8.1)	6(1.8)	4.2(0.5)	4.2(0.7)	4.3(0.7)	5.2(9.4)	4.3(2.1)	29.2	22.2	21.2	28.1	23.7
			p=180	6(0)	6(0.4)	NA	6(21.3)	6(2.3)	4(0.1)	4(0.4)	NA	4.6(25.7)	4.1(2.6)	16.2	14.1	NA	72.5	23.6
		$\rho = 0.9$	p=30	5.8(0.3)	5.8(1.1)	5.8(1.1)	6(8.8)	5.9(2.2)	4.5(2)	4.3(2.3)	4.3(0.7)	5.4(16.6)	4.2(3.4)	28.7	19.8	18.5	27.6	22.7
			p=180	5.7(0.3)	5.7(0.7)	NA	6(22.8)	5.8(2.2)	4.1(3.4)	3.7(4.1)	NA	4.6(59.4)	4.1(13.1)	16.4	11.7	NA	74.5	29.6

Table 6: Performance of subset selection methods on real datasets. The results are averages of leave-one-out (LOO) testing. The selection rules are AICc-hdf for BOSS, AICc-ndf for LASSO and 10-fold CV for the rest, respectively. The intercept term is always fitted and is not counted in the number of predictors. Minimal values for RMSE and running time for each dataset are given in bold face.

Dataset (n, p)	Metrics	BOSS	BS	FS	LASSO	Sparsenet
Housing (506, 13)	RMSE	3.372	3.37	3.37	**3.355**	3.369
	# predictors	11.004	11.006	11.02	11.063	11
	running time (s)	**0.021**	0.107	0.156	0.045	0.259
Hitters (263, 19)	RMSE	**233.853**	235.261	235.473	234.428	242.623
	# predictors	10.152	9.897	9.741	13.449	11.62
	running time (s)	**0.0138**	0.175	0.104	0.058	0.372
Auto (392, 6)	RMSE	**2.628**	**2.628**	**2.628**	2.646	2.631
	# predictors	2	2	2	4.11	2.031
	running time (s)	**0.007**	0.047	0.067	0.038	0.16
College (777, 17)	RMSE	**1565.476**	1566.776	1567.717	1566.246	1573.437
	# predictors	16.99	15.394	15.153	15.066	14.425
	running time (s)	**0.05**	0.169	0.451	**0.05**	0.473
Forest Fires (517, 55)	RMSE	**18.603**	18.683	19.041	18.946	19.115
	# predictors	0	0.422	0.43	5.563	7.081
	running time (s)	**0.084**	8020	0.593	0.121	0.785

Appendix

A Proof of theorem 1 and its corollary

In this section, we assume an orthogonal X and a null true model. This is the only scenario under which both $\mathrm{df}_C(k)$ and $\mathrm{hdf}(k)$ have analytical expressions. We will prove that the ratio of $\mathrm{df}_C(k)$ and $\mathrm{hdf}(k)$ goes to 1 as $k, p \to \infty$ while $k = \lfloor xp \rfloor$, where $\lfloor \cdot \rfloor$ denotes the greatest integer function and $x \in (0, 1)$. We start by laying out a few lemmas to be used in the proof of the main theorem.

Lemma 1. *Assume the design matrix is orthogonal and the true model is null ($\mu = 0$). Then*

$$hdf(k) = df_L(\lambda_k^\star) = k - 2p \cdot \Phi^{-1}(\frac{k}{2p}) \cdot \phi\left[\Phi^{-1}(\frac{k}{2p})\right]. \tag{16}$$

Proof. We follow the steps described in algorithm 1. We first find $\lambda_k^\star$ from (8), by using the fact that $\mu = 0$, and we get $-\dfrac{\sqrt{2\lambda_k^\star}}{\sigma} = \Phi^{-1}(\dfrac{k}{2p})$, which is then substituted into (7) to get (16). □

