Adaptive Forward Stepwise: A Method for High Sparsity Regression

Ivy Zhang   IVYZHANG@STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305, USA

Robert Tibshirani   TIBS@STANFORD.EDU
Department of Statistics and Biomedical Data Science
Stanford University
Stanford, CA 94305, USA

Editor: Han Liu

Abstract

This paper proposes a sparse regression method that continuously interpolates between Forward Stepwise selection (FS) and the LASSO. When tuned appropriately, our solutions are much sparser than typical LASSO fits but, unlike FS fits, benefit from the stabilizing effect of shrinkage. Our method, *Adaptive Forward Stepwise Regression* (AFS) addresses the need for sparser models with shrinkage. We show its connection with boosting via a soft-thresholding viewpoint and demonstrate the ease of adapting the method to classification tasks. In both simulations and real data, our method has lower mean squared error and fewer selected features across multiple settings compared to popular sparse modeling procedures.

Keywords: sparse regression, LASSO, feature selection, high dimensionality, shrinkage

1. Introduction

We consider the problem of sparse regression under linear models. The usual linear regression setting is given by

$$y = X\beta + \epsilon, \tag{1}$$

for a response, $y \in \mathbb{R}^n$, features $X \in \mathbb{R}^{n \times p}$, and random error ϵ. Without loss of generality, assume the data are centered so we do not include an intercept. We also assume the coefficient vector, β, is sparse, i.e., $\|\beta\|_0 = \sum_{k=1}^{p} \mathbb{1}\{\beta_k \neq 0\}$ is small. Here, the ℓ_0 pseudo-norm of β being small tells us that only a small subset of the p predictors are relevant. This assumption naturally motivates the Best Subset (BS) objective,

$$\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|^2 + \lambda \|\beta\|_0, \tag{2}$$

which aims to optimally select a subset of features by introducing a ℓ_0 penalty constraint to the ordinary least squares (OLS) objective. However, solving such a non-convex problem is NP-hard, leading to a preference for alternative methods (Natarajan, 1995).

1.1 Relevant Background

One of the most popular methods for solving such a problem is the LASSO. This method minimizes the ℓ_1 penalty objective

$$\min_{\beta \in \mathbb{R}^p} \|y - X\beta\|^2 + \lambda\|\beta\|_1, \tag{3}$$

whose convex formulation makes it more computationally attractive (Tibshirani, 1996; Chen et al., 1998). Additionally, under certain regularity conditions, the LASSO is model selection consistent (ie, recovers the true support as $n \to \infty$) and has good prediction accuracy (Zhao and Yu, 2006; Dalalyan et al., 2017). A closely related approach, *Least Angle Regression* (LAR), approximates the LASSO solution via a sequential procedure (Efron et al., 2004). In fact, LAR returns the LASSO solution exactly when coefficient estimates do not change signs along the coefficient path. LAR begins by setting all coefficients $\hat{\beta}^{LAR}$ equal to 0 and initializing an empty active set, $\mathcal{A} = \emptyset$. At each step, LAR fits a model on the data subsetted for the selected variables from the previous step. It then selects the predictor most correlated with the residual. LAR then moves the coefficient for that predictor towards its OLS coefficient until another predictor achieves equal correlation with the current residual. Finally, it repeats the variable selection procedure and coefficient estimate in the subsequent steps until it computes the full coefficient path like the LASSO.

A drawback of the LASSO is its tendency to select excess noise variables after cross-validation of λ, leading to denser models especially in medium to high signal-to-noise ratio (SNR) settings. This behavior often arises because of the ℓ_1 penalty's simultaneous role of shrinkage and model selection, especially when regularity conditions are not met (Mazumder et al., 2011; Bühlmann and Hothorn, 2010; Chetverikov et al., 2021). A classical alternative to this is Forward Stepwise selection (FS). In one variant, FS starts with all coefficients set to zero and an empty active set, as in LAR. It then uses the same selection criterion as well. Unlike LAR, FS produces coefficients that are the full OLS estimates fit on the selected features. While FS often yields sparser models when appropriately tuned, the absence of shrinkage results in higher variance coefficient estimates. Additionally, FS tends to perform less accurately than the LASSO in low SNR (Hastie et al., 2020).

1.2 Our method

This paper proposes Adaptive Forward Stepwise Regression (AFS), a data-adaptive sparse regression method. We show that it bridges FS and LASSO, achieving the strengths of both when properly tuned. Specifically, we show its convergence to LAR when the hyperparameter tends to 0. We also derive an approximate soft-thresholding estimator under orthogonal design to show its connection to boosting. Empirically, we show that AFS's performance is more robust across a range of SNR, variable correlations, and dimensions, while also being more computationally efficient than several popular sparse modeling methods. In both simulations and real data, AFS also achieves higher sparsity than LASSO. Additionally, we demonstrate its application on real datasets and show that it can be easily adapted to classification tasks.

1.3 Motivating example

To illustrate the intuition behind our proposed method, we present three examples simulating a high-sparsity environment. The samples, $X_1, ..., X_n$, and noise, $\epsilon_1, ..., \epsilon_n$ are both independently drawn from Gaussian distributions as follows:

$$X_1, ..., X_n \overset{\text{i.i.d}}{\sim} N(0, \Sigma), \ X_i \in \mathbb{R}^p$$
$$y_i = X_i \beta + \epsilon_i, \ \epsilon_i \sim N(0, \sigma_\epsilon^2)$$

Each covariate has the same variance and covariance, and only the first five covariates have non-zero coefficients. Specifically, $\forall k = 1, ..., p, k \neq j$, we set:

$$\Sigma_{k,k} = \sigma_X^2, \ \Sigma_{j,k} = s_X$$
$$\beta_1, ..., \beta_5 = 2, \ \beta_6, ..., \beta_p = 0$$

The first example has the settings (1) $n = 100$, $p = 120$, (2) high SNR of 4.42 and (3) low correlation of 0.06 between covariates. Figure 1 compares the coefficient paths of FS, LASSO, and AFS as a function of the ℓ_1 norm. For FS and AFS, each knot represents one additional step taken in the respective algorithm. For the LASSO, each knot represents the next λ in the λ-sequence evaluated. As each method progresses in their iterations/λ-sequence, the estimated model becomes more dense as seen by the entry of new coefficient path line segments. We use 10-fold cross-validation (CV) to select the hyperparameter for each method: number of steps for FS, ρ and number of steps for AFS, and λ for the LASSO. The vertical dashed line in Figure 1 indicates the CV solution along the path while the horizontal dotted line marks the true coefficient of the non-zero β's.
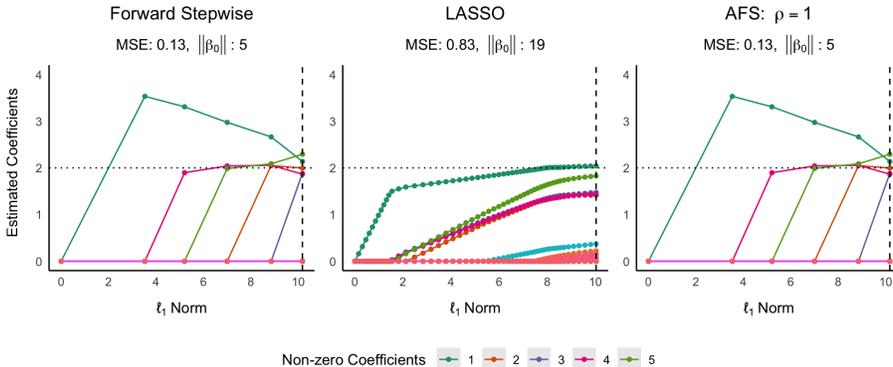

Figure 1: SNR of 4.42, low correlation of 0.06 between covariates, and $n \ll p$. The example illustrates the coefficient path of FS, LASSO, and AFS as a function of their ℓ_1 norm. Paths of $\hat{\beta}_1, ..., \hat{\beta}_5$ are annotated. Mean squared error, $\|X\hat{\beta} - \mu\|_2^2$, is notably lower for FS and AFS estimates compared to the LASSO. Both FS and AFS resulted in a sparser final model with $\|\beta_0\| = 5$ than the LASSO, with $\|\beta_0\| = 19$.

In this high SNR, low correlation example, the CV solution identifies the true support for all three methods, but the LASSO introduces many more false positives with $\|\beta_0\| = 19$, driving up the mean squared error (MSE) $\|X\hat{\beta} - \mu\|_2^2$. Furthermore, $\hat{\beta}_1^{LASSO}, ..., \hat{\beta}_5^{LASSO}$ are further from the true β values than FS. FS clearly performs better, with no false positives and all estimated coefficients close to 2. The CV FS solution recovers the true support more accurately, with coefficient estimates much nearer to the true β than the LASSO. Note that the CV solution for AFS matches that of FS. The second example illustrates the opposite end of the spectrum, with (1) $n = 120$, $p = 100$, (2) SNR of 2.78, and (3) high correlation of 0.56 between covariates where the LASSO outperforms FS. In this more challenging scenario with approximately half the SNR of Example 1 and high correlation, FS struggles to recover the full true support. As shown in Figure 2, the LASSO recovers more of the true support and consequently achieves a significantly lower mean squared error, $\|X\hat{\beta} - \mu\|_2^2$, for its coefficient estimates. Unlike in the first example, the AFS CV solution resembles that of the LASSO more than FS. Here, AFS attains a similar MSE to the LASSO, with a much smaller model.
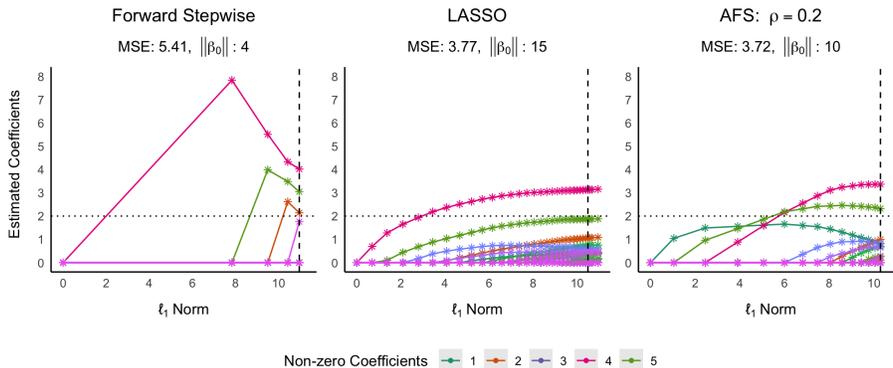

Figure 2: SNR of 2.78, high correlation of 0.56 between covariates, and $n \gg p$ example illustrating the coefficient paths of FS, LASSO, and AFS as a function of their ℓ_1 norm, with the layout as in Figure 1. The LASSO and AFS recover more of the true support than FS, while FS tends to overshoot the coefficient estimates. Both AFS and the LASSO benefit from shrinkage.

