Submitted to the Annals of Statistics

A NEW PERSPECTIVE ON BOOSTING IN LINEAR REGRESSION VIA SUBGRADIENT OPTIMIZATION AND RELATIVES

By Robert M. Freund,[‡,*] Paul Grigas[§,†] and Rahul Mazumder[¶,*]

Massachusetts Institute of Technology and University of California, Berkeley†*

We analyze boosting algorithms [22, 25, 14] in linear regression from a new perspective: that of modern first-order methods in convex optimization. We show that classic boosting algorithms in linear regression, namely the incremental forward stagewise algorithm (FS$_\varepsilon$) and least squares boosting (LS-Boost(ε)), can be viewed as subgradient descent to minimize the loss function defined as the maximum absolute correlation between the features and residuals. We also propose a minor modification of FS$_\varepsilon$ that yields an algorithm for the Lasso, and that may be easily extended to an algorithm that computes the Lasso path for different values of the regularization parameter. Furthermore, we show that these new algorithms for the Lasso may also be interpreted as the same master algorithm (subgradient descent), applied to a regularized version of the maximum absolute correlation loss function. We derive novel, comprehensive computational guarantees for several boosting algorithms in linear regression (including LS-Boost(ε) and FS$_\varepsilon$) by using techniques of first-order methods in convex optimization. Our computational guarantees inform us about the statistical properties of boosting algorithms. In particular they provide, for the first time, a precise theoretical description of the amount of data-fidelity and regularization imparted by running a boosting algorithm with a prespecified learning rate for a fixed but arbitrary number of iterations, for *any* dataset.

1. Introduction. Boosting [45, 16, 25, 46, 33] is an extremely successful and popular supervised learning method that combines multiple weak[1]

[‡]This author's research is supported by AFOSR Grant No. FA9550-11-1-0141 and the MIT-Chile-Pontificia Universidad Catolica de Chile Seed Fund.

[§]This author's research has been partially supported through an NSF Graduate Research Fellowship and the MIT-Chile-Pontificia Universidad Catolica de Chile Seed Fund.

[¶]This author's research has been funded by ONR Grant N000141512342 and a grant from the Betty Moore-Sloan foundation.

MSC 2010 subject classifications: Primary 62J05, 62J07k; secondary 90C25

Keywords and phrases: Boosting, Forward Stagewise Regression, Linear Regression, Bias-Variance tradeoff, Shrinkage, Convex Optimization, Computational Guarantees.

[1]this term originates in the context of boosting for classification, where a "weak" classifier is slightly better than random guessing.

learners into a powerful "committee." AdaBoost [20, 46, 33], one of the earliest boosting algorithms developed in the context of classification, may be viewed as an optimization algorithm: a form of gradient descent in a certain function space [5, 4]. In an influential paper, [25] nicely interpreted boosting methods used in classification problems as instances of stagewise additive modeling [32]. [22] provided an unified view of stagewise additive modeling and steepest descent minimization methods in function space to explain boosting methods. For related perspectives from the machine learning community, we refer the reader to [37, 42] and the references therein.

An important instantiation of boosting, and the topic of the present paper, is its application in linear regression [22, 33, 6, 7]. We use the usual notation with model matrix $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_p] \in \mathbb{R}^{n \times p}$, response vector $\mathbf{y} \in \mathbb{R}^{n \times 1}$, and regression coefficients $\beta \in \mathbb{R}^p$. We assume that the $\mathbf{X}_i$'s have been centered to have zero mean and unit ℓ_2 norm and $\mathbf{y}$ is also centered to have zero mean. For a regression coefficient β, the predicted value of the response is $\mathbf{X}\beta$ and $r = \mathbf{y} - \mathbf{X}\beta$ denotes the residuals.

Boosting and Implicit Regularization. We begin our study with a popular algorithm: Least Squares Boosting – also known as LS-Boost(ε) [22] – which is formally described herein in Section 2. LS-Boost(ε) has been studied by several authors [8, 6, 7, 36, 21]. Starting from the null model $\hat{\beta}^0 = 0$, at the k-th iteration LS-Boost(ε) determines the covariate index j_k with the best univariate fit to the current residuals $\hat{r}^k = \mathbf{y} - \mathbf{X}\hat{\beta}^k$:

$$j_k \in \arg\min_{1 \le m \le p} \sum_{i=1}^{n} (\hat{r}_i^k - x_{im}\tilde{u}_m)^2, \text{ where, } \quad \tilde{u}_m = \arg\min_{u \in \mathbb{R}} \left(\sum_{i=1}^{n} (\hat{r}_i^k - x_{im} u)^2 \right).$$

The algorithm then updates the j_k-th regression coefficient with a shrinkage factor $\varepsilon > 0$: $\hat{\beta}_{j_k}^{k+1} \leftarrow \hat{\beta}_{j_k}^{k} + \varepsilon \tilde{u}_{j_k}$, with all other regression coefficients unchanged. A close cousin of the LS-Boost(ε) algorithm is the Incremental Forward Stagewise algorithm [33, 14] – also known as FS_ε – which is formally described herein in Section 3. FS_ε chooses the covariate most correlated (in absolute value) with the residual $\hat{r}^k$ and performs the update:

$$\hat{\beta}_{j_k}^{k+1} \leftarrow \hat{\beta}_{j_k}^{k} + \varepsilon \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}), \;\; \hat{\beta}_{j}^{k+1} \leftarrow \hat{\beta}_{j}^{k}, j \neq j_k, \text{ where, } j_k \in \arg\max_{j \in \{1,\ldots,p\}} |(\hat{r}^k)^T \mathbf{X}_j| \,.$$

(Since the covariates are standardized, both LS-Boost(ε) and FS_ε lead to the same variable selection for a given $\hat{r}^k$.) LS-Boost(ε) and FS_ε have curious similarities but subtle differences as we characterize in Section 3 (see also [7]). In both algorithms, the shrinkage factor ε, also known as the learning rate, counterbalances the greedy selection strategy of choosing the *best* covariate. Qualitatively speaking, a greedy fitting procedure may overfit quickly – a small value of ε slows down the learning rate as compared to a larger choice of ε (that is fine-tuned to minimize the training error) and leads to slower overfitting. With a small ε it is possible to explore a larger class of models, with varying degrees of shrinkage – this often leads to models with better predictive power [22]. Let M denote the number of boosting iterations. Then both M and ε (the shrinkage factor) together control the training error and the amount of shrinkage. We refer the reader to Figure 1, which illustrates the evolution of the algorithmic properties of the LS-Boost(ε) algorithm as a function of k and ε. Up until now, as pointed out by [33], the understanding of the tradeoff between regularization and data-fidelity for these boosting methods has been rather qualitative. One of the contributions of this paper is a precise quantification of this tradeoff. In Sections 2 and 3 we will derive comprehensive computational guarantees for these algorithms which provide a formal description of how M and ε control the amount of training error and regularization in FS_ε and LS-Boost(ε) , as well as precise bounds on their tradeoffs between regularization and data-fidelity. Furthermore, in Section 3.3 we will provide a unified treatment of LS-Boost(ε) , FS_ε, and a generalization with adaptive step-sizes $\text{FS}_{\varepsilon_k}$ – wherein we will show that all of these methods can be viewed as special instances of (convex) subgradient optimization.

Both LS-Boost(ε) and FS_ε may be interpreted as "cautious" versions of the classical Forward Selection or Forward Stepwise regression [38, 52]. This algorithm identifies the variable most correlated (in absolute value) with the current residual, includes it in the model, and updates the *joint least squares* fit based on the current set of predictors – this update strategy makes stepwise regression *aggressive* and hence different from FS_ε and LS-Boost(ε) .

Lasso *and Explicit Regularization.* All of the algorithms described above impart regularization in an implicit fashion through the choice of ε and M. In contrast, let us consider the constraint version of the Lasso [48]:

$$
\text{(1.1)} \qquad \text{Lasso} : \quad L^*_{n,\delta} := \begin{array}{ll} \min\limits_{\beta} & \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 \\ \text{s.t.} & \|\beta\|_1 \le \delta \ , \end{array}
$$

with regularization parameter $\delta \geq 0$. The nature of regularization via the Lasso is explicit – it is set up to find the best least squares solution subject to a constraint on the ℓ_1 norm of the regression coefficients. We let $\{\hat\beta_\delta\}$ denote the path of solutions of (1.1) for all $\delta \geq 0$, otherwise known as the "Lasso path."

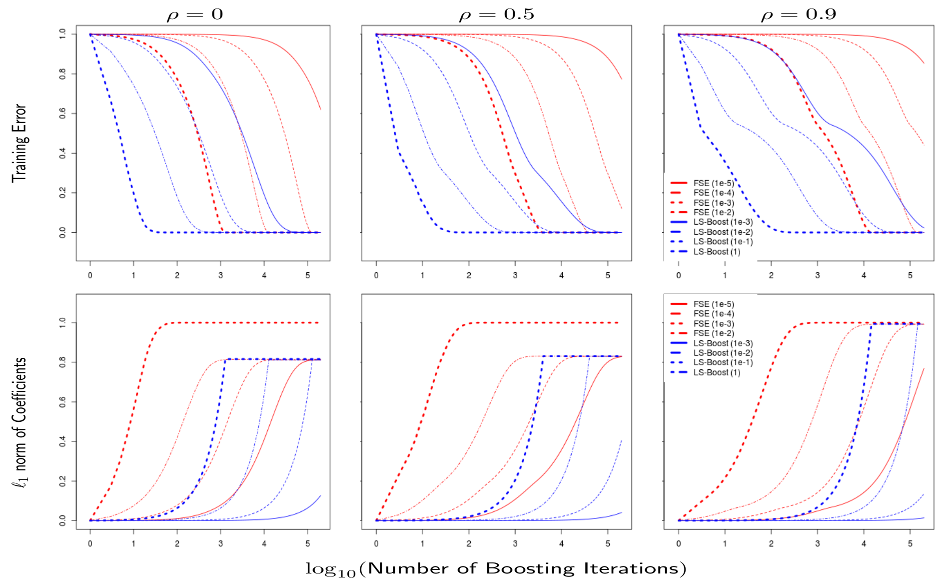

FIG 1. *Evolution of* LS-BOOST(ε) *and* FS_ε *versus iterations (in log-scale), for a synthetic dataset with* $n = 50$, $p = 500$*: the covariates are Gaussian with pairwise correlations* ρ*, the true* β *has ten non-zeros with* $\beta_i = 1, i \leq 10$ *and* $SNR = 1$*. Different* ρ *and* ε *values are considered. [Top Row] Shows the training errors for different learning rates, [Bottom Row] shows the* ℓ_1 *norm of coefficients produced by the algorithms for different learning rates (the y-axis values have been re-scaled to lie in* $[0, 1]$*).*

Boosting and LASSO*: Path Properties.* Although LASSO and the above boosting methods originate from different perspectives, there are curious similarities between the two as is nicely explored in [33, 14, 34]. For certain datasets, the coefficient profiles[2] of LASSO and FS_0 (defined to be the limiting case of the FS_ε algorithm as $\varepsilon \to 0+$) are exactly the same (see Figure 2, top panel) [33]. However, they are different in general (Figure 2, bottom panel). Efforts to understand the FS_ε algorithm paved the way for the Least Angle Regression algorithm – also known as LAR [14] (see also [33]). The LAR is a unified framework: one instance of LAR computes the LASSO path and another delivers a coefficient profile for FS_0.

The similarities between the LASSO and boosting coefficient profiles motivate us to develop a minor modification of boosting that generates the LASSO path, which we will accomplish in Sections 4 and 5. In a different line of approach, [53] describes BLASSO, a modification of the FS_ε algorithm

[2]By a coefficient profile we mean the map $\lambda \mapsto \hat{\beta}_\lambda$ where, $\lambda \in \Lambda$ indexes a family of coefficients $\hat{\beta}_\lambda$. For example, the regression coefficients delivered by FS_0 delivers a coefficient profile as a function of their ℓ_1-norms.