Lemma 2. *Define $\tilde{G}(x) = x - \Phi^{-1}(x) \cdot \phi\left[\Phi^{-1}(x)\right]$, where $x \in (0, 1)$ is a continuous variable. We have*

$$\lim_{x \to 0} \tilde{G}(x) = 0,$$

and

$$\tilde{G}'(x) = \left[\Phi^{-1}(x)\right]^2.$$

Therefore by the fundamental theorem of calculus:

$$\tilde{G}(x) = \int_0^x \left[\Phi^{-1}(u)\right]^2 du.$$

Proof. First note that, since $\phi'(v) = -v \cdot \phi(v)$ and $\lim_{v \to \pm\infty} \phi'(v) = 0$, we have

$$\lim_{v \to \pm\infty} v \cdot \phi(v) = 0.$$

Let $v = \Phi^{-1}(x)$. Then

$$\lim_{x\to 0} \tilde{G}(x) = \lim_{v\to -\infty} -v \cdot \phi(v) = 0,$$

Next, we work out the derivative of $\tilde{G}(x)$. Since $\Phi'(x) = \phi(x)$, and by the property of inverse functions, we have

$$\left[\Phi^{-1}(x)\right]' = \frac{1}{\Phi'\left[\Phi^{-1}(x)\right]} = \frac{1}{\phi\left[\Phi^{-1}(x)\right]}. \tag{17}$$

Also since $\phi'(x) = -x \cdot \phi(x)$, we have

$$\phi'\left[\Phi^{-1}(x)\right] = -\Phi^{-1}(x) \cdot \phi\left[\Phi^{-1}(x)\right] \cdot \left[\Phi^{-1}(x)\right]' = -\Phi^{-1}(x). \tag{18}$$

By (17) and (18), we have

$$\tilde{G}'(x) = 1 - \left[\Phi^{-1}(x)\right]' \cdot \phi\left[\Phi^{-1}(x)\right] - \left[\Phi^{-1}(x)\right] \cdot \phi'\left[\Phi^{-1}(x)\right] = \left[\Phi^{-1}(x)\right]^2.$$

Therefore, by the fundamental theorem of calculus, we have

$$\tilde{G}(x) = \int_0^x \tilde{G}'(u)du + \tilde{G}(0) = \int_0^x \left[\Phi^{-1}(u)\right]^2 du.$$

□

Lemma 3. *Denote $\tilde{Q}$ as the quantile function of a χ^2_1 distribution, and let $\tilde{H}(s) = -\tilde{Q}(1-s)$ where $s \in (0,1)$. For $0 \le s \le t \le 1$, consider the truncated variance function:*

$$\tilde{\sigma}^2(s,t) = \int_s^t \int_s^t (u \wedge v - uv) d\tilde{H}(u) d\tilde{H}(v), \tag{19}$$

where $u \wedge v = \min(u, v)$. We have

$$0 \le \tilde{\sigma}^2(s,t) \le 1.$$

Proof. First note that the following facts:

$$\tilde{H}(s) = -\left[\Phi^{-1}(1 - \frac{s}{2})\right]^2 = -\left[\Phi^{-1}(\frac{s}{2})\right]^2, \tag{20}$$

$$d\tilde{H}(s) = \frac{\Phi^{-1}(1-s/2)}{\phi\left[\Phi^{-1}(1-s/2)\right]} ds = -\frac{\Phi^{-1}(s/2)}{\phi\left[\Phi^{-1}(s/2)\right]} ds, \quad \text{by (17)}, \tag{21}$$

$$\Phi^{-1}(w) = -\sqrt{\log\frac{1}{w^2} - \log\log\frac{1}{w^2} - \log(2\pi)} + o(1), \quad \text{for small } w, \text{ by Fung and Seneta (2017).} \tag{22}$$

Hence for small w,

$$\left[\Phi^{-1}(w)\right]^2 = O(\log \frac{1}{w^2}). \tag{23}$$

Then by (20) and (23), we have:

$$\lim_{s\to 0} s \cdot \tilde{H}(s) = \lim_{s\to 0} -s \cdot \left[\Phi^{-1}(\frac{s}{2})\right]^2 = 0. \tag{24}$$