In the final example, we consider an intermediate regime with medium correlation. Here, we would ideally hope to achieve support recovery similar to the LASSO's, but with fewer false positives. We would hope to also estimate non-zero coefficients closer to those in FS. In this example, (1) $n = 100$, $p = 120$, (2) SNR of 2.59, and (3) medium correlation of 0.2. Figure 3 shows that AFS achieves this best of both worlds, resulting in the lowest MSE of the three with a model size less than half of the LASSO's. From the three examples, we see that AFS achieves robustness by adaptively constructing a model that resembles the clear winner between FS and the LASSO. Furthermore, it improves upon the LASSO by producing a sparser model while also improving upon FS by recovering more of the true support and applying shrinkage.
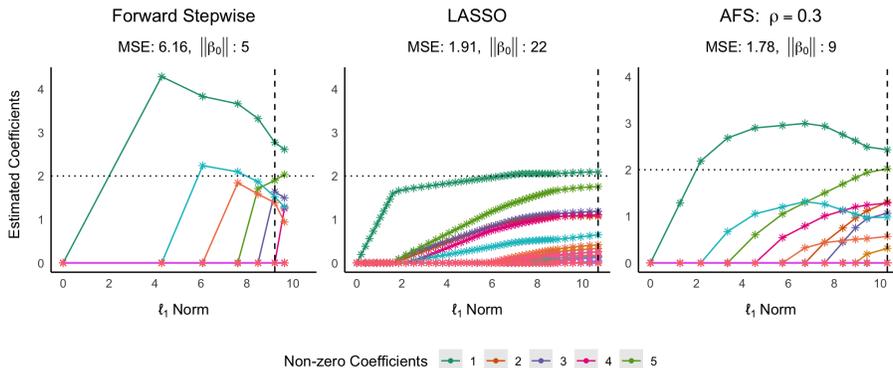
Figure 3: SNR of 2.59, medium correlation of 0.2 between covariates, and $n \ll p$. Unlike in Figure 1 where the AFS CV coefficient path matches that of FS or in Figure 2 where it is similar to that of the LASSO, this example shows an AFS coefficient path that represents a middle ground between FS and the LASSO.

1.4 Related Work

Recently, there has been ongoing development of penalty-based methods to achieve sparser models. For instance, the Relaxed LASSO (RLASSO) addresses some limitations of the LASSO by decoupling model selection from shrinkage (Meinshausen, 2007). This induces sparser estimates and improves prediction accuracy particularly in high SNR, while approximating the LASSO solution in low SNR scenarios. Other approaches explore non-convex optimization methods (Fan and Li, 2001; Zhang, 2010). For example, SparseNet uses a non-convex penalty that transitions between ℓ_1 and ℓ_0 regularization, bridging the LASSO and BS. In simulations, SparseNet yields similar or better predictive accuracy with fewer selected variables. However, unlike the LASSO, SparseNet does not readily generalize to non-Gaussian error models. See Mazumder et al. (2011) for details.

Others approached the problem through iterative methods like boosting, synthetic data techniques, or independence learning. Boosting, as noted by Bühlmann and Hothorn (2007), acts as a regularization method for model estimation. Efron et al. (2004) highlighted the connection between the LASSO and Friedman (2001)'s least squares boosting algorithm. Bühlmann (2006) introduced L_2Boost, a variant aimed at consistently recovering the true regression function in high-dimensional and sparse settings. Meanwhile, Hédou et al. (2024) proposed the Stabl algorithm, which applies a base regularization model, such as the LASSO, on both original and synthetic data samples. This method selects features that meet a "reliability threshold" intended to reduce the number of false positive features selected. Another iterative method is the iterative sure independence screening (ISIS), which updates a set of important variables, selected based on marginal utility, conditional on the previous step (Fan and Lv, 2008).

1.5 Outline of the paper

In Section 2.1, we formally introduce AFS. Next, we prove the connection of our method to LAR, LASSO, and FS in Section 2.3. Additionally, we discuss the relationship between AFS and boosting, presenting AFS as a soft-thresholding procedure under orthogonal design in Section 2.4. Finally, we compare the performance of our method to other popular sparse regression models across various simulations and real data examples in Sections 3 and 4. At the end of Section 4, we outline an adaptation of the algorithm for any generalized linear model. We conclude with a summary of our contributions and a discussion in Section 5.

2. Adaptive Forward Stepwise Regression

2.1 The AFS algorithm

In this section, we introduce our method, Adaptive Forward Stepwise, to address the need for sparser models. Consider the linear regression setting for a response, y:

$$y = X\beta + \epsilon, \ \epsilon \sim N(0, \sigma^2 I), \tag{4}$$

with $y \in \mathbb{R}^n, X \in \mathbb{R}^{nxp}$. Without loss of generality, assume the data are centered so we do not include an intercept. Suppose β is sparse, i.e., the ℓ_0 pseudo-norm of β, $\|\beta\|_0 = \sum_{k=1}^{p} \mathbb{1}\{\beta_k \neq 0\}$ is small. Since this tells us only a small number of variables are relevant, our goal is to estimate a sparse model.

Our proposed iterative method starts at step $m = 0$ with setting all estimated coefficients,

$$\hat{\beta}_{0,\rho}^{AFS} = 0,$$

and an empty active set, $|\mathcal{A}| = 0$, representing no selected variables. We also fix our shrinkage parameter, $\rho \in (0, 1]$, and maximum number of steps, M, to iterate on.

At each step, m, we choose the jth variable which maximizes the inner product between the vector, x_j, and the residual from the estimated coefficients.

$$j_m^* = \underset{j \in \{1,..,p\}}{\operatorname{argmax}} |x_j^\mathsf{T}(y - X\hat{\beta}_{m-1,\rho}^{AFS})|. \tag{5}$$

In the first step, $j_m^* = \operatorname{argmax}_j |x_j^\mathsf{T} y|$ since $\hat{\beta}_{0,\rho}^{AFS} = 0$. For subsequent steps, we can interpret j_m^* as the variable that maximizes the correlation with the response, after projecting out the contribution from the selected variables. Equivalently, we can think of j_m^* as the variable that minimizes the angle between the x_j and the residual, an interpretation which will help us make the connection with *Least Angle Regression* in Section 2.3.

Once the selected variable is added to our active set, we estimate the OLS coefficients using the selected subset of the data:

$$\hat{\nu}_m = (X_{\mathcal{A}_m}^\mathsf{T} X_{\mathcal{A}_m})^{-1} X_{\mathcal{A}_m}^\mathsf{T} y. \tag{6}$$

In the case when $n \ll p$, we can add an early stopping rule, which ends the algorithm when the ℓ_1 norm of the AFS coefficients exceeds the largest of the LASSO coefficient ℓ_1 norm,

i.e., $\|\hat{\beta}^{AFS}\|_1 > \max_\lambda \|\hat{\beta}^{LASSO}(\lambda)\|_1$. This ensures that Eqn. 6 is well defined. We denote this stopping threshold as $h = \max_\lambda \|\hat{\beta}^{\text{LASSO}}(\lambda)\|_1$[1].

Finally, we define the AFS coefficient as

$$\hat{\beta}^{AFS}_{m,\rho} = (1-\rho)\hat{\beta}^{AFS}_{m-1,\rho} + \rho\hat{\nu}_m, \tag{7}$$

which applies a shinkage to both the previous step's estimated coefficients and the current step's estimated OLS coefficient. We summarize our method in Algorithm 1.

Algorithm 1 produces a path of coefficients across m which can be estimated using various resampling or covariance penalty methods to minimize the true test error. However, since commonly used covariance penalty methods like Mallow's C_p, AIC, and BIC depend on the degrees of freedom (df) of the estimator, we recommend selecting ρ via cross-validation. This is for two main reasons: (1) accuracy and (2) computational efficiency. The estimator's dependence on y for covariate selection makes it a non-linear function of y, complicating the estimation of df. While bootstrap methods for estimating df are often computationally prohibitive with large datasets, the traditional df approximation as the trace of the hat matrix M, given by $My = X\hat{\beta}$, can be highly crude for large ρ. Empirical results highlighting these issues are detailed in Appendix B.1.