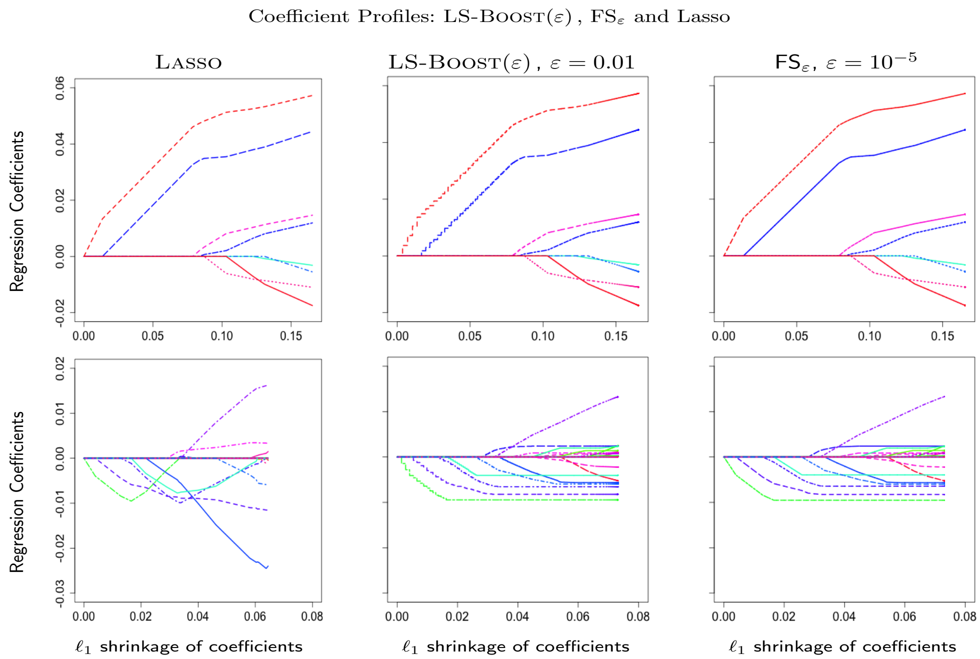

Fig 2. *Coefficient Profiles of several algorithms as as function of the ℓ_1-norm of coefficients on different datasets. [Top Panel] Prostate Cancer dataset described in Section 6 with $n = 98$ and $p = 8$. All profiles look similar. [Bottom Panel] A subset of samples of the Prostate Cancer dataset with $n = 10$; we also included all second order interactions to get $p = 44$. The coefficient profile of* Lasso *is seen to be different from FS$_\varepsilon$ and* LS-Boost(ε) *. Figure A.1 [19] shows training error vis-à-vis the ℓ_1-shrinkage of the models for the same data.*

with the inclusion of additional "backward steps" so that the resultant coefficient profile mimics the Lasso path.

Boosting and Lasso*: Computation.* While solving the Lasso is computationally very attractive for small to moderate-sized datasets, efficient implementations of boosting (for example, FS$_\varepsilon$) are equally efficient[3] [23]. With regard to Lar, for example, computing the FS$_0$ and Lasso profiles have comparable cost. In examples where the number of possible features is extremely large or possibly infinite [43, 44] a boosting algorithm like FS$_\varepsilon$ is computationally more attractive than solving the Lasso.

[3]Friedman [23] shows that an optimized implementation of FS$_\varepsilon$ leads to an entire solution path for a problem with $10,000$ features and 200 samples in 0.5 seconds, whereas solving the Lasso can take up to 6-8 times longer to compute a path of solutions.

Subgradient Optimization as a Unifying Viewpoint of Boosting and the Lasso. In spite of the various nice perspectives on FS_ε and its connections to the LASSO as described above, the present understanding about the relationships between the LASSO path, FS_ε, and LS-BOOST(ε) for arbitrary datasets and $\varepsilon > 0$ has nevertheless been fairly limited. A chief goal of this paper is to contribute some substantial further understanding of the relationship between these objects. Somewhat like the LAR algorithm can be viewed as a master algorithm with special instances yielding the LASSO path and FS_0, we establish herein that FS_ε and LS-BOOST(ε) can be viewed as special instances of one grand algorithm: the subgradient descent method (of convex optimization) applied to the following parametric class of optimization problems:

(1.2)

$$P_\delta \ : \quad \underset{r}{\text{minimize}} \quad \|\mathbf{X}^T r\|_\infty + \frac{1}{2\delta}\|r - \mathbf{y}\|_2^2 \qquad \text{where } r = \mathbf{y} - \mathbf{X}\beta \ \text{ for some } \beta \ ,$$

where $\delta \in (0, \infty]$ is a regularization parameter. Here the first term is the maximum absolute inner-product between the features and residuals, and the second term acts as a regularizer by penalizing residuals that are far from the response. As we describe herein in Section 4, Problem (1.2) is intimately connected to the LASSO problem (1.1) through duality. We then show that the subgradient descent algorithm applied to Problem (1.2) leads to a new boosting algorithm – dubbed $\text{R-FS}_{\varepsilon,\delta}$ (for Regularized incremental Forward Stagewise regression) – that solves the LASSO, and that is almost identical to FS_ε except that it includes an additional simple re-scaling of the coefficients. Section 4 develops a variety of properties of the new algorithm $\text{R-FS}_{\varepsilon,\delta}$ related to regularization, data-fidelity, etc. And in Section 5 we present an adaptive version $\text{PATH-R-FS}_\varepsilon$ of $\text{R-FS}_{\varepsilon,\delta}$ which approximates the LASSO path, with associated approximation guarantees as well. We also observe empirically that $\text{R-FS}_{\varepsilon,\delta}$ has statistical properties similar to LASSO and FS_ε, and it often leads to models that are more sparse than FS_ε.

Summary of Contributions. We view the contributions of this paper as falling into two main streams as follows:

• *Current Boosting Methods and the Subgradient Descent Method.* We show that both LS-BOOST(ε) and FS_ε are instances of the subgradient descent method of convex optimization applied to the problem of minimizing the maximum absolute correlation between features and residuals. This leads to the first-ever computational guarantees for the behavior of these boosting methods as well as precise bounds on the tradeoffs between regularization and data-fidelity for these methods, which hold for *any* dataset. See Theorem

2.1, Proposition 3.2, and Theorem 3.1.

• *New Boosting Method R-FS$_{\varepsilon,\delta}$ connecting FS$_\varepsilon$ and the* LASSO. We present a new boosting algorithm named R-FS$_{\varepsilon,\delta}$ – for Regularized Forward Stagewise regression – that is identical to FS$_\varepsilon$ except for a simple rescaling of the coefficients at each iteration, and that specializes to FS$_\varepsilon$ as well as to an algorithm for solving the LASSO, depending on the choice of δ. We present computational guarantees for convergence of R-FS$_{\varepsilon,\delta}$ to LASSO solutions, and we present a path version of the algorithm that computes an approximation of the LASSO path with associated approximation bounds. See Proposition 4.1, Theorem 4.1, Theorem 5.1, and Corollary 5.1.

Organization of the Paper. The paper is organized as follows. In Section 2 we analyze the convergence behavior of the LS-BOOST(ε) algorithm. In Section 3 we present a unifying algorithmic framework for FS$_\varepsilon$, FS$_{\varepsilon_k}$, and LS-BOOST(ε) as subgradient descent. In Section 4 we introduce R-FS$_{\varepsilon,\delta}$ as a boosting algorithm naturally associated with Problem (1.2). In Section 5, we further expand R-FS$_{\varepsilon,\delta}$ into a method for computing approximate solutions of the LASSO path. Section 6 contains computational experiments. To improve readability, most of the technical details are placed in the supplementary section [19].

1.1. *Notation.* For a vector $x \in \mathbb{R}^m$, we use x_i to denote the i-th coordinate of x. We use superscripts to index vectors in a sequence $\{x^k\}$. Let e_j denote the j-th unit vector in $\mathbb{R}^m$, and let $e = (1, \ldots, 1)$ denote the vector of ones. Let $\|\cdot\|_q$ denote the ℓ_q norm for $q \in [1, \infty]$ with unit ball B_q, and let $\|v\|_0$ denote the number of non-zero coefficients of the vector v. For $A \in \mathbb{R}^{m \times n}$, let $\|A\|_{q_1,q_2} := \max\limits_{x:\|x\|_{q_1} \le 1} \|Ax\|_{q_2}$ be the operator norm. In particular, $\|A\|_{1,2} = \max(\|A_1\|_2, \ldots, \|A_n\|_2)$ is the maximum ℓ_2 norm of the columns of A. For a scalar α, $\mathrm{sgn}(\alpha)$ denotes the sign of α. The notation "$\tilde{v} \leftarrow \arg\max\limits_{v \in S}\{f(v)\}$" denotes assigning $\tilde{v}$ to be any optimal solution of the problem $\max\limits_{v \in S}\{f(v)\}$. For a convex set P let $\Pi_P(\cdot)$ denote the Euclidean projection operator onto P, namely $\Pi_P(\bar{x}) := \arg\min_{x \in P} \|x - \bar{x}\|_2$. Let $\partial f(\cdot)$ denote the subdifferential operator of a convex function $f(\cdot)$. If $Q \neq 0$ is a symmetric positive semidefinite matrix, let $\lambda_{\max}(Q)$, $\lambda_{\min}(Q)$, and $\lambda_{\mathrm{pmin}}(Q)$ denote the largest, smallest, and smallest nonzero (and hence positive) eigenvalues of Q, respectively. We use the notation $L_n(\beta) := \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\beta\|_2^2$ for the least squares loss function.

2. LS-Boost(ε) : Computational Guarantees and Statistical Implications.

Roadmap. We begin our formal study by examining the LS-Boost(ε) algorithm. We study the rate at which the coefficients generated by LS-Boost(ε) converge to the set of unregularized least square solutions. This characterizes the amount of data-fidelity as a function of the number of iterations and ε. In particular, we show (global) linear convergence of the regression coefficients to the set of least squares coefficients, with similar convergence rates derived for the prediction estimates and the boosting training errors delivered by LS-Boost(ε) . We also present bounds on the shrinkage of the regression coefficients $\hat{\beta}^k$ as a function of k and ε, thereby describing how the amount of shrinkage of the regression coefficients changes as a function of the number of iterations k.