Also, by (20) and lemma 2,

$$-\int_0^x \tilde{H}(s)ds = 2 \cdot \tilde{G}(\frac{x}{2}). \tag{25}$$

Since $u, v \in [0,1]$, we have $u \wedge v - uv \geq 0$. By (21), it's easy to see that $d\tilde{H}(s)/ds \geq 0$. Therefore, the integrand in (19) is non-negative, so that:

$$\tilde{\sigma}^2(s,t) \geq 0,$$

and

$$\begin{aligned}
\tilde{\sigma}^2(s,t) &\leq \int_0^1 \int_0^1 (u \wedge v - uv) d\tilde{H}(u) d\tilde{H}(v), \\
&= \int_0^1 \left[\int_0^v u(1-v) d\tilde{H}(u) + \int_v^1 v(1-u) d\tilde{H}(u) \right] d\tilde{H}(v), \\
&= \int_0^1 \left[\int_0^v u d\tilde{H}(u) + v \int_v^1 d\tilde{H}(u) - v \int_0^1 u d\tilde{H}(u) \right] d\tilde{H}(v).
\end{aligned}$$

Denote

$$\tilde{M}(v) = \int_0^v u d\tilde{H}(u) + v \int_v^1 d\tilde{H}(u) - v \int_0^1 u d\tilde{H}(u).$$

Now, let's work out the three integrals in $\tilde{M}(v)$. First note that:

$$\begin{aligned}
\int_0^x u d\tilde{H}(u) &= u \cdot \tilde{H}(u)\Big|_0^x - \int_0^x \tilde{H}(u) du, \\
&= x \cdot \tilde{H}(x) - \int_0^x \tilde{H}(u) du, \quad \text{by (24)} \\
&= x \cdot \tilde{H}(x) + 2 \cdot \tilde{G}(x/2), \quad \text{by (25)}.
\end{aligned}$$

And since it's easy to see that $\tilde{H}(1) = 0$ and $\tilde{G}(1/2) = 1/2$, we have:

$$\int_0^1 u d\tilde{H}(u) = 2 \cdot \tilde{G}(1/2) = 1,$$

and

$$v\int_v^1 d\tilde{H}(u) = -v\cdot\tilde{H}(v).$$

Therefore,

$$\begin{aligned}\tilde{M}(v) &= v\cdot\tilde{H}(v) + 2\cdot\tilde{G}(v/2) - v\cdot\tilde{H}(v) - 2v\cdot\tilde{G}(1/2)\\ &= 2\cdot\tilde{G}(v/2) - v.\end{aligned}$$

Finally,

$$\begin{aligned}\int_0^1 \tilde{M}(v)d\tilde{H}(v) &= \int_0^1 2\cdot\tilde{G}(v/2)d\tilde{H}(v) - \int_0^1 v d\tilde{H}(v),\\ &= -\int_0^1 \Phi^{-1}(\frac{v}{2})\cdot\phi\left[\Phi^{-1}(\frac{v}{2})\right]d\tilde{H}(v), \quad \text{by the definition of } \tilde{G}(x),\\ &= 2\int_0^{1/2}\left[\Phi^{-1}(v)\right]^2 dv, \quad \text{by (21)},\\ &= 2\cdot\tilde{G}(1/2),\\ &= 1.\end{aligned}$$

Therefore,

$$0 \le \tilde{\sigma}^2(s,t) \le 1.$$

□

Theorem 3. *Assume the design matrix is orthogonal and the true model is null ($\mu = 0$). Let $\tilde{X}_{(i)}$ be the i-th largest order statistic in an i.i.d sample of size p from a χ^2_1 distribution. Denote $\tilde{Y}_p = \tilde{\sigma}_p^{-1}(\sum_{i=1}^k \tilde{X}_{(i)} - \tilde{\mu}_p)$, where*

$$\tilde{\sigma}_p = \sqrt{p}\cdot\sigma(1/p, k/p),$$

and

$$\tilde{\mu}_p = -p\int_{1/p}^{k/p}\tilde{H}(u)du - \tilde{H}(\frac{1}{p}),$$

where $\sigma(s,t)$ and $\tilde{H}(x)$ are defined in Lemma 3.