Algorithm 1 Adaptive Forward Stepwise

1: Initialize all p AFS coefficients $\hat{\beta}^{AFS}_{0,\rho} = 0$ and active set, $\mathcal{A} = \{\emptyset\}$.
2: For the following parameters, set
3:     M, the number of iterations, large   ▷ Choose by CV
4:     $\rho \in (0,1]$, the stepsize   ▷ Choose by CV
5:     $h = \max_\lambda \|\hat{\beta}^{\text{LASSO}}(\lambda)\|_1$, the maximum allowable ℓ_1 norm
6: While $m < M$ and $\|\hat{\beta}^{AFS}_{m,\rho}\|_1 < h$, let
7:     $j^*_m = \underset{j\in\{1,..,p\}}{\operatorname{argmax}} |x^\intercal_j(y - (\hat{\beta}^{AFS}_{m-1,\rho})^\intercal X)|$ ▷ Select most correlated variable with current residuals
8:     $\mathcal{A}_m = \mathcal{A}_{m-1} \cup j^*_m$ ▷ Update active set
9:     $\hat{\nu}_m = \hat{\beta}^{OLS}_{\mathcal{A}_m}$, the OLS coefficients of y on the active set $\mathcal{A}_m$ ▷ Compute OLS coefficients
10:     $\hat{\beta}^{AFS}_{m,\rho} = (1-\rho)\hat{\beta}^{AFS}_{m-1,\rho} + \rho\hat{\nu}_m$ ▷ Update AFS coefficients

---

Remark 1 *One advantage of AFS is that the structure of the algorithm allows us to easily conduct post-selection inference per the framework of Lee et al. (2016). Formally, we wish to conduct the following test:*

$$H_0 : v^\intercal_k \theta = 0$$

---

1. We note that using this stopping rule when $n \ll p$ requires calculating a full LASSO path, which costs $O(np(1+q)) + \sum_{k=1}^{K} O(w_k p r_k)$ using `glmnet`'s covariance update algorithm. Here, q is the number of distinct variables that ever enter the path, K is the number of λ's considered, w_k is the number of cycles it takes to converge for λ_k, and r_k is the size of the active set at λ_k. See Friedman et al. (2010) for details.

conditional on the chosen $\mathcal{A}_m$ at step m. I.e., we are testing the coefficient of the variable, k, selected at step m is 0. In the case when $p < n$, their framework allows us to perform inference on the true population β_k by defining $v_k = (X^\intercal X)^{-1} X^\intercal e_k$, where e_k is the kth standard basis vector. In the case when $p > n$, note that we can only perform inference on the submodel β i.e., $v = (X_{A_k}^\intercal X_{A_k})^{-1} X_{A_k}^\intercal e_k$. Details can be found in Appendix C.

2.2 Extension to Generalized Linear Models

Another advantage of AFS is the adaptability of the method to any generalized linear model (GLM) by modifying the selection criterion with the appropriate score function and fitting $\hat{\beta}^{GLM}$ instead of $\hat{\beta}^{OLS}$. For example, a binary classification task using a logistic regression can be adapted into the AFS framework as in Algorithm 2. Compared to Algorithm 1, only lines $7 - 9$ differ. We illustrate its application for binary classification on real data in Section 4.

Algorithm 2 Adaptive Forward Stepwise - Logit Link Modifications

---

1: Initialize all p AFS coefficients $\hat{\beta}_{0,\rho}^{AFS} = 0$ and active set, $\mathcal{A} = \{\emptyset\}$.
2: For the following parameters, set
3:     M, the number of iterations, large   ▷ Choose by CV
4:     $\rho \in (0,1]$, the stepsize   ▷ Choose by CV
5:     $h = \max\limits_{\lambda} \|\hat{\beta}^{\text{LASSO}}(\lambda)\|_1$, the maximum allowable ℓ_1 norm
6: While $m < M$ and $\|\hat{\beta}_{m,\rho}^{AFS}\|_1 < h$, let
7:     $j_m^* = \underset{j \in \{1,..,p\}}{\text{argmax}} \left| x_j^\intercal \left(y - \frac{\exp((\hat{\beta}_{m-1,\rho}^{AFS})^\intercal X)}{(1+\exp((\hat{\beta}_{m-1,\rho}^{AFS})^\intercal X))} \right) \right|$ ▷ Select most correlated variable with current residuals
8:     $A_m = A_{m-1} \cup j_m^*$ ▷ Update active set
9:     $\hat{\nu}_m = \hat{\beta}_{\mathcal{A}_m}^{Logistic}$, the logistic regression coefficients of y on the active set, using $\hat{\beta}_{\mathcal{A}_{m-1,\rho}}^{Logistic}$ as warm start
10:    $\hat{\beta}_{m,\rho}^{AFS} = (1-\rho)\hat{\beta}_{m-1,\rho}^{AFS} + \rho\hat{\nu}_m$ ▷ Update AFS coefficients

---

2.3 Connection to LAR, the LASSO, and Forward Stepwise

We now formally show that AFS interpolates between FS and LAR. For $\rho = 1$, we have the FS procedure, which sequentially adds j_m^* and fits $\hat{\beta}_m^{FS} = \hat{\beta}_{\mathcal{A}_m}^{OLS}$. For $\rho \to 0$, interestingly, we recover the LAR solution path as seen in Figure 4. The plot shows the solution path of LAR and AFS for a simulation with $X_1, .., X_7$ drawn *i.i.d* from a standard Gaussian distribution and $y = X\beta + N(0,1)$, with $\beta_1, .., \beta_5 = 1$, $\beta_6, \beta_7 = 0$. As we can see, for this small choice of $\rho = 0.05$, AFS takes many small steps and traces out a similar path to LAR.

This result, formalized in Theorem 2, relies on the the fact that both methods use the same criterion for selecting a variable to add to the active set and the piecewise linear nature of their solution paths. The details of the proof can be found in Appendix A.1.
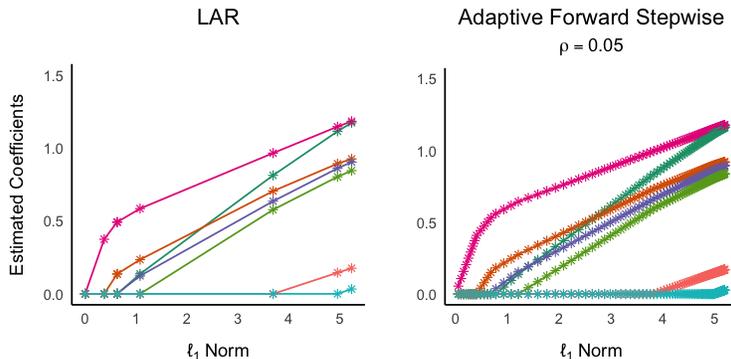
Figure 4: Coefficient paths for LAR and AFS ($\rho = 0.05$) from simulation of $n = 100, p = 7$. AFS takes many small steps to approximately trace out each line segment that make up the solution path of LAR. The AFS plot is truncated to exclude solutions with $\|\hat{\beta}^{AFS}\|_1 > 5.25$, the ℓ_1 norm of the last step of the LAR coefficient path.

Theorem 2 *Let $t = \|\hat{\beta}\|_1$ and assume there are no ties in variable selection for both procedures. Then*

$$\lim_{\rho \to 0} \hat{\beta}_\rho^{AFS}(t) = \hat{\beta}^{LAR}(t)$$

This result suggests the following modification to Algorithm 1 to approximately recover the LAR coefficients in practice: iteratively reduce ρ in each step until the selection criterion selects a new variable. Since LAR approximates the LASSO when there is no change in any non-zero coefficient sign, this also approximately recovers the LASSO. In the case of a sign change, we can add one further modification, as detailed in Appendix D.

2.4 Connection to Boosting and a Soft Thresholding Viewpoint

In this section, we provide another perspective on AFS to better understand its behavior. The fitting of a model based on residuals from the previous iteration suggests a possible connection with boosting. This connection is made clear from a soft-thresholding viewpoint. For the remainder of this section, we consider an orthogonal design, i.e. $X^\intercal X = I_p$, to gain some more intuition on our method.

At each iteration, m, the jth AFS coefficient is given by

$$\hat{\beta}_{j,\rho}^{AFS}(m) = \sum_{i=0}^{m-1} (x_j^\intercal y \rho)(1-\rho)^{i-k_j+1} = \hat{\beta}_j^{OLS}(1 - (1-\rho)^{m-k_j+1}), \qquad (8)$$

where k_j is the iteration at which x_j enters the active set and $\hat{\beta}_j^{OLS}$ is the OLS coefficient for the jth variable fit on the full design matrix. The first equality follows from the definition of $\hat{\beta}_{j,\rho}^{AFS}$ since the columns of X are orthogonal. The second equality follows from evaluating the geometric sequence. The exponent, $\ell_{j,m} := m - k_j + 1$, denotes the number of times the jth variable has been in the active set by step m, inclusive. Therefore, the AFS coefficient

9

for each variable is the OLS coefficient shrunk by a factor $1 - (1 - \rho)_{j,m}^{\ell}$, which decreases in m. Using this representation of the AFS coefficients helps us arrive at the approximate soft-thresholding estimator in Theorem 3.

Theorem 3 *Define the soft-thresholding estimator*

$$\hat{\beta}_j^{ST}(\lambda) = \begin{cases} \hat{\beta}_j^{OLS} - \lambda & \text{if } \hat{\beta}_j^{OLS} \geq \lambda \\ 0 & \text{if } |\hat{\beta}_j^{OLS}| < \lambda \\ \hat{\beta}_j^{OLS} + \lambda & \text{if } \hat{\beta}_j^{OLS} \leq -\lambda \end{cases}$$

There exists some threshold $\lambda_{j,m}(\rho) \in [c(1 - \rho)^{2\ell_{j,m}}, c(1 - \rho)^2)$ for $c = \sqrt{1 - (1 - \rho)^2}$ such that Equation 8 is well approximated by a soft-thresholding estimator with threshold $\lambda_{j,m}(\rho)$ for small ρ:

$$\lim_{\rho \to 0} |\hat{\beta}_{j,\rho}^{AFS}(m) - \hat{\beta}_j^{ST}(\lambda_{j,m}(\rho))| = 0$$

under an orthogonal X.