We present below a formal description of LS-Boost(ε) following [22]:

Algorithm: Least Squares Boosting – LS-Boost(ε)

Fix the learning rate $\varepsilon > 0$, the number of iterations M, and initialize $\hat{\beta}^0 = 0$ and $\hat{r}^0 = \mathbf{y}$.
1. For $0 \leq k \leq M$ do the following:
2. Select the covariate index j_k and $\tilde{u}_{j_k}$ as follows:

$$j_k \in \operatorname*{arg\,min}_{1 \leq m \leq p} \sum_{i=1}^{n} (\hat{r}_i^k - x_{im}\tilde{u}_m)^2, \text{ where, } \tilde{u}_m = \operatorname*{arg\,min}_{u \in \mathbb{R}} \left(\sum_{i=1}^{n} (\hat{r}_i^k - x_{im}u)^2 \right),$$

for $m = 1, \ldots, p$.
3. Update the regression coefficients and residuals as:

$$\hat{\beta}_{j_k}^{k+1} \leftarrow \hat{\beta}_{j_k}^{k} + \varepsilon \tilde{u}_{j_k}, \quad \hat{\beta}_{j}^{k+1} \leftarrow \hat{\beta}_{j}^{k} \ , j \neq j_k \ , \quad \text{and} \quad \hat{r}^{k+1} \leftarrow \hat{r}^k - \varepsilon \mathbf{X}_{j_k} \tilde{u}_{j_k} \ .$$

A special instance of the LS-Boost(ε) algorithm with $\varepsilon = 1$ is known as LS-Boost [22] or Forward Stagewise [33] — it is essentially a method of repeated simple least squares fitting of the residuals [7]. In the signal processing literature LS-Boost is known as Matching Pursuit [36], a scheme used to approximate a signal (herein, response) as a sparse linear sum of dictionary elements (herein, features). In words, the LS-Boost algorithm at the k-th iteration determines the covariate index j_k resulting in the maximal decrease in the univariate regression fit to the current residuals. If $\mathbf{X}_{j_k}\tilde{u}_{j_k}$ denotes the *best* univariate fit for the current residuals, LS-Boost updates the residuals: $\hat{r}^{k+1} \leftarrow \hat{r}^k - \mathbf{X}_{j_k}\tilde{u}_{j_k}$ and the j_k-th regression coefficient: $\hat{\beta}_{j_k}^{k+1} \leftarrow \hat{\beta}_{j_k}^{k} + \tilde{u}_{j_k}$, in an attempt to minimize the training error. LS-Boost(ε) has old roots – as noted by [7], LS-Boost with $M = 2$ is known as "twicing," a method proposed by Tukey [50]. The papers [8, 6, 7] present very interesting perspectives on LS-Boost(ε), where they refer to the algorithm as $L2$-Boost. [7] also obtains approximate expressions for the effective degrees of freedom of the $L2$-Boost algorithm. LS-Boost(ε) is also closely related to Friedman's MART algorithm [21]. LS-Boost(ε) is a slow-learning variant of LS-Boost, which diminishes the fast and greedy learning style of LS-Boost with an additional damping factor of ε, which consequently leads to a richer family of models, as we study in this section.

2.1. *Computational Guarantees and Intuition.* We first review some useful properties associated with the familiar least squares regression problem:

$$\begin{array}{llll} \text{LS}: & L_n^* := & \min\limits_{\beta} & L_n(\beta) := \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 \\ & & \text{s.t.} & \beta \in \mathbb{R}^p \ , \end{array} \tag{2.1}$$

where $L_n(\cdot)$ is the least squares loss, whose gradient is:

$$\nabla L_n(\beta) = -\tfrac{1}{n}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = -\tfrac{1}{n}\mathbf{X}^T r \tag{2.2}$$

where $r = \mathbf{y} - \mathbf{X}\beta$ is the vector of residuals corresponding to the regression coefficients β. It follows that β is a least-squares solution of LS if and only if $\nabla L_n(\beta) = 0$, which leads to the well known normal equations:

$$0 = -\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = -\mathbf{X}^T r \ . \tag{2.3}$$

It also holds that:

$$n \cdot \|\nabla L_n(\beta)\|_\infty = \|\mathbf{X}^T r\|_\infty = \max_{j \in \{1,\ldots,p\}} \{|r^T \mathbf{X}_j|\} \ . \tag{2.4}$$

The following theorem describes precise computational guarantees for LS-Boost(ε): linear convergence of LS-Boost(ε) with respect to (2.1), and bounds on the ℓ_1 shrinkage of the coefficients produced. Note that the theorem uses the quantity $\lambda_{\mathrm{pmin}}(\mathbf{X}^T\mathbf{X})$ which denotes the smallest nonzero (and hence positive) eigenvalue of $\mathbf{X}^T\mathbf{X}$.

Theorem 2.1. **(Linear Convergence of LS-Boost(ε) for Least Squares)** *Consider the* LS-Boost(ε) *algorithm with learning rate $\varepsilon \in (0, 1]$, and define the linear convergence rate coefficient γ:*

$$\gamma := \left(1 - \frac{\varepsilon(2-\varepsilon)\lambda_{\mathrm{pmin}}(\mathbf{X}^T\mathbf{X})}{4p}\right) < 1 \ . \tag{2.5}$$

For all $k \geq 0$ the following bounds hold:

(i) (training error): $L_n(\hat{\beta}^k) - L_n^* \leq \frac{1}{2n}\|\mathbf{X}\hat{\beta}_{LS}\|_2^2 \cdot \gamma^k$

(ii) (regression coefficients): there exists a least squares solution $\hat{\beta}_{LS}^k$ *such that:*

$$\|\hat{\beta}^k - \hat{\beta}_{LS}^k\|_2 \leq \frac{\|\mathbf{X}\hat{\beta}_{LS}\|_2}{\sqrt{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}} \cdot \gamma^{k/2}$$

(iii) (predictions): for every least-squares solution $\hat{\beta}_{LS}$ *it holds that*

$$\|\mathbf{X}\hat{\beta}^k - \mathbf{X}\hat{\beta}_{LS}\|_2 \leq \|\mathbf{X}\hat{\beta}_{LS}\|_2 \cdot \gamma^{k/2}$$

(iv) (gradient norm/correlation values): $\|\nabla L_n(\hat{\beta}^k)\|_\infty = \frac{1}{n}\|\mathbf{X}^T\hat{r}^k\|_\infty \leq \frac{1}{n}\|\mathbf{X}\hat{\beta}_{LS}\|_2 \cdot \gamma^{k/2}$

(v) (ℓ_1-shrinkage of coefficients):

$$\|\hat{\beta}^k\|_1 \leq \min\left\{\sqrt{k}\sqrt{\tfrac{\varepsilon}{2-\varepsilon}}\sqrt{\|\mathbf{X}\hat{\beta}_{LS}\|_2^2 - \|\mathbf{X}\hat{\beta}_{LS} - \mathbf{X}\hat{\beta}^k\|_2^2}\ \ ,\ \frac{\varepsilon\|\mathbf{X}\hat{\beta}_{LS}\|_2}{1-\sqrt{\gamma}}\left(1-\gamma^{k/2}\right)\right\}$$

(vi) (sparsity of coefficients): $\|\hat{\beta}^k\|_0 \leq k$. □

Before remarking on the various parts of Theorem 2.1, we first discuss the quantity γ defined in (2.5), which is called the linear convergence rate coefficient. We can write $\gamma = 1 - \frac{\varepsilon(2-\varepsilon)}{4\kappa(\mathbf{X}^T\mathbf{X})}$ where $\kappa(\mathbf{X}^T\mathbf{X})$ is defined to be the ratio $\kappa(\mathbf{X}^T\mathbf{X}) := \frac{p}{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}$. Note that $\kappa(\mathbf{X}^T\mathbf{X}) \in [1, \infty)$. To see this, let $\tilde{\beta}$ be an eigenvector associated with the largest eigenvalue of $\mathbf{X}^T\mathbf{X}$, then:

$$0 < \lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X}) \leq \lambda_{\max}(\mathbf{X}^T\mathbf{X}) = \frac{\|\mathbf{X}\tilde{\beta}\|_2^2}{\|\tilde{\beta}\|_2^2} \leq \frac{\|\mathbf{X}\|_{1,2}^2\|\tilde{\beta}\|_1^2}{\|\tilde{\beta}\|_2^2} \leq p \ , \tag{2.6}$$

where the last inequality uses our assumption that the columns of $\mathbf{X}$ have been normalized (whereby $\|\mathbf{X}\|_{1,2} = 1$), and the fact that $\|\tilde{\beta}\|_1 \leq \sqrt{p}\|\tilde{\beta}\|_2$. This then implies that $\gamma \in [0.75, 1.0)$ – independent of any assumption on the dataset – and most importantly it holds that $\gamma < 1$.

Let us now make the following immediate remarks on Theorem 2.1:

• The bounds in parts *(i)-(iv)* state that the training errors, regression coefficients, predictions, and correlation values produced by LS-Boost(ε) converge linearly (also known as geometric or exponential convergence) to their least squares counterparts: they decrease by at least the constant multiplicative factor $\gamma < 1$ for part *(i)*, and by $\sqrt{\gamma}$ for parts *(ii)-(iv)*, at every iteration. The bounds go to zero at this linear rate as $k \to \infty$.

• The computational guarantees in parts *(i)* - *(vi)* provide characterizations of the data-fidelity and shrinkage of the LS-Boost(ε) algorithm for any given specifications of the learning rate ε and the number of boosting iterations k. Moreover, the quantities appearing in the bounds can be computed from simple characteristics of the data that can be obtained *a priori* without even running the boosting algorithm. (And indeed, one can even substitute $\|\mathbf{y}\|_2$ in place of $\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2$ throughout the bounds if desired since $\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2 \leq \|\mathbf{y}\|_2$.)

Some Intuition Behind Theorem 2.1. Let us now study the LS-Boost(ε) algorithm and build intuition regarding its progress with respect to solving the unconstrained least squares problem (2.1), which will inform the results in Theorem 2.1. Since the predictors are all standardized to have unit ℓ_2 norm, it follows that the coefficient index j_k and corresponding step-size $\tilde{u}_{j_k}$ selected by LS-Boost(ε) satisfy:

$$j_k \in \underset{j \in \{1,\ldots,p\}}{\arg\max} |(\hat{r}^k)^T \mathbf{X}_j| \qquad \text{and} \qquad \tilde{u}_{j_k} = (\hat{r}^k)^T \mathbf{X}_{j_k} \; . \tag{2.7}$$

Combining (2.4) and (2.7), we see that

$$|\tilde{u}_{j_k}| = |(\hat{r}^k)^T \mathbf{X}_{j_k}| = n \cdot \|\nabla L_n(\hat{\beta}^k)\|_\infty \; . \tag{2.8}$$

Using the formula for $\tilde{u}_{j_k}$ in (2.7), we have the following convenient way to express the change in residuals at each iteration of LS-Boost(ε):

$$\hat{r}^{k+1} = \hat{r}^k - \varepsilon \left((\hat{r}^k)^T \mathbf{X}_{j_k} \right) \mathbf{X}_{j_k} \; . \tag{2.9}$$

Intuitively, since (2.9) expresses $\hat{r}^{k+1}$ as the difference of two correlated variables, $\hat{r}^k$ and $\mathrm{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k})\mathbf{X}_{j_k}$, we expect the squared ℓ_2 norm of $\hat{r}^{k+1}$ to be smaller than that of $\hat{r}^k$. On the other hand, as we will see from (3.2), convergence of the residuals is ensured by the dependence of the change in residuals on $|(\hat{r}^k)^T \mathbf{X}_{j_k}|$, which goes to 0 as we approach a least squares solution. In the proof of Theorem 2.1 in Section A.2.2 (in [19]) we make this intuition precise by using (2.9) to quantify the amount of decrease in the least squares objective function at each iteration of LS-Boost(ε). The final ingredient of the proof uses properties of convex quadratic functions (Section A.2.1 in [19]) to relate the exact amount of the decrease from iteration k to $k+1$ to the current optimality gap $L_n(\hat{\beta}^k) - L_n^*$, which yields the following strong linear convergence property:

$$L_n(\hat{\beta}^{k+1}) - L_n^* \; \leq \; \gamma \cdot (L_n(\hat{\beta}^k) - L_n^*) \; . \tag{2.10}$$

The above states that the training error gap decreases at each iteration by at least the multiplicative factor of γ, and clearly implies item *(i)* of Theorem 2.1. The bounds in Theorem 2.1 are indeed tight, as addressed in Section A.2.8 [19].