As $k \to \infty$, $p \to \infty$ and $k = \lfloor px \rfloor$ with $x \in (0,1)$, we have:

$$\frac{df_C(k)}{2p} = \frac{1}{2p} E\left[\sum_{i=1}^{k} \tilde{X}_{(i)}\right] = \frac{\tilde{\sigma}_p}{2p} E(\tilde{Y}_p) + \tilde{G}(\frac{k}{2p}) + O\left(\frac{\log(p)}{p}\right), \tag{26}$$

where $\lfloor \cdot \rfloor$ denotes the greatest integer function, $\tilde{G}(x)$ is defined in Lemma 2.

Proof. The proof proceeds as follows. We first apply a result in Csorgo et al. (1991), to show that $\tilde{Y}_p = \tilde{\sigma}_p^{-1}(\sum_{i=1}^{k} \tilde{X}_{(i)} - \tilde{\mu}_p)$ converges in distribution to a standard normal. We then show how $\tilde{\mu}_p$ can be expressed in terms of function G plus a remainder term, which further leads to expression (26).

It follows from Csorgo et al. (1991) Corollary 2, that if there exist centering and normalizing constants c_p and $d_p > 0$, s.t.

$$d_p^{-1}(\tilde{X}_{(1)} - c_p) \xrightarrow{D} Y, \quad \text{where Y is the standard Gumbel distribution,} \tag{27}$$

then as $k \to \infty$, $p \to \infty$ and $k = \lfloor px \rfloor$ with $x \in (0,1)$,

$$(\sum_{i=1}^{k} \tilde{X}_{(i)} - \tilde{\mu}_p)/\tilde{\sigma}_p \xrightarrow{D} Z, \quad \text{where Z is standard normal.} \tag{28}$$

First, it follows from Embrechts et al. (2013) that (27) holds, with $c_p = 2\log(p) - \log\log(p) - \log(\pi)$ and $d_p = 2$.

Next, we have

$$\begin{aligned}
\tilde{\mu}_p &= -p \int_{1/p}^{k/p} \tilde{H}(u)du - \tilde{H}(\frac{1}{p}), \\
&= -p \int_{0}^{k/p} \tilde{H}(u)du + p \int_{0}^{1/p} \tilde{H}(u)du - \tilde{H}(\frac{1}{p}), \\
&= 2p \cdot \tilde{G}(\frac{k}{2p}) - 2p \cdot \tilde{G}(\frac{1}{2p}) + \left[\Phi^{-1}(\frac{1}{2p})\right]^2, \quad \text{by (25).}
\end{aligned}$$

Also, since

$$\begin{aligned}
\tilde{G}(\frac{1}{2p}) &= \frac{1}{2p} - \Phi^{-1}(\frac{1}{2p}) \cdot \phi\left[\Phi^{-1}(\frac{1}{2p})\right], \quad \text{by definition of } \tilde{G}(x) \text{ in Lemma 2,} \\
&= \frac{1}{2p} - \frac{1}{\sqrt{2\pi}}\Phi^{-1}(\frac{1}{2p}) \cdot \exp\left(-\frac{1}{2}\left[\Phi^{-1}(\frac{1}{2p})\right]^2\right), \\
&= \frac{1}{2p} + \frac{1}{\sqrt{2\pi}} \cdot \left(\sqrt{\log(4p^2) - \log\log(4p^2) - \log(2\pi)} + o(1)\right) \cdot \\
&\qquad \exp\left[-\frac{1}{2}\left(\log(4p^2) - \log\log(4p^2) - \log(2\pi) + o(1)\right)\right], \quad \text{by (22),} \\
&= \frac{1}{2p} + \left(\sqrt{\log(4p^2) - \log\log(4p^2) - \log(2\pi)} + o(1)\right) \cdot \frac{\sqrt{\log(4p^2)}}{2p}, \\
&= O\left(\frac{\log(p)}{p}\right).
\end{aligned}$$

Also

$$\frac{1}{2p}\left[\Phi^{-1}(\frac{1}{2p})\right]^2 = O\left(\frac{\log(p)}{p}\right), \quad \text{by (22),}$$

and hence

$$\begin{aligned}
\frac{\tilde{\mu}_p}{2p} &= \tilde{G}(\frac{k}{2p}) - \tilde{G}(\frac{1}{2p}) + \frac{1}{2p}\left[\Phi^{-1}(\frac{1}{2p})\right]^2, \\
&= \tilde{G}(\frac{k}{2p}) + O\left(\frac{\log(p)}{p}\right).
\end{aligned}$$