We now compare this to boosting with a linear estimator under squared error loss, L2Boosting. Firstly, the L2Boosting analog of the right hand side of Equation 8 is

$$\hat{\beta}_{j,m}^{Boost} = \hat{\beta}_j^{OLS}(1 - (1 - \nu)^{\tilde{k}_j})$$

for step size ν and $\tilde{k}_j$ number of times variable j has been chosen by iteration m. Since $\ell_{j,m} \geq \tilde{k}_j$, AFS will produce more shrunken coefficients when the same step sizes are chosen. Second, from the details of Theorem 3 (see Appendix A.2), we see that the residual sum of squares (RSS) for AFS is decreasing in m and the difference in RSS decreases in m by a factor of $(1 - \rho)^2$ just like L2Boosting. This further connects the behavior of the AFS and boosting algorithms, illustrated in Figure 5.
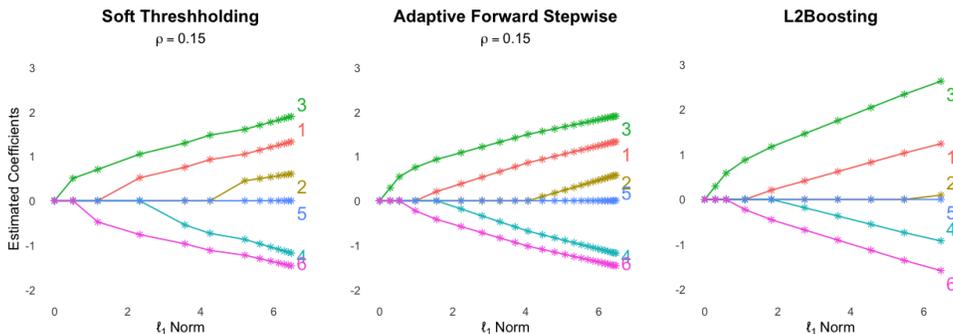

Figure 5: Coefficient paths for the approximate soft thresholding estimator 3 and AFS at $\rho = 0.15$ compared to L2Boosting. Simulation for $n = 100, p = 6$ under an orthogonal design matrix X.

3. Simulation studies

In this section, we present simulations across various SNR, correlation between covariates, and dimension settings to compare the results of AFS, FS, LASSO, RLASSO, SparseNet, and Stabl[2]. For each simulation, we draw data from a Gaussian model, apply CV to select any hyperparameters, and refit the model on the full dataset. The data is centered so we do not estimate an intercept. No test-training split was used since the true mean, $\mu = X\beta$, is known. Each simulation sets $\beta_1, ..., \beta_5 = 2$ and 0 otherwise.

3.1 Performance

We now provide detailed results to compare the performance of AFS and other sparse regression methods. We consider regimes covering combinations from the following settings: SNR of 0.5 (low), 1.0 (medium), 1.5, (medium), and 2.0 (high); correlation of 0, 0.15 (low), and 0.6 (high); $n = 100$ and $n = 120$; $p = 100$ and $p = 120$.

The low, medium, and high SNR levels used here are based on the analysis in Hastie et al. (2020) to reflect SNRs most commonly found in real data. The figures below show the median MSE, $\|X\hat{\beta} - \mu\|_2^2$ and FPR, proportion of false positive features selected, for each configuration. Error bars represent ± 1 standard deviation across 50 trials. In the $n = 120, p = 100$ setting, AFS achieves one of the lowest median MSEs, while no other method consistently demonstrates top performance across all SNR levels. This demonstrates that AFS is a more robust compared to other popular methods, while producing highly sparse models. However, we note that in the high dimensional, high correlation setting, RLASSO outperforms AFS in robustness and sparsity.

3.2 Computation time

To assess computational time, we measured the runtime required for each method to fit the full coefficient path (excluding Stabl, which lacks a coefficient path) for $n = 200$ and varying p. Figure 8 presents the average computation time over 50 trials with the following settings: (1) 0.15 correlation (2) 1.0 SNR and (3) $\beta_1, ..., \beta_5 = 2$ and 0 otherwise. To allow for better comparison, hyperparameter selection via CV was excluded from the timing. However, we note that computations for LASSO/RLASSO (`glmnet`) and FS (`bestsubset`) are based in `C`, SparseNet in `fortran`, and AFS and Stabl in `R`. AFS shows lower computational expense than most methods due to efficient matrix inverse updates at each iteration. Stabl, the least efficient method, incurs added costs from incorporating synthetic data, while the LASSO remains faster than all methods across all dataset sizes.

4. Real data examples

In this section, we compare the performance of AFS and other sparse regression methods across eight publically available datasets: prostate ($n = 97, p = 8$) (Hastie et al., 2009), diabetes ($n = 442, p = 10$) (Efron et al., 2004), wine ($n = 4898, p = 12$) (Yellow, 2021), productivity ($n = 1197, p = 23$) (Mexwell, 2021), student grades ($n = 649, p = 42$) (Cortez,

---

2. `bestsubset::fs()` and `sparsenet::sparsenet()` in R only supports the Gaussian model so we do not compare their performance in the binary classification task of Figure 11
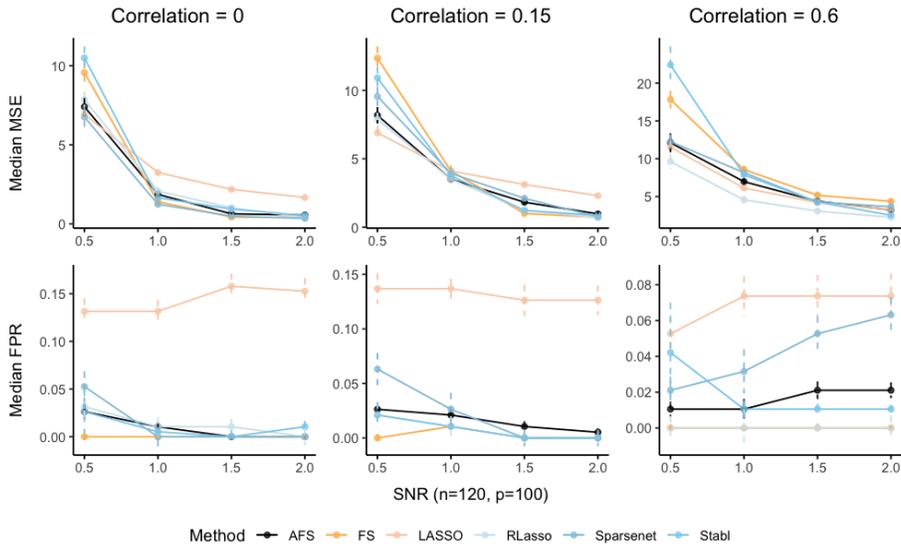
Figure 6: $n > p$ simulation results across 50 trials. Error bars represent ± 1 standard deviation. Unlike other methods, AFS performance is robust across various configurations while maintaining high sparsity.

2008), soy ($n = 320, p = 49$) (Michalski and Chilausky, 1980), energy ($n = 19735, p = 28$) (Tsanas and Xifara, 2012), nrti ($n = 1005, p = 211$) (Lockhart et al., 2014), and genome ($n = 404, p = 18580$) (Seal et al., 2020). In the first plot, Figure 9, we present MSE on a held-out test set over 50 trials of a 15-85% test-train split, relative to that of the LASSO, on a log scale. Hyperparameters for AFS, FS, SparseNet, RLASSO, and the LASSO are selected by 10-fold CV. The second plot, Figure 10, displays the number of features selected in the final models.

From these real data examples, we observe that AFS achieves comparable to or better performance than the LASSO and other methods on datasets with fewer than 50 covariates, while yielding sparser models. Some methods, such as FS, achieve higher sparsity than AFS but compromise predictive performance. AFS particularly excels on the larger nrti dataset. However, it faces more challenge in the high-dimensional, high-correlation setting of the genome dataset, aligning with simulation results in Section 3.

In Figure 11, we present an application of AFS for binary classification using four datasets from the UCI repository (Mansouri et al., 2013; Tasci et al., 2022; Malani et al., 2019; Ramana et al., 2012). For each dataset, we compare the performance of AFS, LASSO, RLASSO, and Stabl over 50 trials. A 15-85% train-test split was applied, and 10-fold CV was used to select hyperparameters for AFS, LASSO, and RLASSO on the training set. The misclassification percentage (log-scaled) on the test set and the final selected model size are reported.