Comments on the global linear convergence rate in Theorem 2.1. The global linear convergence of LS-Boost(ε) proved in Theorem 2.1, while novel, is not at odds with the present understanding of such convergence for optimization problems. One can view LS-Boost(ε) as performing steepest descent optimization steps with respect to the ℓ_1 norm unit ball (rather than the ℓ_2 norm unit ball which is the canonical version of the steepest descent method, see [41]). It is known [41] that canonical steepest decent exhibits global linear convergence for convex quadratic optimization so long as the Hessian matrix Q of the quadratic objective function is positive definite, i.e., $\lambda_{\min}(Q) > 0$. And for the least squares loss function $Q = \frac{1}{n}\mathbf{X}^T\mathbf{X}$, which yields the condition that $\lambda_{\min}(\mathbf{X}^T\mathbf{X}) > 0$. As discussed in [3], this result extends to other norms defining steepest descent as well. Hence what is modestly surprising herein is not the linear convergence *per se*, but rather that LS-Boost(ε) exhibits global linear convergence even when $\lambda_{\min}(\mathbf{X}^T\mathbf{X}) = 0$, i.e., even when $\mathbf{X}$ does not have full column rank (essentially replacing $\lambda_{\min}(\mathbf{X}^T\mathbf{X})$ with $\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})$ in our analysis). This derives specifically from the structure of the least squares loss function, whose function values (and whose gradient) are invariant in the null space of $\mathbf{X}$, i.e., $L_n(\beta + d) = L_n(\beta)$ for all d satisfying $\mathbf{X}d = 0$, and is thus rendered "immune" to changes in β in the null space of $\mathbf{X}^T\mathbf{X}$.

2.2. *Statistical Insights from the Computational Guarantees.* Note that in most noisy problems, the limiting least squares solution is statistically less interesting than an estimate obtained in the interior of the boosting profile, since the latter typically corresponds to a model with better bias-variance tradeoff. We thus caution the reader that the bounds in Theorem 2.1 should *not* be merely interpreted as statements about how rapidly the boosting iterations reach the least squares fit. We rather intend for these bounds to inform us about the *evolution* of the training errors and the amount of shrinkage of the coefficients as the LS-Boost(ε) algorithm progresses and when k is at most moderately large. When the training errors are paired with the profile of the ℓ_1-shrinkage values of the regression coefficients, they lead to the ordered pairs:

$$\left(\frac{1}{2n}\|\mathbf{y} - \mathbf{X}\hat{\beta}^k\|_2^2\ ,\ \|\hat{\beta}^k\|_1\right), \quad k \geq 1\ , \tag{2.11}$$

which describes the data-fidelity and ℓ_1-shrinkage tradeoff as a function of k, for the given learning rate $\varepsilon > 0$. This profile is described in Figure A.1 in Section A.1.1 [19] for several data instances. The bounds in Theorem 2.1 provide estimates for the two components of the ordered pair (2.11), and they can be computed prior to running the boosting algorithm. For simplicity, let us use the following crude estimate:

$$\ell_k := \min\left\{\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2\sqrt{\frac{k\varepsilon}{2-\varepsilon}} \;\;,\;\; \frac{\varepsilon\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2}{1-\sqrt{\gamma}}\left(1-\gamma^{\frac{k}{2}}\right)\right\},$$

which is an upper bound of the bound in part *(v)* of the theorem, to provide an upper approximation of $\|\hat{\beta}_k\|_1$. Combining the above estimate with the guarantee in part *(i)* of Theorem 2.1 in (2.11), we obtain the following ordered pairs:

$$\left(\frac{1}{2n}\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2^2\cdot\gamma^k + L_n^* \;\;,\;\; \ell_k\right), \quad k \geq 1\ , \tag{2.12}$$

which describe the *entire* profile of the training error bounds and the ℓ_1-shrinkage bounds as a function of k as suggested by Theorem 2.1. These profiles, as described above in (2.12), are illustrated in Figure 3.

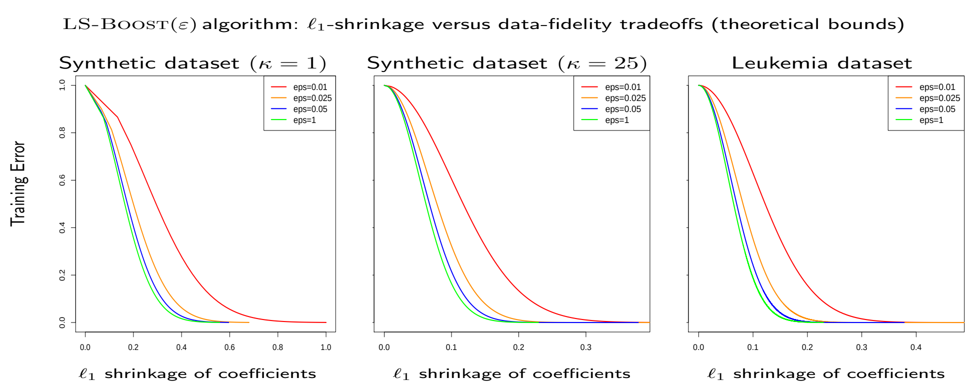

FIG 3. *Figure showing profiles of ℓ_1 shrinkage of the regression coefficients versus training error for the* LS-Boost(ε) *algorithm, for different values of the learning rate ε (denoted by the moniker "eps" in the legend). The profiles have been obtained from the computational bounds in Theorem 2.1. The left and middle panels correspond to synthetic values of the ratio $\kappa = \frac{p}{\lambda_{pmin}}$, and for the right panel profiles the value of κ (here, $\kappa = 270.05$) is extracted from the Leukemia dataset, described in Section 6. The vertical axes have been normalized so that the training error at $k = 0$ is one, and the horizontal axes have been scaled to the unit interval.*

It is interesting to consider the profiles of Figure 3 alongside the *explicit* regularization framework of the Lasso (1.1) which also traces out a profile of the form (2.11), namely:

$$\left(\frac{1}{2n}\|\mathbf{y}-\mathbf{X}\hat{\beta}^*_\delta\|_2^2 \ \ , \ \ \|\hat{\beta}^*_\delta\|_1\right), \quad \delta \geq 0 \ , \tag{2.13}$$

as a function of δ, where, $\hat{\beta}^*_\delta$ is a solution to the LASSO problem (1.1). For a value of $\delta := \ell_k$ the optimal objective value of the LASSO problem will serve as a lower bound of the corresponding LS-BOOST(ε) loss function value at iteration k. Thus the training error of $\hat{\beta}^k$ delivered by the LS-BOOST(ε) algorithm will be sandwiched between the following lower and upper bounds:

$$L_{i,k} := \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\hat{\beta}^*_{\ell_k}\|_2^2 \leq \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\hat{\beta}^k\|_2^2 \leq \frac{1}{2n}\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2^2 \cdot \gamma^k + L_n^* =: U_{i,k}$$

for every k. Note that the difference between the upper and lower bounds above, given by $U_{i,k} - L_{i,k}$, converges to zero as $k \to \infty$. Figure A.1, Section A.1.1 in [19] shows the training error versus shrinkage profiles for LS-BOOST(ε) and LASSO solutions, for different datasets.

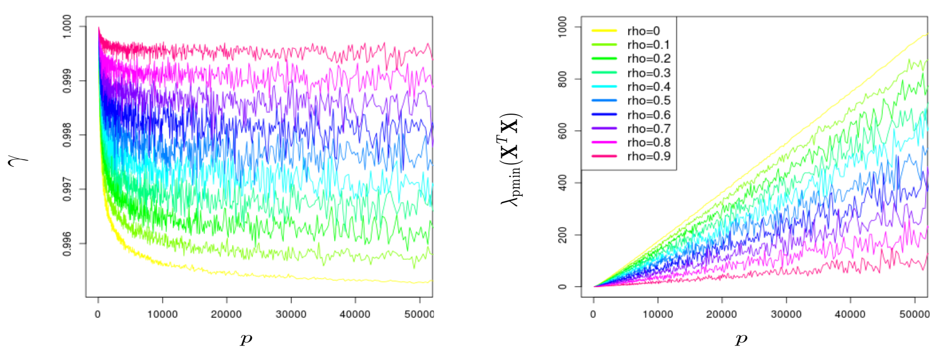

FIG 4. *Figure showing the behavior of γ [left panel] and $\lambda_{pmin}(\mathbf{X}^T\mathbf{X})$ [right panel] for different values of ρ (denoted by the moniker "rho" in the legend) and p, with $\varepsilon = 1$. There are ten profiles in each panel corresponding to different values of ρ for $\rho = 0,\ 0.1,\ \ldots,\ 0.9$. Each profile documents the change in γ as a function of p, the smallest value of p appearing in the display is $p = 73$. Here, the data matrix $\mathbf{X}$ is comprised of $n = 50$ samples from a p-dimensional multivariate Gaussian distribution with mean zero, and all pairwise correlations equal to ρ, and the features are then standardized to have unit ℓ_2 norm. The left panel shows that γ exhibits a phase of rapid decay (as a function of p) after which it stabilizes into the regime of* fastest *convergence. Interestingly, the behavior shows a monotone trend in ρ: the rate of progress of* LS-BOOST(ε) *becomes slower for larger values of ρ and faster for smaller values of ρ.*

For the bounds in parts *(i)* and *(iii)* of Theorem 2.1, the asymptotic limits (as $k \to \infty$) are the unregularized least squares training error and predictions – which are quantities that are uniquely defined even in the underdetermined case.

The bound in part *(ii)* of Theorem 2.1 is a statement concerning the regression coefficients. In this case, the notion of convergence needs to be appropriately modified from parts *(i)* and *(iii)*, since the *natural* limiting object $\hat{\beta}_{\text{LS}}$ is not necessarily unique. In this case, perhaps not surprisingly, the regression coefficients $\hat{\beta}^k$ need not converge. The result in part *(ii)* of the theorem states that $\hat{\beta}^k$ converges at a linear rate to the *set* of least squares solutions. In other words, at every LS-Boost(ε) boosting iteration, there exists a least squares solution $\hat{\beta}^k_{\text{LS}}$ for which the presented bound holds. Here $\hat{\beta}^k_{\text{LS}}$ is in fact the closest least squares solution to $\hat{\beta}^k$ in the ℓ_2 norm – and the particular candidate least squares solution $\hat{\beta}^k_{\text{LS}}$ may be different for each iteration.

Interpreting the parameters and algorithm dynamics. There are several determinants of the quality of the bounds in the different parts of Theorem 2.1 which can be grouped into:

- algorithmic parameters: this includes the learning rate ε and the number of iterations k, and
- data dependent quantities: $\|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2$, $\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})$, and p.

The coefficient of linear convergence is given by the quantity $\gamma := 1 - \frac{\varepsilon(2-\varepsilon)}{4\kappa(\mathbf{X}^T\mathbf{X})}$, where $\kappa(\mathbf{X}^T\mathbf{X}) := \frac{p}{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}$. Note that γ is monotone decreasing in ε for $\varepsilon \in (0,1]$, and is minimized at $\varepsilon = 1$. This simple observation confirms the general intuition about LS-Boost(ε): $\varepsilon = 1$ corresponds to the most aggressive model fitting behavior in the LS-Boost(ε) family, with smaller values of ε corresponding to a slower model fitting process. The ratio $\kappa(\mathbf{X}^T\mathbf{X})$ is a close cousin of the condition number associated with the data matrix $\mathbf{X}$ – and smaller values of $\kappa(\mathbf{X}^T\mathbf{X})$ imply a faster rate of convergence.