Therefore, it's easy to see that (26) holds, i.e.

$$\frac{\mathrm{df}_C(k)}{2p} = \frac{1}{2p}E(\sum_{i=1}^{k}\tilde{X}_{(i)}) = \frac{\tilde{\sigma}_p}{2p}E(\tilde{Y}_p) + \frac{\tilde{\mu}_p}{2p} = \frac{\tilde{\sigma}_p}{2p}E(\tilde{Y}_p) + \tilde{G}(\frac{k}{2p}) + O\left(\frac{\log(p)}{p}\right).$$

□

Corollary 3.1. *If* $\limsup|E(\tilde{Y}_p)| < \infty$*, we further have:*

$$\frac{df_C(k)}{2p} = \tilde{G}(\frac{k}{2p}) + O\left(\frac{\log(p)}{p}\right) + O\left(\frac{1}{\sqrt{p}}\right). \tag{29}$$

Proof. By Lemma 3 we have $0 \le \sigma(1/p, k/p) \le 1$, and hence $\tilde{\sigma}_p = O(\sqrt{p})$. Therefore by theorem 3, we have

$$\frac{\mathrm{df}_C(k)}{2p} = \tilde{G}(\frac{k}{2p}) + O\left(\frac{\log(p)}{p}\right) + O\left(\frac{1}{\sqrt{p}}\right).$$

□

Theorem 1. *Assume X is orthogonal and the true model is null ($\mu = 0$). As $p \to \infty$, $k \to \infty$ with $k = \lfloor px \rfloor$, we have*

$$\frac{1}{2p} hdf(k) = \frac{1}{2p} df_C(k) - \frac{\tilde{\sigma}_p}{2p} E(\tilde{Y}_p) + O\left(\frac{\log(p)}{p}\right), \tag{9}$$

where $x \in (0,1)$ is a constant and $\lfloor \cdot \rfloor$ denotes the greatest integer function.

Proof. By Lemma 1, we have

$$\mathrm{hdf}(k) = df_L(\tilde{M}^{-1}(k)) = k - 2p \cdot \Phi^{-1}(\frac{k}{2p}) \cdot \phi\left[\Phi^{-1}(\frac{k}{2p})\right].$$

Then by the definition of $\tilde{G}(x)$ in Lemma 2,

$$\frac{1}{2p}\mathrm{hdf}(k) = \tilde{G}(\frac{k}{2p}).$$

By Theorem 3, we also have

$$\frac{1}{2p}\mathrm{df}_C(k) = \frac{\sigma_p}{2p} E(\tilde{Y}_p) + \tilde{G}(\frac{k}{2p}) + O\left(\frac{\log(p)}{p}\right).$$

Therefore, (9) holds, i.e.

$$\frac{1}{2p}\mathrm{hdf}(k) = \frac{1}{2p}\mathrm{df}_C(k) - \frac{\tilde{\sigma}_p}{2p} E(\tilde{Y}_p) + O\left(\frac{\log(p)}{p}\right).$$

□

Corollary 1.1. *If $\limsup |E(\tilde{Y}_p)| < \infty$, we further have*

$$\frac{df_C(k)}{hdf(k)} \to 1. \tag{10}$$

Proof. By Theorem 1 and Corollary 3.1, we know

$$\frac{1}{2p}\mathrm{hdf}(k) = \frac{1}{2p}\mathrm{df}_C(k) + O\left(\frac{1}{\sqrt{p}}\right) + O\left(\frac{\log(p)}{p}\right).$$