Our experiments show that AFS generates significantly sparser models than the LASSO and RLASSO, while achieving comparable or improved accuracy. Although Stabl sometimes produced even sparser models than AFS, this often came at the cost of reduced accuracy.
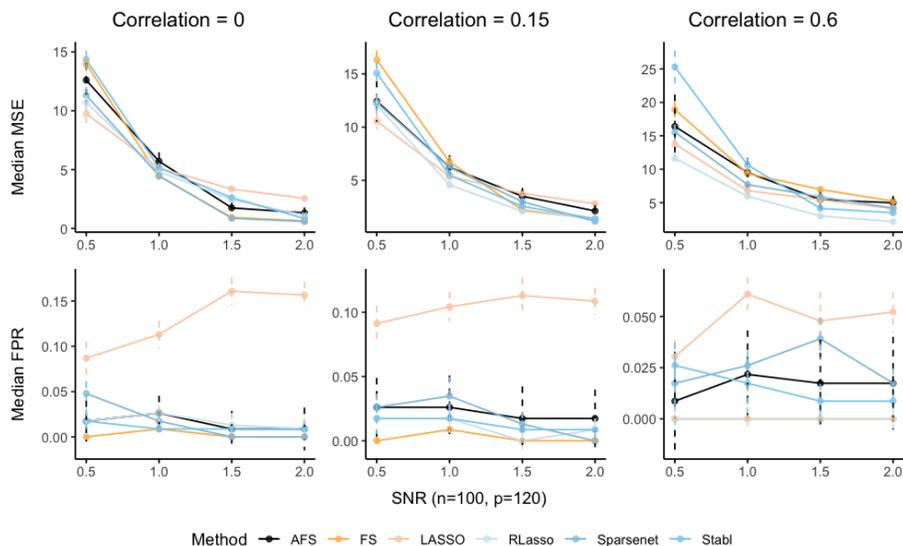
Figure 7: $n < p$ simulation results across 50 trials. As in the $n > p$ setting, AFS MSE does not fluctuate as strongly as other methods across SNR for a fixed data correlation–for example, under 0 and 0.15 correlation, the LASSO has the lowest MSE under 0.5 SNR but highest MSE under $1.5 - 2$ SNR. However, compared to the low dimension case, AFS struggles more here.
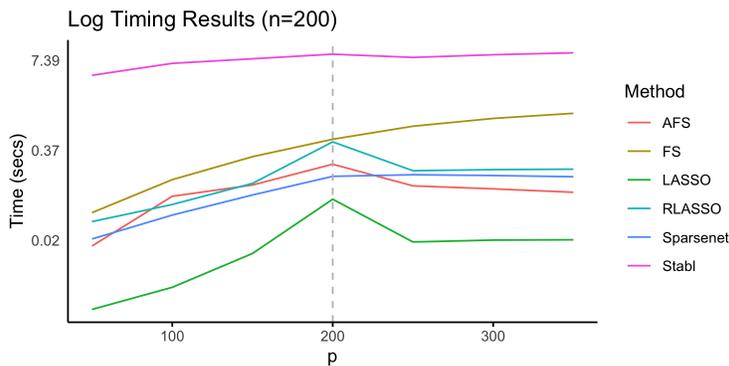

Figure 8: Time in seconds to fit each model under $n = 200$ and varying p, averaged across 50 trials. AFS (computations in `R`), demonstrates computational efficiency similar to that of SparseNet (computations in `fortran`), beating Stabl (computations in `R`), FS (computations in `C`), and RLASSO (computations in `C`). The LASSO (computations in `C`) remains the most computationally efficient across all p

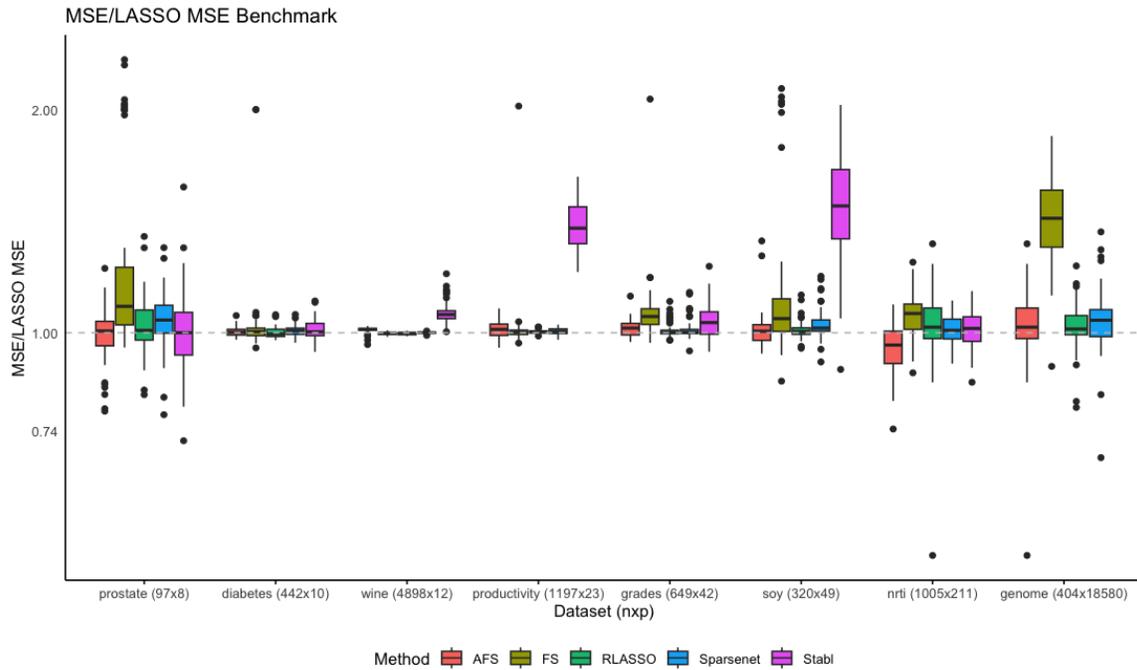Notably, on the most challenging dataset—where all methods performed worse than random

Figure 9: MSE relative to LASSO MSE results from 50 trials on a log scale. Stabl was unable to run for the genome dataset due to insufficient memory allocation for vector storage in R.

chance—AFS still outperformed the other approaches and showed lower variance across the 50 trials.

5. Discussion

We propose a method, Adaptive Forward Stepwise, to address the need for sparser solutions than the LASSO while balancing predictive performance and computational efficiency. AFS is a sparse regression method that bridges Forward Stepwise and the LASSO. Our method produces sparser solutions than the LASSO under appropriate tuning while still allowing for shrinkage, unlike FS. Across numerous simulations under varying signal-to-noise ratios and correlation structures, AFS produces robust performance. Comparatively, other methods experienced large variation of MSE and sparsity across different settings. Similarly, in numerous real data examples, AFS matches or outperforms the LASSO, FS, SparseNet, RLASSO, and Stabl. Our method is also easily modifiable to classification tasks and can be implemented in less time than several other methods. While AFS excels in maintaining one of the lowest MSEs across different settings, it encounters challenges in high-correlation, $n \ll p$ scenarios. An implementation of our method as a Python and R package is forthcoming.
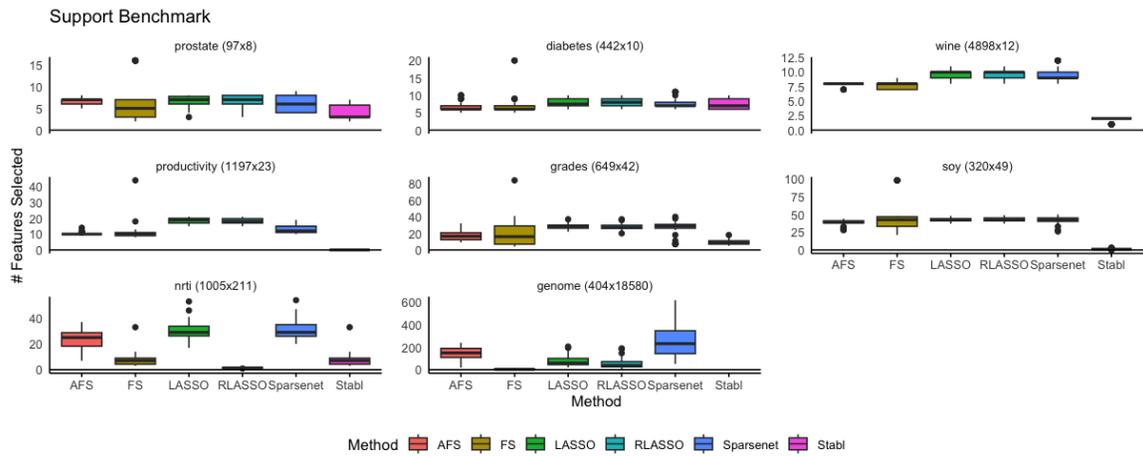
Figure 10: Number of features selected in the final model. AFS generally selects fewer features than other methods on smaller datasets, while maintaining competitive performance, except in the high dimensional, high correlation genome dataset. There, RLASSO has the best combination of sparsity and MSE.
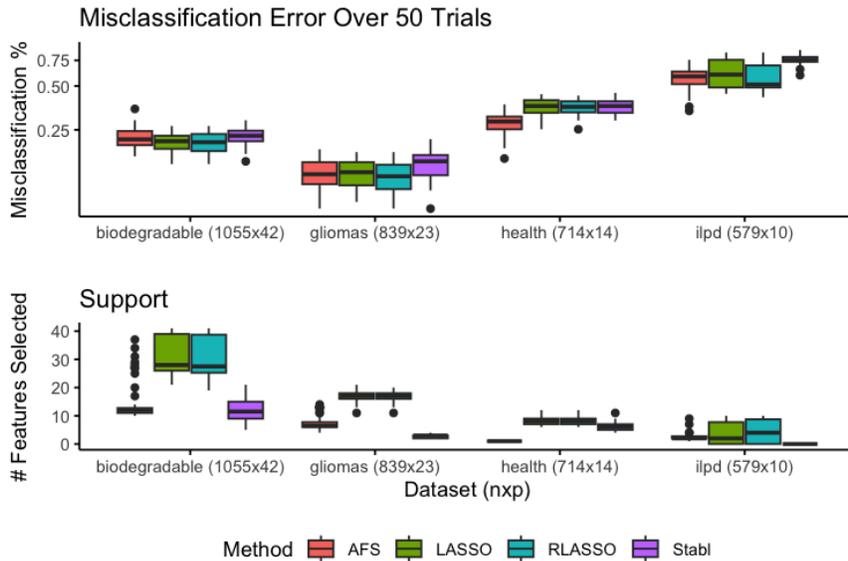
Acknowledgments

Figure 11: Comparison of AFS, LASSO, RLASSO, and Stabl for a binary classification task across UCI datasets. Test misclassification % on the log scale and number of non-zero coefficients (support) of the final model are reported.