In the overdetermined case with $n \geq p$ and $\text{rank}(\mathbf{X}) = p$, the condition number $\bar{\kappa}(\mathbf{X}^T\mathbf{X}) := \frac{\lambda_{\max}(\mathbf{X}^T\mathbf{X})}{\lambda_{\min}(\mathbf{X}^T\mathbf{X})}$ plays a key role in determining the stability of the least-squares solution $\hat{\beta}_{\text{LS}}$ and in measuring the degree of multicollinearity present. Note that $\bar{\kappa}(\mathbf{X}^T\mathbf{X}) \in [1,\infty)$, and that the problem is better conditioned for smaller values of this ratio. Furthermore, since $\text{rank}(\mathbf{X}) = p$ it holds that $\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X}) = \lambda_{\min}(\mathbf{X}^T\mathbf{X})$, and thus $\bar{\kappa}(\mathbf{X}^T\mathbf{X}) \leq \kappa(\mathbf{X}^T\mathbf{X})$ by (2.6). Thus the condition number $\kappa(\mathbf{X}^T\mathbf{X})$ always upper bounds the classical condition number $\bar{\kappa}(\mathbf{X}^T\mathbf{X})$, and if $\lambda_{\max}(\mathbf{X}^T\mathbf{X})$ is close to p, then $\bar{\kappa}(\mathbf{X}^T\mathbf{X}) \approx \kappa(\mathbf{X}^T\mathbf{X})$ and the two measures essentially coincide. Finally, since in this setup $\hat{\beta}_{\text{LS}}$ is unique, part *(ii)* of Theorem 2.1 implies that the sequence $\{\hat{\beta}^k\}$ converges linearly to the unique least squares solution $\hat{\beta}_{\text{LS}}$.

In the underdetermined case with $p > n$, $\lambda_{\min}(\mathbf{X}^T\mathbf{X}) = 0$ and thus $\bar{\kappa}(\mathbf{X}^T\mathbf{X}) = \infty$. On the other hand, $\kappa(\mathbf{X}^T\mathbf{X}) < \infty$ since $\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})$ is the smallest *nonzero* (hence positive) eigenvalue of $\mathbf{X}^T\mathbf{X}$. Therefore the condition number $\kappa(\mathbf{X}^T\mathbf{X})$ is similar to the classical condition number $\bar{\kappa}(\cdot)$ restricted to the subspace $\mathcal{S}$ spanned by the columns of $\mathbf{X}$ (whose dimension is $\text{rank}(\mathbf{X})$). Interestingly, the linear rate of convergence enjoyed by LS-Boost(ε) is in a sense adaptive – the algorithm automatically adjusts itself to the convergence rate dictated by the parameter γ "as if" it knows that the null space of $\mathbf{X}$ is not relevant.

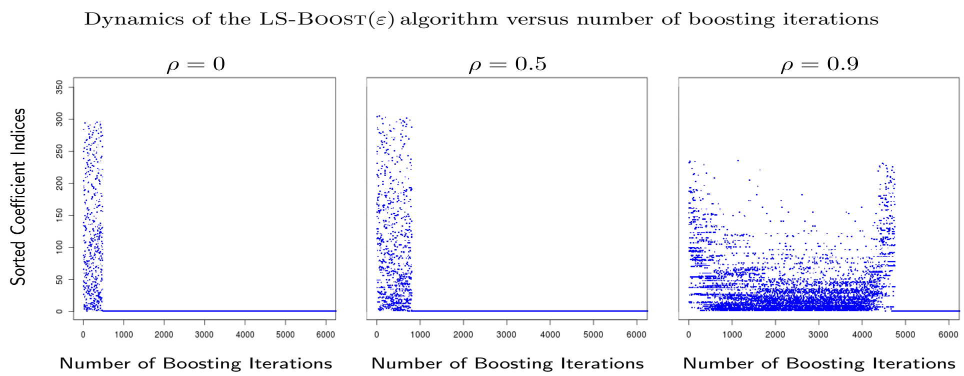

Fig 5. *Showing the* LS-Boost(ε) *algorithm run on the same synthetic dataset as was used in Figure 1 with $\varepsilon = 1$, for three different values of the pairwise correlation ρ. A point is "on" if the corresponding regression coefficient is updated at iteration k. Here the vertical axes have been reoriented so that the coefficients that are updated the maximum number of times appear lower on the axes. For larger values of ρ, we see that the* LS-Boost(ε) *algorithm aggressively updates the coefficients for a large number of iterations, whereas the dynamics of the algorithm for smaller values of ρ are less pronounced. For larger values of ρ the* LS-Boost(ε) *algorithm takes longer to reach the least squares fit and this is reflected in the above figure from the update patterns in the regression coefficients. The dynamics of the algorithm evident in this figure nicely complements the insights gained from Figure 1.*

As the dataset is varied, the value of γ can change substantially from one dataset to another, thereby leading to differences in the convergence behavior bounds in parts *(i)*-*(v)* of Theorem 2.1. To settle all of these ideas, we can derive some simple bounds on γ using tools from random matrix theory. Towards this end, let us suppose that the entries of $\mathbf{X}$ are drawn from a standard Gaussian ensemble, which are subsequently standardized such that every column of $\mathbf{X}$ has unit ℓ_2 norm. Then it follows from random matrix theory [51] that $\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X}) \gtrapprox \frac{1}{n}(\sqrt{p}-\sqrt{n})^2$ with high probability. (See Section A.2.4 in [19] for a more detailed discussion of this fact.) To gain better insights into the behavior of γ and how it depends on the values of pairwise correlations of the features, we performed some computational experiments, the results of which are shown in Figure 4. Figure 4 shows the behavior of γ as a function of p for a fixed $n = 50$ and $\varepsilon = 1$, for different datasets $\mathbf{X}$ simulated as follows. We first generated a multivariate data matrix from a Gaussian distribution with mean zero and covariance $\Sigma_{p\times p} = (\sigma_{ij})$, where, $\sigma_{ij} = \rho$ for all $i \neq j$; and then all of the columns of the data matrix were standardized to have unit ℓ_2 norm. The resulting matrix was taken as $\mathbf{X}$. We considered different cases by varying the magnitude of pairwise correlations of the features ρ – when ρ is small, the rate of convergence is typically faster (smaller γ) and the rate becomes slower (higher γ) for higher values of ρ. Figure 4 shows that the coefficient of linear convergence γ is quite close to 1.0 – which suggests a slowly converging algorithm and confirms our intuition about the algorithmic behavior of LS-Boost(ε) . Indeed, LS-Boost(ε) , like any other boosting algorithm, should indeed converge slowly to the unregularized least squares solution. The slowly converging nature of the LS-Boost(ε) algorithm provides, for the first time, a precise theoretical justification of the empirical observation made in [33] that stagewise regression is widely considered ineffective as a tool to obtain the unregularized least squares fit, as compared to other stepwise model fitting procedures like Forward Stepwise regression (discussed in Section 1).

The above discussion sheds some interesting insight into the behavior of the LS-Boost(ε) algorithm. For larger values of ρ, the observed covariates tend to be even more highly correlated (since $p \gg n$). Whenever a pair of features are highly correlated, the LS-Boost(ε) algorithm finds it *difficult* to prefer one over the other and thus takes turns in updating both coefficients, thereby distributing the effects of a covariate to all of its correlated cousins. Since a group of correlated covariates are all competing to be updated by the LS-Boost(ε) algorithm, the progress made by the algorithm in decreasing the loss function is naturally slowed down. In contrast, when ρ is small, the LS-Boost(ε) algorithm brings in a covariate and in a sense completes the process by doing the exact line-search on that feature. This heuristic explanation attempts to explain the slower rate of convergence of the LS-Boost(ε) algorithm for large values of ρ – a phenomenon that we observe in practice and which is also substantiated by the computational guarantees in Theorem 2.1. We refer the reader to Figures 1 and 5 which further illustrate the above justification. Statement *(v)* of Theorem 2.1 provides upper bounds on the ℓ_1 shrinkage of the coefficients. Figure 3 illustrates the evolution of the data-fidelity versus ℓ_1-shrinkage as obtained from the computational bounds in Theorem 2.1. Some additional discussion and properties of LS-Boost(ε) are presented in Supplementary Material Section A.2.3, [19].

We briefly discuss computational guarantees for Forward Stepwise regression (also known as Forward Selection) [33, 14, 52]. The Forward Stepwise algorithm, at the k-th iteration, selects a covariate $\mathbf{X}_{j_k}$ maximally correlated with the current residual[4]: $j_k \in \arg\max_j |(\hat{r}^k)^T \mathbf{X}_j|$. If the set of active (i.e., nonzero) regression coefficient indices at iteration k is denoted by $\mathcal{I}_k$, then the algorithm appends j_k to this set, $\mathcal{I}_{k+1} \leftarrow \mathcal{I}_k \cup \{j_k\}$, and updates the regression coefficients by performing a *joint* least squares regression of $\mathbf{y}$ restricted to the covariates in $\mathcal{I}_{k+1}$:

$$\hat{\beta}^{k+1} \in \arg\min_{\beta} \quad \|\mathbf{y} - \mathbf{X}\beta\|_2^2 \quad \text{s.t.} \quad \beta_i = 0, \forall i \notin \mathcal{I}_{k+1}.$$

Notice that the least-squares loss function value for Forward Stepwise is bounded by the corresponding value if one was instead using LS-Boost, i.e., $L_n(\hat{\beta}^{k+1}) \leq L_n(\hat{\beta}^k + \tilde{u}_{j_k} e_{j_k})$. Thus it is straightforward to show that parts *(i)-(iv)* (and also part *(vi)*) of Theorem 2.1 (with $\varepsilon = 1$) hold in their current form for Forward Stepwise as well. Interestingly, part *(v)* does not hold for Forward Stepwise nor should we expect Forward Stepwise – which is the most aggressive model fitting procedure described herein – to have any such guarantees about the shrinkage of coefficients. In the signal processing literature, Forward Stepwise is popularly known as Orthogonal Matching Pursuit [40].

3. Boosting Algorithms as Subgradient Descent.

Roadmap. In this section we present a new unifying framework for interpreting the three boosting algorithms that were discussed in Section 1, namely FS_ε, its non-uniform learning rate extension $\text{FS}_{\varepsilon_k}$, and LS-Boost($\varepsilon$). We show herein that all three algorithmic families can be interpreted as instances of the subgradient descent method of convex optimization, applied to the problem of minimizing the largest correlation between residuals and predictors. Interestingly, this unifying lens will also result in a natural generalization of FS_ε with very strong ties to the Lasso problem and its solution, as we will present in Sections 4 and 5. The framework presented in this section leads to convergence guarantees for FS_ε and $\text{FS}_{\varepsilon_k}$. In Theorem 3.1 herein, we present a theoretical description of the evolution of the FS_ε algorithm, in terms of its data-fidelity and shrinkage guarantees as a function of the number of boosting iterations. These results are a consequence of the computational guarantees for FS_ε that inform us about the rate at which the FS_ε training error, regression coefficients, and predictions make their way to their least squares counterparts. In order to develop these results, we

[4]Recall that we assume all the covariates to have unit ℓ_2 norm

first motivate and briefly review the subgradient descent method of convex optimization.

3.1. *Boosting Algorithms FS_ε*, LS-Boost(ε) *and FS_{ε_k}*. We present a formal description of the FS_ε algorithm introduced in Section 1.