From Lemma 2, we know that $\tilde{G}(x)$ is a non-decreasing function with $\tilde{G}(0+) = 0$ and $\tilde{G}(1/2) = 1/2$. Thus, it's easy to see that

$$\frac{2p}{\mathrm{hdf}(k)} = \frac{1}{\tilde{G}(\frac{k}{2p})} = O(1),$$

since $k = \lfloor px \rfloor$ and $x \in (0, 1)$. Therefore,

$$\frac{\mathrm{df}_C(k)}{\mathrm{hdf}(k)} = 1 + O\left(\frac{1}{\sqrt{p}}\right) + O\left(\frac{\log(p)}{p}\right),$$

and hence

$$\frac{\mathrm{df}_C(k)}{\mathrm{hdf}(k)} \to 1.$$

□

B Expected KL-based optimism, in the context of BS

In this section, we work out the expected KullbackLeibler (KL) based optimism for BS at size k. Let's first consider fitting least squares regression on k prefixed predictors. Recall that we have the true model:

$$y = \mu + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. Use the deviance to measure the predictive error, that is

$$\Theta = -2 \log f(y|\mu, \sigma^2).$$

The training error is then

$$\mathrm{err}_{\mathrm{KL}} = -2 \log f(y|\hat{\mu}, \hat{\sigma}^2),$$

and the testing error (KL information) is

$$\mathrm{Err}_{\mathrm{KL}} = -2E_0 \left[\log f(y^0|\hat{\mu}, \hat{\sigma}^2)\right],$$

where $\hat{\mu}$ and $\hat{\sigma}^2$ are MLE based on training data (X, y), y^0 is independent and has the same distribution of y and E_0 is the expectation over y^0.

Due to the assumption of normality, the deviance can be expressed as:

$$\Theta = n \log(2\pi\sigma^2) + \frac{\|y - \mu\|_2^2}{\sigma^2}. \tag{30}$$

Maximizing the likelihood, or minimizing the deviance (30), gives:

$$\begin{aligned} \hat{\mu} &= \arg\min_{\mu} \|y-\mu\|_2^2, \\ \hat{\sigma}^2 &= \frac{1}{n}\|y-\hat{\mu}\|_2^2. \end{aligned} \tag{31}$$

Using these expressions, we then have

$$\mathrm{err}_{\mathrm{KL}} = n\log(2\pi\hat{\sigma}^2) + n, \tag{32}$$

and

$$\mathrm{Err}_{\mathrm{KL}} = n\log(2\pi\hat{\sigma}^2) + n\frac{\sigma^2}{\hat{\sigma}^2} + \frac{\|\mu-\hat{\mu}\|_2^2}{\hat{\sigma}^2}.$$

The expected optimism is then

$$\begin{aligned} E(\mathrm{op}_{\mathrm{KL}}) &= E(\mathrm{Err}_{\mathrm{KL}}) - E(\mathrm{err}_{\mathrm{KL}}), \\ &= E(n\frac{\sigma_0^2}{\hat{\sigma}^2}) + E(\frac{\|\mu-h(\hat{\beta})\|_2^2}{\hat{\sigma}^2}) - n. \end{aligned} \tag{33}$$

So far we've been considering a subset with k fixed predictors. At subset size k, BS chooses the one with minimum residual sum of squares (RSS) from all $\binom{p}{k}$ possible subsets. In order for the above derivation to hold, we need to show that the MLE we get from (31) is also the BS fit. This can be easily obtained from the full likelihood (-2 times) (32), which by plugging the expression of $\hat{\sigma}$ leads to

$$n\log(\frac{2\pi}{n}\|y-\hat{\mu}\|_2^2) + n.$$

Therefore, for all $\binom{p}{k}$ models with size k, the one with largest log likelihood, is also the one with smallest RSS. Hence (33) holds for BS fit at subset size k as well.

C Proof of Thoerem 2

Proof. Since $[X_1, X_2, \cdots, X_j]$ and $[Q_1, Q_2, \cdots, Q_j]$ span the same space, we have

$$\hat{\alpha}^{(j)} = \hat{\beta}^{(j)}. \tag{34}$$

We can express $\hat{\gamma}(k)$ as

$$\hat{\gamma}(k) = \sum_{j \in S_k} \hat{\gamma}^{(j)} - \hat{\gamma}^{(j-1)}. \tag{35}$$

We multiply both sides by R^{-1} (X is assumed to have full column rank), and use (34) to get

$$\hat{\beta}(k) = \sum_{j \in S_k} \hat{\alpha}^{(j)} - \hat{\alpha}^{(j-1)}.$$

□

References

Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. P. F. Csaki (Ed.), *2nd International Symposium on Information Theory*, Budapest, Hungary, pp. 267–281. Akadmiai Kiad.