Appendix A. Proof of results

A.1 Proof of Theorem 2

Proof We first show that for any fixed $\epsilon > 0$, there exists a $\rho \to 0$ such that for any t, the AFS step index,

$$\|\hat{\beta}_\rho^{AFS}(t) - \hat{\beta}^{LAR}(t)\|_2 \leq \epsilon, \tag{9}$$

which implies that the residuals $\hat{r} = y - X\hat{\beta}$ satisfy

$$\|\hat{r}^{AFS}(t) - \hat{r}^{LAR}(t)\|_2 \leq \epsilon\|X\|_{op}. \tag{10}$$

Let $s^{(k)}(\rho)$ represent the AFS iteration when the kth unique variable enters the active set. To show the bound in Eqn 9, we show that for a sequence of $\epsilon_k > 0$, there exists a $\rho \to 0$ such that

$$\|\hat{\beta}_{s^{(k)},\rho}^{AFS} - \hat{\beta}_k^{LAR}\|_2 \leq \epsilon_k \quad \forall k \in \{1,..,p\} \tag{11}$$

Eqn. 11 says that as $\rho \to 0$, the AFS coefficient is sufficiently close to that of LAR at the knots where a new variable enters the active set, since both LAR and AFS are piecewise linear functions.

Consider the base case at $k = 0$. Both procedures initiate with $\hat{\beta}_0 = 0$ so $\epsilon_0 = 0$ and $\hat{r}_{s^{(0)}}^{AFS} = \hat{r}_0^{LAR}$. This also implies that $\mathcal{A}_{s^{(1)}}^{AFS} \equiv \mathcal{A}_1^{LAR}$. For the inductive step, assume that

$$\|\hat{\beta}_{s^{(k-1)},\rho}^{AFS} - \hat{\beta}_{k-1}^{LAR}\|_2 \leq \epsilon_{k-1} \text{ and } \mathcal{A}_{s^{(k)}}^{AFS} \equiv \mathcal{A}_k^{LAR}.$$

16

We now proceed with proving Eqn. 11 by induction. Let $d(\rho) := s^{(k)}(\rho) - s^{(k-1)}(\rho)$ be the number of steps AFS takes before a new variable enters the active set. The coefficients in AFS updates as $\hat{\beta}^{\text{AFS}}_{m+1,\rho} = (1-\rho)\hat{\beta}^{\text{AFS}}_{m,\rho} + \rho\hat{\beta}^{OLS}_{\mathcal{A}_{m+1}}$ so expanding out the geometric sequence yields

$$\hat{\beta}^{\text{AFS}}_{s^{(k)}} = (1-\rho)^{d(\rho)}\hat{\beta}^{\text{AFS}}_{s^{(k-1)}} + \hat{\beta}^{OLS}_{\mathcal{A}_{s^{(k)}}}(1 - (1-\rho)^{d(\rho)}) \tag{12}$$

Meanwhile, the update from LAR from step k to $k+1$ can be written as

$$\hat{\beta}^{\text{LAR}}_{k+1} = (1-\zeta_{k+1})\hat{\beta}^{\text{LAR}}_k + \zeta_{k+1}\hat{\beta}^{OLS}_{\mathcal{A}_{k+1}} \tag{13}$$

where $\zeta_{k+1} := \frac{\hat{\gamma}_{k+1}}{\bar{\gamma}_{k+1}} \in (0,1)$. Two properties of the procedures are key:

1. ζ_{k+1} is the smallest step in the direction of $(\hat{\beta}^{\text{OLS}}_{\mathcal{A}_{k+1}})_{j^*_{k+1}}$ before a new $j^* \notin \mathcal{A}_{k+1}$ enters the active set for the variable selection criterion.

2. In both LAR and AFS, once a new j^* enters the active set at time $k+1$, no $j \in \mathcal{A}_k$ will be chosen again, by construction.

We will also use the following property, which follows from triangle inequality:
    For any vectors v_1 and v_2 such that $\|v_1 - v_2\| \leq c, c > 0$ and any constants a, b,

$$\|av_1 - bv_2\| \leq a\|v_1 - v_2\| + |a - b|\|v_2\|_2 \tag{14}$$

Let $\delta_k := (1 - (1-\rho)^{d(\rho)}) - \zeta_k$ be the difference in the total distance AFS and LAR moves in the direction of a OLS coefficient before a new variable enters their respective active sets. Then we have

$$\|\hat{\beta}^{AFS}_{s^{(k)},\rho} - \hat{\beta}^{LAR}_k\|_2 \leq \|(1-\zeta_k)\hat{\beta}^{LAR}_{k-1} - (1-\rho)^{d(\rho)}\hat{\beta}^{\text{AFS}}_{s^{(k-1)},\rho}\|_2 + \|(1 - (1-\rho)^{d(\rho)} - \zeta_k)\hat{\beta}^{OLS}_{\mathcal{A}_k}\|_2 \tag{15}$$

$$\leq |\delta_k|\|\hat{\beta}^{LAR}_{k-1}\|_2 + (1-\rho)^{d(\rho)}\|\hat{\beta}^{AFS}_{s^{(k-1)}} - \hat{\beta}^{LAR}_{k-1}\|_2 + \delta_k\|\hat{\beta}^{OLS}_{\mathcal{A}_k}\|_2 \tag{16}$$

$$\leq 2|\delta_k|\|\hat{\beta}^{LAR}_{k-1}\|_2 + (1-\rho)^{d(\rho)}\epsilon_{k-1} \tag{17}$$

where the first inequality uses the representation of the AFS and LAR updates (12, 13) and Cauchy-Schwarz while the second inequality follows by property 14 and Cauchy-Schwarz.
    Now, we need to show that for any k, as $\rho \to 0$,

$$d(\rho)\log(1-\rho) \to \log(1-\zeta_k) \tag{18}$$

To show this, we have by the definition of $d(\rho)$ that

$$1 - (1-\rho)^{d(\rho)-1} \leq \zeta_k \leq 1 - (1-\rho)^{d(\rho)}$$

and therefore

$$\frac{d(\rho) - 1}{d(\rho)}d(\rho)\log(1-\rho) \leq \log(1-\zeta_k) \leq d(\rho)\log(1-\rho)$$

Since $d(\rho) \to \infty$ as $\rho \to 0$, taking the limit as $\rho \to 0$ gives us Eqn. 18.

Since $\|\hat{\beta}_{k-1}^{LAR}\|$ is fixed and not dependent on ρ, we have by Eqn. 18 that as $\rho \to 0$, $\delta_k \to 0$. We also have $(1 - \rho)^{d(\rho)} < 1$, giving us

$$\|\hat{\beta}_{s(k)}^{AFS} - \hat{\beta}_k^{LAR}\|_2 \leq \epsilon_{k-1}$$

when $\rho \to 0$. Since this holds for any arbitrary k, we can choose ρ to be sufficiently small such that we can fix an ϵ_k that is arbitrarily close to ϵ_{k-1}. By assumption, ϵ_0 is 0, which proves Eqn. 9.

So far, we have only shown that the LAR and AFS coefficients agree at the knots where a new variable enters the active set. To see that Eqn. 11 holds, recall that both procedures are piecewise linear functions moving in the direction of the OLS coefficient. Therefore, it must be that they also agree for any t since those are points between the straight line that connects matching endpoints.

Now, we prove that the active sets must also agree by showing that the difference between the selection criterion objective for AFS and LAR is arbitrarily small at any step. By the assumption of no ties in variable selection, for any $\tilde{\epsilon}_k > 0$,

$$(x_{j_k^{*LAR}})^{\mathsf{T}} \hat{r}_{k-1}^{LAR} \geq \max_{j \in \{1,..p\}} x_j^{\mathsf{T}} \hat{r}_k^{LAR} + \tilde{\epsilon}_k \tag{19}$$

This inequality follows from the selection criterion of LAR,

$$j_k^{*LAR} = \underset{j \in \{1,..p\}}{\operatorname{argmax}} \, x_j^{\mathsf{T}} \hat{r}_{k-1}^{LAR},$$

since the inner product of the selected variable column at step $k-1$ must be as large as the inner product of any variable column and the residual in the next step. ϵ_k can be arbitrarily close to 0, so we can fix

$$\epsilon_k \leq \frac{\tilde{\epsilon}_k}{2 \max_{j \in \{1,..p\}} \|x_j\|_2}.$$

Since AFS has the selection rule as LAR, Eqn. 19 also holds for the AFS residual. This gives us

$$\begin{aligned}
|x_j^{\mathsf{T}} (\hat{r}_{s(k)}^{AFS} - \hat{r}_k^{LAR})| &\leq |x_j^{\mathsf{T}} (\hat{r}_{s(k-1)}^{AFS} - \hat{r}_{k-1}^{LAR})| \\
&\leq |x_j^{\mathsf{T}} (X(\hat{\beta}_{s(k-1)}^{AFS} - \hat{\beta}_{k-1}^{LAR}))| \\
&\leq \epsilon_k \|X\|_{op} \\
&\leq \tilde{\epsilon}_k,
\end{aligned}$$

when combined with our inductive step assumption. Taking $\tilde{\epsilon}_k$ to 0 concludes the proof. ∎