Algorithm: Incremental Forward Stagewise Regression – FS_ε

Fix the learning rate $\varepsilon > 0$, the number of iterations M, and initialize $\hat{\beta}^0 = 0$ and $\hat{r}^0 = \mathbf{y}$.
1. For $0 \leq k \leq M$ do the following:
2. Compute $j_k \in \arg\max_{j \in \{1,\ldots,p\}} |(\hat{r}^k)^T \mathbf{X}_j|$
3. Update the regression coefficients and residuals as:

$$\begin{aligned} \hat{\beta}^{k+1}_{j_k} &\leftarrow \hat{\beta}^k_{j_k} + \varepsilon \ \text{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}), \ \hat{\beta}^{k+1}_j \leftarrow \hat{\beta}^k_j \ , j \neq j_k \ , \ \text{and} \\ \hat{r}^{k+1} &\leftarrow \hat{r}^k - \varepsilon \ \text{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k} \end{aligned} \tag{3.1}$$

The FS_ε algorithm in the k-th iteration greedily chooses a covariate $\mathbf{X}_{j_k}$ that is the most correlated (in absolute value) with the current residual and updates the j_k-th regression coefficient, along with the residuals, with a shrinkage factor ε. Firstly, since all of the covariates are standardized to have unit ℓ_2 norm, for same given residual value $\hat{r}^k$ it is simple to derive that LS-Boost(ε) and FS_ε lead to the same choice of j_k. Qualitatively speaking, as in the case of LS-Boost(ε), a smaller value of ε corresponds to a slower learning procedure. However, in contrast to LS-Boost(ε), where, ε lies naturally in $(0, 1]$, the choice of ε in FS_ε is more sensitive to the scale of the problem. Indeed LS-Boost(ε) and FS_ε inspite of their similarities contain subtle differences, for example, in their residual updates:

$$\begin{aligned} \text{LS-Boost}(\varepsilon): \quad & \|\hat{r}^{k+1} - \hat{r}^k\|_2 = \varepsilon |(\hat{r}^k)^T \mathbf{X}_{j_k}| = \varepsilon \cdot n \cdot \|\nabla L_n(\hat{\beta}^k)\|_\infty \\ \text{FS}_\varepsilon: \quad & \|\hat{r}^{k+1} - \hat{r}^k\|_2 = \varepsilon |s_k| \quad \text{where} \ \ s_k = \text{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \ , \end{aligned} \tag{3.2}$$

where $\nabla L_n(\cdot)$ is the gradient of $L_n(\beta)$. Note that for both of the algorithms, the quantity $\|\hat{r}^{k+1} - \hat{r}^k\|_2$ involves the shrinkage factor ε. Their difference thus lies in the multiplicative factor, which is $n \cdot \|\nabla L_n(\hat{\beta}^k)\|_\infty$ for LS-Boost(ε) and is $|\text{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k})|$ for FS_ε. The norm of the successive residual differences for LS-Boost(ε) is proportional to the ℓ_∞ norm of the gradient of the least squares loss function (see herein equations (2.2) and (2.4)). For FS_ε, the norm of the successive residual differences depends on the absolute value of the sign of the j_k-th coordinate of the gradient. Note that

$s_k \in \{-1, 0, 1\}$ depending upon whether $(\hat{r}^k)^T \mathbf{X}_{j_k}$ is negative, zero, or positive; and $s_k = 0$ only when $(\hat{r}^k)^T \mathbf{X}_{j_k} = 0$, i.e., only when $\|\nabla L_n(\hat{\beta}^k)\|_\infty = 0$ and hence $\hat{\beta}^k$ is a least squares solution. Thus, for FS_ε the ℓ_2 norm of the difference in residuals is almost always ε during the course of the algorithm. For the LS-Boost(ε) algorithm, progress is considerably more sensitive to the norm of the gradient — as the algorithm makes its way to the unregularized least squares fit, one should expect the norm of the gradient to also shrink to zero, as we have established formally in Section 2. Qualitatively speaking, this means that the updates of LS-Boost(ε) are more well-behaved when compared to the updates of FS_ε, which are more erratically behaved. Of course, the additional shrinkage factor ε further dampens the progress for both algorithms.

While Section 2 shows that the predicted values $\mathbf{X}\hat{\beta}^k$ obtained from LS-Boost(ε) converge (at a globally linear rate) to the least squares fit as $k \to \infty$ (for any value of $\varepsilon \in (0, 1]$); on the other hand, for FS_ε with $\varepsilon > 0$, the iterates $\mathbf{X}\hat{\beta}^k$ need not necessarily converge to the least squares fit as $k \to \infty$. Indeed, the FS_ε algorithm, by its operational definition, has a uniform learning rate ε which remains fixed for all iterations; this makes it impossible to always guarantee convergence to a least squares solution with accuracy less than $O(\varepsilon)$. We show in this section that the predictions from the FS_ε algorithm converges to an approximate least squares solution, albeit at a global sublinear rate.[5]

Since the main difference between FS_ε and LS-Boost(ε) lies in the choice of the step-size used to update the coefficients, let us therefore consider a non-constant step-size/non-uniform learning rate version of FS_ε, which we call $\text{FS}_{\varepsilon_k}$. $\text{FS}_{\varepsilon_k}$ replaces update (3.1) of FS_ε by:

residual update: $\hat{r}^{k+1} \leftarrow \hat{r}^k - \varepsilon_k \, \text{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k}$

coefficient update: $\hat{\beta}^{k+1}_{j_k} \leftarrow \hat{\beta}^k_{j_k} + \varepsilon_k \, \text{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k})$ and $\hat{\beta}^{k+1}_j \leftarrow \hat{\beta}^k_j, j \neq j_k$,

where $\{\varepsilon_k\}$ is a sequence of learning-rates (or step-sizes) which depend upon the iteration index k. LS-Boost(ε) can thus be thought of as a version of $\text{FS}_{\varepsilon_k}$, where the step-size ε_k is given by $\varepsilon_k := \varepsilon \tilde{u}_{j_k} \text{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k})$.

In Section 3.3 we provide a unified treatment of LS-Boost(ε) , FS_ε, and $\text{FS}_{\varepsilon_k}$, wherein we show that all these methods can be viewed as special instances of subgradient optimization.

[5] For the purposes of this paper, linear convergence of a sequence $\{a_i\}$ will mean that $a_i \to \bar{a}$ and there exists a scalar $\gamma < 1$ for which $(a_i - \bar{a})/(a_{i-1} - \bar{a}) \leq \gamma$ for all i. Sublinear convergence will mean that there is no such $\gamma < 1$ that satisfies the above property. For much more general versions of linear and sublinear convergence, see [2] for example.

3.2. *Brief Review of Subgradient Descent.* We briefly motivate and review the subgradient descent method for non-differentiable convex optimization problems. Consider the following optimization problem:

$$f^* := \min_x \quad f(x) \quad \text{s.t.} \quad x \in P \ , \tag{3.3}$$

where $P \subseteq \mathbb{R}^n$ is a closed convex set and $f(\cdot) : P \to \mathbb{R}$ is a convex function. If $f(\cdot)$ is differentiable, then $f(\cdot)$ will satisfy the following gradient inequality:

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \text{for any } x, y \in P \ ,$$

which states that $f(\cdot)$ lies above its first-order (linear) approximation at x. One of the most intuitive optimization schemes for solving (3.3) is the method of gradient descent. This method is initiated at a given point $x^0 \in P$. If x^k is the current iterate, then the next iterate is given by the update formula: $x^{k+1} \leftarrow \Pi_P(x^k - \alpha_k \nabla f(x^k))$. In this method the potential new point is $x^k - \alpha_k \nabla f(x^k)$, where $\alpha_k > 0$ is called the step-size at iteration k, and the step is taken in the direction of the negative of the gradient. If this potential new point lies outside of the feasible region P, it is then projected back onto P. Here recall that $\Pi_P(\cdot)$ is the Euclidean projection operator, namely $\Pi_P(x) := \arg\min_{y \in P} \|x - y\|_2$.

Now suppose that $f(\cdot)$ is not differentiable. By virtue of the fact that $f(\cdot)$ is convex, then $f(\cdot)$ will have a *subgradient* at each point x. Recall that g is a subgradient of $f(\cdot)$ at x if the following subgradient inequality holds:

$$f(y) \geq f(x) + g^T (y - x) \quad \text{for all } y \in P \ , \tag{3.4}$$

which generalizes the gradient inequality above and states that $f(\cdot)$ lies above the linear function on the right side of (3.4). Because there may exist more than one subgradient of $f(\cdot)$ at x, let $\partial f(x)$ denote the set of subgradients of $f(\cdot)$ at x. Then "$g \in \partial f(x)$" denotes that g is a subgradient of $f(\cdot)$ at the point x, and so g satisfies (3.4) for all y. The subgradient descent method (see [47], for example) is a simple generalization of the method of gradient descent to the case when $f(\cdot)$ is not differentiable. One simply replaces the gradient by the subgradient, yielding the following update scheme:

$$\begin{array}{lll} \text{Compute a subgradient of } f(\cdot) \text{ at } x^k & : & g^k \in \partial f(x^k) \\ \text{Peform update at } x^k & : & x^{k+1} \leftarrow \Pi_P(x^k - \alpha_k g^k) \ . \end{array} \tag{3.5}$$

The following proposition summarizes a well-known computational guarantee associated with the subgradient descent method.

PROPOSITION 3.1. **(Convergence Bound for Subgradient Descent[41, 39])** *Consider the subgradient descent method (3.5), using a constant step-size $\alpha_i = \alpha$ for all i. Let x^* be an optimal solution of (3.3) and suppose that the subgradients are uniformly bounded, namely $\|g^i\|_2 \leq G$ for all $i \geq 0$. Then for each $k \geq 0$, the following inequality holds:*

$$\min_{i \in \{0,\ldots,k\}} f(x^i) \quad \leq \quad f^* \ + \ \frac{\|x^0 - x^*\|_2^2}{2(k+1)\alpha} + \frac{\alpha G^2}{2} \ . \qquad \square \tag{3.6}$$

The left side of (3.6) is simply the best objective function value obtained among the first k iterations. The right side of (3.6) bounds the best objective function value from above, namely the optimal value f^* plus a nonnegative quantity that is a function of the number of iterations k, the constant step-size α, the bound G on the norms of subgradients, and the distance from the initial point to an optimal solution x^* of (3.3). Note that for a fixed step-size $\alpha > 0$, the right side of (3.6) goes to $\frac{\alpha G^2}{2}$ as $k \to \infty$. In the interest of completeness, we include a proof of Proposition 3.1 in Supplementary Material Section A.2.5, [19].

3.3. *A Subgradient Descent Framework for Boosting.* We now show that the boosting algorithms discussed in Section 1, namely FS_ε and its relatives $\text{FS}_{\varepsilon_k}$ and LS-BOOST(ε), can all be interpreted as instantiations of the subgradient descent method to minimize the largest absolute correlation between the residuals and predictors.

Let $P_{\text{res}} := \{r \in \mathbb{R}^n : r = \mathbf{y} - \mathbf{X}\beta \text{ for some } \beta \in \mathbb{R}^p\}$ denote the affine space of residuals and consider the following convex optimization problem:

$$\text{Correlation Minimization (CM)}: \ f^* := \begin{array}{ll} \min\limits_{r} & f(r) := \quad \|\mathbf{X}^T r\|_\infty \\ \text{s.t.} & r \in P_{\text{res}}, \end{array} \tag{3.7}$$

which we dub the "Correlation Minimization" problem, or CM for short. Note an important subtlety in the CM problem, namely that the optimization variable in CM is the *residual* r and *not* the regression coefficient vector β.

Since the columns of $\mathbf{X}$ have unit ℓ_2 norm by assumption, $f(r)$ is the largest absolute correlation between the residual vector r and the predictors.

Therefore (3.7) is the convex optimization problem of minimizing the largest correlation between the residuals and the predictors, over all possible values of the residuals. From (2.3) with $r = \mathbf{y} - \mathbf{X}\beta$ we observe that $\mathbf{X}^T r = 0$ if and only if β is a least squares solution, whereby $f(r) = \|\mathbf{X}^T r\|_\infty = 0$ for the least squares residual vector $r = \hat{r}_{\text{LS}} = \mathbf{y} - \mathbf{X}\hat{\beta}_{\text{LS}}$. Since the objective function in (3.7) is nonnegative, we conclude that $f^* = 0$ and the least squares residual vector $\hat{r}_{\text{LS}}$ is also the unique optimal solution of the CM problem (3.7). Thus CM can be viewed as an optimization problem which also produces the least squares solution.

The following proposition states that the three boosting algorithms FS_ε, $\text{FS}_{\varepsilon_k}$ and LS-Boost(ε) can all be viewed as instantiations of the subgradient descent method to solve the CM problem (3.7).