Bertsimas, D., A. King, R. Mazumder, et al. (2016). Best subset selection via a modern optimization lens. *The Annals of Statistics 44*(2), 813–852.

Cortez, P. and A. d. J. R. Morais (2007). A data mining approach to predict forest fires using meteorological data.

Csorgo, S., E. Haeusler, and D. M. Mason (1991). The asymptotic distribution of extreme sums. *The Annals of Probability*, 783–811.

Efron, B. (1986). How biased is the apparent error rate of a prediction rule? *Journal of the American Statistical Association 81*(394), 461–470.

Efron, B. (2004). The estimation of prediction error: covariance penalties and cross-validation. *Journal of the American Statistical Association 99*(467), 619–632.

Efroymson, M. (1960). Multiple regression analysis. *Mathematical methods for digital computers*, 191–203.

Embrechts, P., C. Klüppelberg, and T. Mikosch (2013). *Modelling extremal events: for insurance and finance*, Volume 33. Springer Science & Business Media.

Flynn, C. J., C. M. Hurvich, and J. S. Simonoff (2013). Efficiency for regularization parameter selection in penalized likelihood estimation of misspecified models. *Journal of the American Statistical Association 108*(503), 1031–1043.

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

Fung, T. and E. Seneta (2017). Quantile function expansion using regularly varying functions. *Methodology and Computing in Applied Probability*, 1–13.

Furnival, G. M. and R. W. Wilson (1974). Regressions by leaps and bounds. *Technometrics 16*(4), 499–511.

Hammarling, S. and C. Lucas (2008). Updating the qr factorization and the least squares problem.

Harris, X. T. (2016). Prediction error after model search. *arXiv preprint arXiv:1610.06107*.

Hastie, T., R. Tibshirani, and R. J. Tibshirani (2017). Extended comparisons of best subset selection, forward stepwise selection, and the lasso. *arXiv preprint arXiv:1707.08692*.

Hocking, R. and R. Leslie (1967). Selection of the best subset in regression analysis. *Technometrics 9*(4), 531–540.

Hurvich, C. M. and C.-L. Tsai (1989). Regression and time series model selection in small samples. *Biometrika 76*(2), 297–307.

Hurvich, C. M. and C.-L. Tsai (1991). Bias of the corrected AIC criterion for underfitted regression and time series models. *Biometrika 78*(3), 499–509.

Janson, L., W. Fithian, and T. J. Hastie (2015). Effective degrees of freedom: a flawed metaphor. *Biometrika 102*(2), 479–485.

Konishi, S. and G. Kitagawa (2008). *Information criteria and statistical modeling*. Springer Science & Business Media.

Liao, J., J. E. Cavanaugh, and T. L. McMurry (2018). Extending AIC to best subset regression. *Computational Statistics 33*(2), 787–806.

Mallows, C. L. (1973). Some comments on Cp. *Technometrics 15*(4), 661–675.

Mazumder, R., J. H. Friedman, and T. Hastie (2011). Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association 106*(495), 1125–1138.

Meinshausen, N. (2007). Relaxed lasso. *Computational Statistics & Data Analysis 52*(1), 374–393.

Meinshausen, N. (2012). *relaxo: Relaxed Lasso*. R package version 0.1-2.

Natarajan, B. K. (1995). Sparse approximate solutions to linear systems. *SIAM Journal on Computing 24*(2), 227–234.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics 6*(2), 461–464.

Taddy, M. (2017). One-step estimator paths for concave regularization. *Journal of Computational and Graphical Statistics 26*(3), 525–536.

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

Tibshirani, R. J. (2015). Degrees of freedom and model search. *Statistica Sinica*, 1265–1296.

Ye, J. (1998). On measuring and correcting the effects of data mining and model selection. *Journal of the American Statistical Association 93*(441), 120–131.