A.2 Proof of Theorem 3

Proof

The below derivation draws from the proof of Theorem 2 in Bühlmann and Yu (2006). We can equivalently write 8 as

$$\hat{\beta}_m^{AFS} = D_m X^{\mathsf{T}} y = D_m \hat{\beta}^{OLS}$$

where $D_m \in \mathbb{R}^{p \times p}$ is a diagonal matrix with the j-th entry as $(1 - (1-\rho)^{\ell_{j,m}})$. Then $\Delta_{m-1,m} \coloneqq RSS_m - RSS_{m-1}$ decreases in m such that $\Delta_{m-1,m} = (1-\rho)^2 \Delta_{m,m+1}$ since

$$
\begin{aligned}
\Delta_{m-1,m} &= \|y - XD_m X^\intercal y\|_2^2 - \|y - XD_{m-1} X^\intercal y\|_2^2 \\
&= \|X^\intercal(y - XD_m X^\intercal y)\|_2^2 - \|X^\intercal(y - XD_{m-1} X^\intercal y)\|_2^2 \\
&= \|(I - D_m) X^\intercal y\|_2^2 - \|(I - D_{m-1}) X^\intercal y\|_2^2 \\
&= \sum_{j=1}^{p} \|[[(I - D_m)^2 - (I - D_{m-1})^2](\hat{\beta}^{OLS})_j^2\|_2.
\end{aligned}
$$

Here, we apply the orthogonality of X to get the third equality. Then for each fixed $j \in \mathcal{A}$, $\exists \delta^2 > 0$ such that

$$
\begin{aligned}
\left[(1-\rho)^{2(\ell_{j,h})} - (1-\rho)^{2(\ell_{j,h+1})}\right](\hat{\beta}^{OLS})_j^2 &> \delta_{h,j}^2, \quad h \in \{1, .., m-1\} \\
\left[(1-\rho)^{2(\ell_{j,m+1})} - (1-\rho)^{2(\ell_{j,m+1})}\right](\hat{\beta}^{OLS})_j^2 &\le \delta_{m,j}^2
\end{aligned}
$$

From the above, consider the approximation $\delta_{m,j}^2 \approx (1-\rho)^{2\ell_{j,m}}(1 - (1-\rho)^2)(\hat{\beta}^{OLS})_j^2$. This gives us the following soft-threshold approximation:

$$
\hat{\beta}_{j,m}^{AFS} \approx \hat{\beta}_{j,m}^{ST} \coloneqq \begin{cases} \hat{\beta}_j^{OLS} - \lambda_{j,m} & \text{if } \hat{\beta}_j^{OLS} \ge \lambda_{j,m} \\ 0 & \text{if } |\hat{\beta}_j^{OLS}| < \lambda_{j,m} \\ \hat{\beta}_j^{OLS} + \lambda_{j,m} & \text{if } \hat{\beta}_j^{OLS} \le -\lambda_{j,m} \end{cases} \tag{20}
$$

where $\lambda_{j,m} = \frac{\delta_{m,j}}{\sqrt{1-(1-\rho)^2}}$. In fact, when $\rho \to 0$, the difference in the estimators goes to 0. We refer readers to Bühlmann and Yu (2006) for details. Finally, since the largest possible $\Delta_{m-1,m}$ occurs when $m = k_j$, we get that the value of $\lambda_{j,m}$ must be between

$$
(1-\rho)^{2\ell_{j,m}}\sqrt{1 - (1-\rho)^2} \text{ and } (1-\rho)^2 \sqrt{1 - (1-\rho)^2}
$$

$\blacksquare$

Remark 4 *Although the monotonic behavior of $\Delta_{m-1,m}$ no longer holds under non-orthogonal design, RSS_m is still decreasing in m. As a result, for small ρ, we would expect 3 to still be a reasonable estimator, as seen empirically in Figure 12 if $\Delta_{m-1,m}$ is decreasing in m for most iterations.*

Appendix B. Simulations

B.1 Degrees of freedom

We use $B = 1000$ bootstrap iterations to estimate the dof $\sum_{i=1}^{n} \text{Cov}(y_i, \hat{y}_i)/\sigma^2$ (Efron, 1986). As anticipated, the dof varies between that of the LASSO (dof = $|\mathcal{A}|$) and FS, as seen in Hastie et al. (2020).
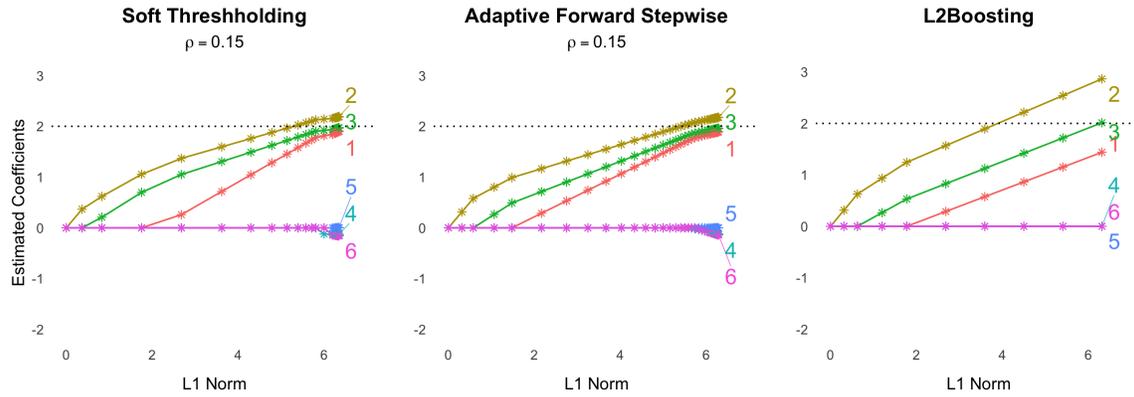
Figure 12: Coefficient paths for the approximate soft thresholding estimator 3 at $\rho = 0.15$, AFS at $\rho = 0.15$, and L2Boosting. Simulation for $n = 100, p = 6$ under a non-orthogonal design matrix X, with $\beta_1, \beta_2, \beta_3 = 2$ and 0 otherwise.
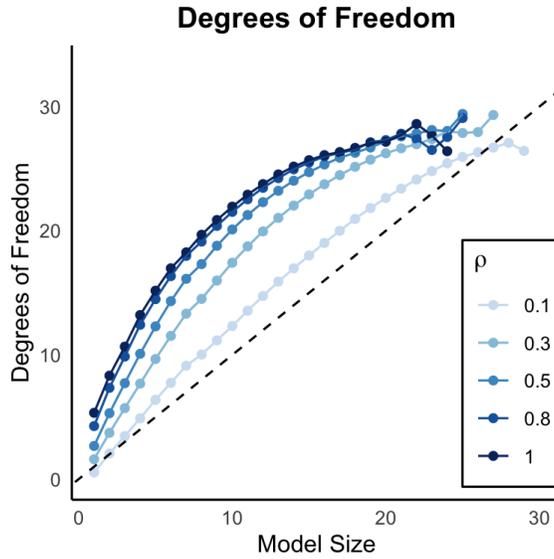

Figure 13: Bootstrap estimates of dof for AFS under different ρ. Simulation uses $\sigma = 1.75$

B.2 Benchmark results

Appendix C. Inference

We are interested in the following test
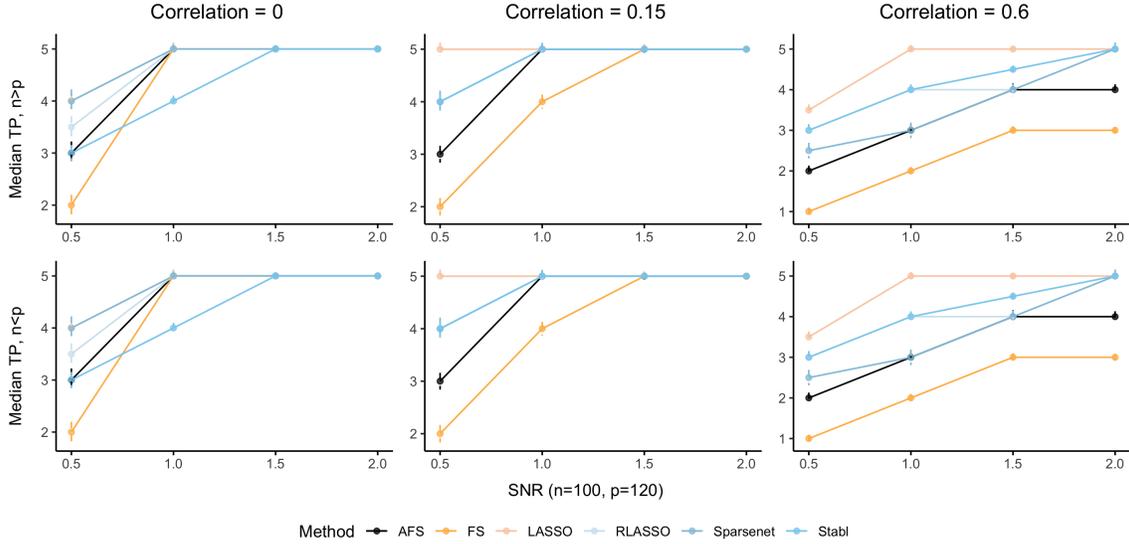
$$H_0 : v^\mathsf{T}\theta = 0$$

20

Figure 14: Simulation results averaged across 50 trials. AFS recovers the true support under no and low correlation settings. However, it tends to perform less well in the lowest SNR and high correlation setting.