Proposition 3.2. *Consider the subgradient descent method* (3.5) *with step-size sequence* $\{\alpha_k\}$ *to solve the correlation minimization (CM) problem* (3.7)*, initialized at* $\hat{r}^0 = \mathbf{y}$*. Then:*

(i) the FS_ε *algorithm is an instance of subgradient descent, with a constant step-size* $\alpha_k := \varepsilon$ *at each iteration,*
(ii) the FS_{ε_k} *algorithm is an instance of subgradient descent, with non-uniform step-sizes* $\alpha_k := \varepsilon_k$ *at iteration* k*, and*
(iii) the LS-Boost(ε) *algorithm is an instance of subgradient descent, with non-uniform step-sizes* $\alpha_k := \varepsilon|\tilde{u}_{j_k}|$ *at iteration* k*, where* $\tilde{u}_{j_k} := \arg\min_u \|\hat{r}^k - \mathbf{X}_{j_k} u\|_2^2$.

Proof. We first prove *(i)*. Recall the update of the residuals in FS_ε:

$$\hat{r}^{k+1} = \hat{r}^k - \varepsilon \cdot \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k} \ .$$

We first show that $g^k := \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k}$ is a subgradient of the objective function $f(r) = \|\mathbf{X}^T r\|_\infty$ of the correlation minimization problem CM (3.7) at $r = \hat{r}^k$. At iteration k, FS_ε chooses the coefficient to update by selecting $j_k \in \arg\max_{j \in \{1,\ldots,p\}} |(\hat{r}^k)^T \mathbf{X}_j|$, whereby $\operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \left((\hat{r}^k)^T \mathbf{X}_{j_k}\right) = \|\mathbf{X}^T(\hat{r}^k)\|_\infty$, and therefore for any r it holds that:

$$\begin{aligned} f(r) = \|\mathbf{X}^T r\|_\infty &\geq \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \left((\mathbf{X}_{j_k})^T r\right) \\ &= \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \left((\mathbf{X}_{j_k})^T (\hat{r}^k + r - \hat{r}^k)\right) \\ &= \|\mathbf{X}^T(\hat{r}^k)\|_\infty + \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \left((\mathbf{X}_{j_k})^T (r - \hat{r}^k)\right) \\ &= f(\hat{r}^k) + \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \left((\mathbf{X}_{j_k})^T (r - \hat{r}^k)\right) \ . \end{aligned}$$

Therefore using the definition of a subgradient in (3.4), it follows that $g^k := \mathrm{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k})\mathbf{X}_{j_k}$ is a subgradient of $f(r) = \|\mathbf{X}^T r\|_\infty$ at $r = \hat{r}^k$. Therefore the update $\hat{r}^{k+1} = \hat{r}^k - \varepsilon \cdot \mathrm{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k})\mathbf{X}_{j_k}$ is of the form $\hat{r}^{k+1} = \hat{r}^k - \varepsilon g^k$ where $g^k \in \partial f(\hat{r}^k)$. Last of all notice that the update can also be written as $\hat{r}^k - \varepsilon g^k = \hat{r}^{k+1} = \mathbf{y} - \mathbf{X}\hat{\beta}^{k+1} \in P_{\mathrm{res}}$, hence $\Pi_{P_{\mathrm{res}}}(\hat{r}^k - \varepsilon g^k) = \hat{r}^k - \varepsilon g^k$, i.e., the projection step is superfluous here, and therefore $\hat{r}^{k+1} = \Pi_{P_{\mathrm{res}}}(\hat{r}^k - \varepsilon g^k)$, which is precisely the update for the subgradient descent method with step-size $\alpha_k := \varepsilon$.

The proof of *(ii)* is the same as *(i)* with a step-size choice of $\alpha_k = \varepsilon_k$ at iteration k. Furthermore, as discussed in Section 1, LS-Boost(ε) may be thought of as a specific instance of $\mathrm{FS}_{\varepsilon_k}$, whereby the proof of *(iii)* follows as a special case of *(ii)*. □

Proposition 3.2 presents a new interpretation of the boosting algorithms FS_ε and its cousins as subgradient descent. This is interesting especially since FS_ε and LS-Boost(ε) have been traditionally interpreted as greedy coordinate descent or steepest descent type procedures [33, 21]. This has the following consequences of note:

• We take recourse to existing tools and results about subgradient descent optimization to inform us about the computational guarantees of these methods. When translated to the setting of linear regression, these results will shed light on the data fidelity *vis-à-vis* shrinkage characteristics of FS_ε and its cousins — all using quantities that can be easily obtained prior to running the boosting algorithm. We will show the details of this in Theorem 3.1 below.

• The subgradient optimization viewpoint provides a unifying algorithmic theme which we will also apply to a regularized version of problem CM (3.7), and that we will show is very strongly connected to the Lasso. This will be developed in Section 4. Indeed, the regularized version of the CM problem that we will develop in Section 4 will lead to a new family of boosting algorithms which are a seemingly minor variant of the basic FS_ε algorithm but deliver ($O(\varepsilon)$-approximate) solutions of the Lasso.

3.4. *Deriving and Interpreting Computational Guarantees for FS_ε.* The following theorem presents the convergence properties of FS_ε, which are a consequence of the interpretation of FS_ε as an instance of the subgradient descent method.

Theorem 3.1. **(Convergence Properties of $\mathbf{FS}_\varepsilon$)** *Consider the FS_ε algorithm with learning rate ε. Let $k \geq 0$ be the total number of iterations.*

Then there exists an index $i \in \{0, \ldots, k\}$ for which the following bounds hold:

(i) (training error): $L_n(\hat{\beta}^i) - L_n^* \leq \frac{p}{2n\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}\left[\frac{\|\mathbf{X}\hat{\beta}_{LS}\|_2^2}{\varepsilon(k+1)} + \varepsilon\right]^2$

(ii) (regression coefficients): there exists a least squares solution $\hat{\beta}_{LS}^i$ such that:

$$\|\hat{\beta}^i - \hat{\beta}_{LS}^i\|_2 \leq \frac{\sqrt{p}}{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}\left[\frac{\|\mathbf{X}\hat{\beta}_{LS}\|_2^2}{\varepsilon(k+1)} + \varepsilon\right]$$

(iii) (predictions): for every least-squares solution $\hat{\beta}_{LS}$ it holds that

$$\|\mathbf{X}\hat{\beta}^i - \mathbf{X}\hat{\beta}_{LS}\|_2 \leq \frac{\sqrt{p}}{\sqrt{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}}\left[\frac{\|\mathbf{X}\hat{\beta}_{LS}\|_2^2}{\varepsilon(k+1)} + \varepsilon\right]$$

(iv) (correlation values) $\|\mathbf{X}^T\hat{r}^i\|_\infty \leq \dfrac{\|\mathbf{X}\hat{\beta}_{LS}\|_2^2}{2\varepsilon(k+1)} + \dfrac{\varepsilon}{2}$

(v) (ℓ_1-shrinkage of coefficients): $\|\hat{\beta}^i\|_1 \leq k\varepsilon$

(vi) (sparsity of coefficients): $\|\hat{\beta}^i\|_0 \leq k$. □

The proof of Theorem 3.1 is presented in Section A.2.6, in [19].

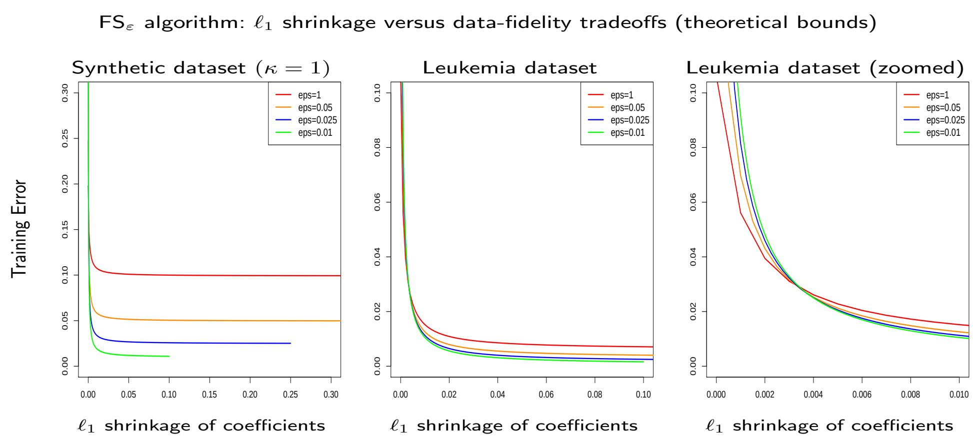

FIG 6. *Figure showing profiles of ℓ_1 shrinkage bounds of the regression coefficients versus training error bounds for the FS$_\varepsilon$ algorithm, for different values of the learning rate ε. The profiles have been obtained from the bounds in parts* (i) *and* (v) *of Theorem 3.1. The left panel corresponds to a hypothetical dataset using $\kappa = \frac{p}{\lambda_{\text{pmin}}} = 1$, and the middle and right panels use the parameters of the Leukemia dataset.*

Interpreting the Computational Guarantees. Theorem 3.1 accomplishes for FS_ε what Theorem 2.1 did for LS-BOOST(ε) — parts *(i)* – *(iv)* of the theorem describe the rate in which the training error, regression coefficients, and related quantities make their way towards their ($O(\varepsilon)$-approximate) unregularized least squares counterparts. Part *(v)* of the theorem also describes the rate at which the shrinkage of the regression coefficients evolve as a function of the number of boosting iterations. The rate of convergence of FS_ε is sublinear, unlike the linear rate of convergence for LS-BOOST(ε) . Note that this type of sublinear convergence implies that the rate of decrease of the training error (for instance) is dramatically faster in the very early iterations as compared to later iterations. Taken together, Theorems 3.1 and 2.1 highlight an important difference between the behavior of algorithms LS-BOOST(ε) and FS_ε:

- the limiting solution of the LS-BOOST(ε) algorithm (as $k \to \infty$) corresponds to the unregularized least squares solution, but
- the limiting solution of the FS_ε algorithm (as $k \to \infty$) corresponds to an $O(\varepsilon)$ approximate least squares solution.

As demonstrated in Theorems 2.1 and 3.1, both LS-BOOST(ε) and FS_ε have nice convergence properties with respect to the unconstrained least squares problem (2.1). However, unlike the convergence results for LS-BOOST(ε) in Theorem 2.1, FS_ε exhibits a *sublinear* rate of convergence towards a *suboptimal* least squares solution. For example, part *(i)* of Theorem 3.1 implies in the limit as $k \to \infty$ that FS_ε identifies a model with training error at most:

$$L_n^* + \frac{p\varepsilon^2}{2n(\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X}))} \ . \tag{3.8}$$

In addition, part *(ii)* of Theorem 3.1 implies that as $k \to \infty$, FS_ε identifies a model whose distance to the set of least squares solutions $\{\hat{\beta}_{\text{LS}} : \mathbf{X}^T\mathbf{X}\hat{\beta}_{\text{LS}} = \mathbf{X}^T\mathbf{y}\}$ is at most: $\frac{\varepsilon\sqrt{p}}{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}$.

Note that the computational guarantees in Theorem 3.1 involve the quantities $\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})$ and $\|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2$, assuming n and p are fixed. To settle ideas, let us consider the synthetic datasets used in Figures 4 and 1, where the covariates were generated from a multivariate Gaussian distribution with pairwise correlation ρ. Figure 4 suggests that $\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})$ decreases with increasing ρ values. Thus, controlling for other factors appearing in the computational bounds[6] , it follows from the statements of Theorem 3.1 that the

[6]To control for other factors, for example, we may assume that $p > n$ and for different values of ρ we have $\|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2 = \|\mathbf{y}\|_2 = 1$ with ε fixed across the different examples.

training error decreases much more rapidly for smaller ρ values, as a function of k. This is nicely validated by the computational results in Figure 1 (the three top panel figures), which show that the training errors decay at a faster rate for smaller values of ρ.