conditional on the chosen A_k at step k. Then the selection event, y of AFS at step k can be represented as a polyhedron of the form

$$\mathcal{P} = \{y : \hat{A}_k = A_k, \hat{s}_{A_k} = s_{A_k}\} = \{y : \Gamma y > 0\}$$

where $A_k = [j_1, ..., j_k]$ is the set of active variables after k steps, $s_k = \text{sign}(X_k^\mathsf{T} r)$, and r is the residual from regressing y onto $X_{A_{k-1}}$. The construction follows that of the polyhedral sets for FS selection events as in Tibshirani et al. (2015). By induction, we get the matrix Γ with rows based on the following conditions:

$$s_k X_{j_k}^\mathsf{T} \left(I - X\rho \sum_{i=0}^{k-1} (1-\rho)^k \hat{\nu}_{A_{k-1-i}} \right) y \geq \pm X_j^\mathsf{T} \left(I - X\rho \sum_{i=0}^{k-1} (1-\rho)^k \hat{\nu}_{A_{k-1-i}} \right) y, \ \forall j \neq j_k$$

$$s_k X_{j_k}^\mathsf{T} \left(I - X\rho \sum_{i=0}^{k-1} (1-\rho)^k \hat{\nu}_{A_{k-1-i}} \right) y > \pm X_j^\mathsf{T} \left(I - X\rho \sum_{i=0}^{k-1} (1-\rho)^k \hat{\nu}_{A_{k-1-i}} \right) y, \ \forall j \in A_{k-1}$$

Unlike in FS, the additional condition is needed to guarantee we are looking at the set for which X_{j_k} gets chosen for the first time. Then we can directly apply Theorem 5.2 of Lee et al. (2016) to get a conditional test statistic. We refer readers to the paper for details on the test and confidence interval construction.

Appendix D. Modification of AFS to recover LAR and the LASSO

We can modify the AFS algorithm to recover LAR for a fixed ρ:

Algorithm 3 Adaptive Forward Stepwise - LAR Modifications

6: While $m < M$ and $\|\hat{\beta}^{AFS}\|_1 < h$, let

7:     $j_m^* = \underset{j \in \{1,..,p\}}{\operatorname{argmax}} |x_j^\mathsf{T}(y - X\hat{\beta}_{m-1}^{AFS})|$ ▷ Select most correlated variable with current residuals

8:     $A_m = A_{m-1} \cup j_m^*$ ▷ Update active set

9:     $\hat{\nu}_m = \hat{\beta}_{\mathcal{A}_m}^{OLS}$, the OLS coefficients of y on the active set $\mathcal{A}_m$ ▷ Compute OLS coefficients

10:     Begin recovery of LAR coefficient:

11:        Fix $\epsilon_m > 0$ small and let $\tilde{\rho} = \rho - \epsilon_m$

12:        $\tilde{\beta}^{AFS} = (1 - \tilde{\rho})\hat{\beta}_{m-1}^{AFS} + \tilde{\rho}\hat{\nu}_m$

13:        $\tilde{j} = \underset{j \in \{1,...,p\}}{\operatorname{argmax}} |x_j^\mathsf{T}(\mathbf{y} - \sum_{p=1}^P \tilde{\beta}_{k-2,p}^{AFS} x_p)|$

14:        While $\tilde{j} = j_m^*$

15:           Increase ϵ_m and repeat LAR coefficient recovery. As $\tilde{\rho} \to \rho$, recovers the LAR coefficient

16:     $\hat{\beta}_m^{AFS} = (1 - \tilde{\rho})\hat{\beta}_{m-1}^{AFS} + \tilde{\rho}\hat{\nu}_m$ ▷ Update AFS coefficients

---

We can further modify the above to recover the LASSO with the same restriction as in Efron et al. (2004): If $\operatorname{sign}(\hat{\beta}_{k,j}^{AFS}) \neq \operatorname{sign}(\hat{\beta}_{k-1,j}^{AFS}) \neq 0$, remove x_j from $\mathcal{A}_k$ and repeat procedure.

References

Peter Bühlmann. Boosting for high-dimensional linear models. *The Annals of Statistics*, 34 (2):559–583, 2006. doi: 10.1214/009053606000000092.

Peter Bühlmann and Torsten Hothorn. Boosting Algorithms: Regularization, Prediction and Model Fitting. *Statistical Science*, 22(4):477–505, 2007. doi: 10.1214/07-STS242.

Peter Bühlmann and Torsten Hothorn. Twin Boosting: improved feature selection and prediction. *Statistics and Computing*, 20(2):119–138, 2010.

Peter Bühlmann and Bin Yu. Sparse Boosting. *Journal of Machine Learning Research*, 7 (36):1001–1024, 2006.

Scott Chen, David Donoho, and Michael Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.

Denis Chetverikov, Zhipeng Liao, and Victor Chernozhukov. On cross-validated lasso in high dimensions. *The Annals of Statistics*, 49(3):1300–1317, 2021.

Paulo Cortez. Student Performance. UCI Machine Learning Repository, 2008.

Arnak Dalalyan, Mohamed Hebiri, and Johannes Lederer. On the prediction performance of the Lasso. *Bernoulli*, 23(1):552–581, 2017. doi: 10.3150/15-BEJ756.

Bradley Efron. How Biased is the Apparent Error Rate of a Prediction Rule? *Journal of the American Statistical Association*, 81(394):461–470, 1986. doi: 10.1080/01621459.1986.10478291.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, 2004. doi: 10.1214/009053604000000067.

Jianqing Fan and Runze Li. Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001. doi: 10.1198/016214501753382273.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society Series B*, 70(5):849–911, 2008.

Jerome Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001. doi: 10.1214/aos/1013203451.

Jerome H. Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010. doi: 10.18637/jss.v033.i01.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2 edition, 2009. doi: 10.1007/978-0-387-84858-7.

Trevor Hastie, Robert Tibshirani, and Ryan Tibshirani. Best Subset, Forward Stepwise or Lasso? Analysis and Recommendations Based on Extensive Comparisons. *Statistical Science*, 35(4):579–592, 2020. doi: 10.1214/19-STS733.

Julien Hédou, Ivana Marić, Grégoire Bellan, Jakob Einhaus, Dyani Gaudillière, Francois-Xavier Ladant, Franck Verdonk, Ina Stelzer, Dorien Feyaerts, Amy Tsai, Edward Ganio, Maximilian Sabayev, Joshua Gillard, Jonas Amar, Amelie Cambriel, Tomiko Oskotsky, Alennie Roldan, Jonathan Golob, Marina Sirota, Thomas Bonham, Masaki Sato, Maïgane Diop, Xavier Durand, Martin Angst, David Stevenson, Nima Aghaeepour, Andrea Montanari, and Brice Gaudillière. Discovery of sparse, reliable omic biomarkers with Stabl. *Nature Biotechnology*, 2024.

Jason Lee, Dennis Sun, Yuekai Sun, and Jonathan Taylor. Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907–927, 2016. doi: 10.1214/15-AOS1371.

Richard Lockhart, Jonathan Taylor, Ryan Tibshirani, and Robert Tibshirani. A significance test for the lasso. *The Annals of Statistics*, 42(2):413 – 468, 2014. doi: 10.1214/13-AOS1175. URL https://doi.org/10.1214/13-AOS1175.

Preeti Malani, Jeffrey Kullgren, and Erica Solway. National Poll on Healthy Aging (NPHA), [United States], April 2017. Inter-university Consortium for Political and Social Research, 2019.

Kamel Mansouri, Tine Ringsted, Davide Ballabio, Roberto Todeschini, and Viviana Consonni. QSAR biodegradation. UCI Machine Learning Repository, 2013.

Rahul Mazumder, Jerome Friedman, and Trevor Hastie. SparseNet: Coordinate Descent With Nonconvex Penalties. *Journal of the American Statistical Association*, 106(495): 1125–1138, 2011. doi: 10.1198/jasa.2011.tm09738.

Nicolai Meinshausen. Relaxed Lasso. *Computational Statistics & Data Analysis*, 52(1): 374–393, 2007. doi: 10.1016/j.csda.2006.12.019.

Mexwell. Employee Performance and Productivity Data. Kaggle Dataset, 2021.

Ryszard Michalski and RL Chilausky. Soybean (Large). UCI Machine Learning Repository, 1980.

Balas Natarajan. Sparse Approximate Solutions to Linear Systems. *SIAM Journal on Computing*, 24(2):227–234, 1995. doi: 10.1137/S0097539792240406.

Bendi Ramana, Surendra Prasad, and Nagasuri Venkateswarlu. A Critical Comparative Study of Liver Patients from USA and INDIA: An Exploratory Analysis. *International Journal of Computer Science*, 9, 2012.

Dibyendu Seal, Vivek Das, Saptarsi Goswami, and Rajat De. Estimating gene expression from DNA methylation and copy number variation: A deep learning regression model for multi-omics integration. *Genomics*, 112(4):2833–2841, 2020. doi: 10.1016/j.ygeno.2020. 03.021.

E Tasci, Y Zhuge, H Kaur, K Camphausen, and A Krauze. Hierarchical Voting-Based Feature Selection and Ensemble Learning Model Scheme for Glioma Grading with Clinical and Molecular Characteristics. *International Journal of Molecular Sciences*, 23(22):14155, 2022. doi: 10.3390/ijms232214155.

Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B*, 58:267–288, 1996.

Ryan Tibshirani, Jonathan Taylor, Richard Lockhart, and Robert Tibshirani. Exact Post-Selection Inference for Sequential Regression Procedures. *arXiv preprint*, 2015. doi: 10.48550/arXiv.1401.3889.

Athanasios Tsanas and Angeliki Xifara. Energy Efficiency. UCI Machine Learning Repository, 2012.

Raj Yellow. Wine Quality. Kaggle Dataset, 2021.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010. doi: 10.1214/09-AOS729.

Peng Zhao and Bin Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.