Let us examine more carefully the properties of the sequence of models explored by FS_ε and the corresponding tradeoffs between data fidelity and model complexity. Let TBOUND and SBOUND denote the training error bound and shrinkage bound in parts *(i)* and *(v)* of Theorem 3.1, respectively. Then simple manipulation of the arithmetic in these two bounds yields the following tradeoff equation:

$$\text{TBOUND} = \frac{p}{2n\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}\left[\frac{\|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2^2}{\text{SBOUND}+\varepsilon}+\varepsilon\right]^2 .$$

The above tradeoff between the training error bound and the shrinkage bound is illustrated in Figure 6, which shows this tradeoff curve for four different values of the learning rate ε. Except for very small shrinkage levels, lower values of ε produce smaller training errors. But unlike the corresponding tradeoff curves for LS-BOOST(ε), there is a range of values of the shrinkage for which smaller values of ε actually produce larger training errors, though admittedly this range is for very small shrinkage values. For more reasonable shrinkage values, smaller values of ε will correspond to smaller values of the training error.

Part *(v)* of Theorems 2.1 and 3.1 presents shrinkage bounds for FS_ε and LS-BOOST(ε), respectively. Let us briefly compare these bounds. Examining the shrinkage bound for LS-BOOST(ε), we can bound the left term from above by $\sqrt{k}\sqrt{\varepsilon}\|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2$. We can also bound the right term from above by $\varepsilon\|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2/(1-\sqrt{\gamma})$ where recall from Section 2 that γ is the linear convergence rate coefficient $\gamma := 1-\frac{\varepsilon(2-\varepsilon)\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}{4p}$. We may therefore alternatively write the following shrinkage bound for LS-BOOST(ε):

$$\|\hat{\beta}^k\|_1 \le \|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2 \min\left\{\sqrt{k}\sqrt{\varepsilon}\ ,\ \varepsilon/(1-\sqrt{\gamma})\right\} . \tag{3.9}$$

The shrinkage bound for FS_ε is simply $k\varepsilon$. Comparing these two bounds, we observe that not only does the shrinkage bound for FS_ε grow at a faster rate as a function of k for large enough k, but also the shrinkage bound for FS_ε grows unbounded in k, unlike the right term above for the shrinkage bound of LS-BOOST(ε).

One can also compare FS_ε and LS-Boost(ε) in terms of the efficiency with which these two methods achieve a certain pre-specified data-fidelity. In Section A.2.7 [19] we show, at least in theory, that LS-Boost(ε) is much more efficient than FS_ε at achieving such data-fidelity, and furthermore it does so with much better shrinkage.

Generalizations of FS_ε. Here we briefly mention some recent work that generalize the FS_ε algorithm: [24] study extensions to incorporate non-convex penalization schemes, and [49] propose a framework generalizing FS_ε to a flexible family of convex loss functions and sparsity inducing convex regularizers.

4. Regularized Correlation Minimization, Boosting, and Lasso.

Roadmap. In this section, we introduce a new boosting algorithm, parameterized by a scalar $\delta \geq 0$, which we denote by R-$\text{FS}_{\varepsilon,\delta}$ (for Regularized incremental Forward Stagewise regression), that is obtained by incorporating a simple rescaling step to the coefficient updates in FS_ε. We then introduce a regularized version of the Correlation Minimization (CM) problem (3.7) which we refer to as RCM. We show that the adaptation of the subgradient descent algorithmic framework to the Regularized Correlation Minimization problem RCM exactly yields the algorithm R-$\text{FS}_{\varepsilon,\delta}$. The new algorithm R-$\text{FS}_{\varepsilon,\delta}$ may be interpreted as a natural extension of popular boosting algorithms like FS_ε, and has the following notable properties:

• Whereas FS_ε updates the coefficients in an additive fashion by adding a small amount ε to the coefficient most correlated with the current residuals, R-$\text{FS}_{\varepsilon,\delta}$ first shrinks *all* of the coefficients by a scaling factor $1 - \frac{\varepsilon}{\delta} < 1$ and then updates the selected coefficient in the same additive fashion as FS_ε.

• R-$\text{FS}_{\varepsilon,\delta}$ delivers $O(\varepsilon)$-accurate solutions to the Lasso in the limit as $k \to \infty$, unlike FS_ε which delivers $O(\varepsilon)$-accurate solutions to the unregularized least squares problem.

•R-$\text{FS}_{\varepsilon,\delta}$ has computational guarantees similar in spirit to the ones described in the context of FS_ε – these quantities directly inform us about the data-fidelity *vis-à-vis* shrinkage tradeoffs as a function of the number of boosting iterations and the learning rate ε.

The notion of using additional regularization along with the implicit shrinkage imparted by boosting is not new in the literature. Various interesting notions have been proposed in [26, 9, 53, 13, 23], see also the discussion in Section A.3.4 [19]. However, the framework we present here is new. We present a unified subgradient descent framework for a class of regularized

CM problems that results in algorithms that have appealing structural similarities with forward stagewise regression type algorithms, while also being very strongly connected to the LASSO.

Boosting with additional shrinkage – R-FS$_{\varepsilon,\delta}$. Here we give a formal description of the R-FS$_{\varepsilon,\delta}$ algorithm. R-FS$_{\varepsilon,\delta}$ is controlled by two parameters: the learning rate ε, which plays the same role as the learning rate in FS$_\varepsilon$, and the "regularization parameter" $\delta \geq \varepsilon$. Our reason for referring to δ as a regularization parameter is due to the connection between R-FS$_{\varepsilon,\delta}$ and the LASSO, which will be made clear later. The shrinkage factor, i.e., the amount by which we shrink the coefficients before updating the selected coefficient, is determined as $1 - \frac{\varepsilon}{\delta}$. Supposing that we choose to update the coefficient indexed by j_k at iteration k, then the coefficient update may be written as:

$$\hat{\beta}^{k+1} \leftarrow \left(1 - \tfrac{\varepsilon}{\delta}\right) \hat{\beta}^k + \varepsilon \cdot \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) e_{j_k} \ .$$

Below we give a concise description of R-FS$_{\varepsilon,\delta}$, including the update for the residuals that corresponds to the update for the coefficients stated above.

Algorithm: R-FS$_{\varepsilon,\delta}$

Fix a learning rate $\varepsilon > 0$, regularization parameter $\delta > 0$ (with $\varepsilon \leq \delta$), number of iterations M; and initialize at $\hat{\beta}^0 = 0$.

For $0 \leq k \leq M$ select $j_k \in \arg\max_{j \in \{1,\ldots,p\}} |(\hat{r}^k)^T \mathbf{X}_j|$ and perform the update:

(4.1)

$$\begin{aligned} \hat{r}^{k+1} &\leftarrow \hat{r}^k - \varepsilon \left[\operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k} + \tfrac{1}{\delta}(\hat{r}^k - \mathbf{y}) \right] \\ \hat{\beta}^{k+1}_{j_k} &\leftarrow \left(1 - \tfrac{\varepsilon}{\delta}\right) \hat{\beta}^k_{j_k} + \varepsilon \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \ \text{ and } \ \hat{\beta}^{k+1}_j \leftarrow \left(1 - \tfrac{\varepsilon}{\delta}\right) \hat{\beta}^k_j \ , j \neq j_k . \end{aligned}$$

Note that R-FS$_{\varepsilon,\delta}$ and FS$_\varepsilon$ are structurally very similar – and indeed when $\delta = \infty$ then R-FS$_{\varepsilon,\delta}$ is exactly FS$_\varepsilon$. Note also that R-FS$_{\varepsilon,\delta}$ shares the same upper bound on the sparsity of the regression coefficients as FS$_\varepsilon$, namely for all k it holds that: $\|\hat{\beta}^k\|_0 \leq k$. When $\delta < \infty$ then, as previously mentioned, the main structural difference between R-FS$_{\varepsilon,\delta}$ and FS$_\varepsilon$ is the additional rescaling of the coefficients by the factor $1 - \frac{\varepsilon}{\delta}$. This rescaling better controls the growth of the coefficients and, as will be demonstrated next, plays a key role in connecting R-FS$_{\varepsilon,\delta}$ to the LASSO.

Regularized Correlation Minimization (RCM) and LASSO. The starting point of our formal analysis of R-FS$_{\varepsilon,\delta}$ is the Correlation Minimization (CM) problem (3.7), which we now modify by introducing a regularization term that penalizes residuals that are far from the vector of observations $\mathbf{y}$. This modification leads to the following parametric family of optimization problems indexed by $\delta \in (0, \infty]$:

(4.2)
$$\begin{array}{rlcll} \mathrm{RCM}_\delta: & f^*_\delta := \min\limits_{r} & f_\delta(r) & := & \|\mathbf{X}^T r\|_\infty + \frac{1}{2\delta}\|r - \mathbf{y}\|_2^2 \\ & \text{s.t.} & r \in P_{\text{res}} & := & \{r \in \mathbb{R}^n : r = \mathbf{y} - \mathbf{X}\beta \text{ for some } \beta \in \mathbb{R}^p\}\ , \end{array}$$

where "RCM" connotes Regularlized Correlation Minimization. Note that RCM reduces to the correlation minimization problem CM (3.7) when $\delta = \infty$. RCM may be interpreted as the problem of minimizing, over the space of residuals, the largest correlation between the residuals and the predictors plus a regularization term that penalizes residuals that are far from the response $\mathbf{y}$ (which itself can be interpreted as the residuals associated with the model $\beta = 0$).

Interestingly, as we show in [19] Section A.3.1, RCM (4.2) is equivalent to the LASSO (1.1) via duality. This equivalence provides further insight about the regularization used to obtain RCM_δ. Comparing the LASSO and RCM, notice that the space of the variables of the LASSO is the space of regression coefficients β, namely $\mathbb{R}^p$, whereas the space of the variables of RCM is the space of model residuals, namely $P_{\text{res}} \subset \mathbb{R}^n$. The duality relationship shows that RCM_δ (4.2) is an equivalent characterization of the LASSO problem, just like the correlation minimization (CM) problem (3.7) is an equivalent characterization of the (unregularized) least squares problem. Recall that Proposition 3.2 showed that subgradient descent applied to the CM problem (4.2) (which is RCM_δ with $\delta = \infty$) leads to the well-known boosting algorithm FS_ε. We now extend this theme with the following Proposition, which states that R-$\mathrm{FS}_{\varepsilon,\delta}$ is equivalent to subgradient descent applied to RCM_δ.

PROPOSITION 4.1. *The R-$FS_{\varepsilon,\delta}$ algorithm is an instance of subgradient descent to solve the regularized correlation minimization (RCM_δ) problem (4.2), initialized at $\hat{r}^0 = \mathbf{y}$, with a constant step-size $\alpha_k := \varepsilon$ at each iteration.*

The proof of Proposition 4.1 is presented in [19] Section A.3.2.

4.1. *R-$FS_{\varepsilon,\delta}$: Computational Guarantees and their Implications.* In this subsection we present computational guarantees and convergence properties of the boosting algorithm R-$\mathrm{FS}_{\varepsilon,\delta}$. Due to the structural equivalence between R-$\mathrm{FS}_{\varepsilon,\delta}$ and subgradient descent applied to the RCM_δ problem (4.2) (Proposition 4.1) and the close connection between RCM_δ and the LASSO (see [19], Section A.3.1), the convergence properties of R-FS$_{\varepsilon,\delta}$ are naturally stated with respect to the LASSO problem (1.1). Similar to Theorem 3.1 which described such properties for FS$_\varepsilon$ (with respect to the unregularized least squares problem), we have the following properties for R-FS$_{\varepsilon,\delta}$.

THEOREM 4.1. **(Convergence Properties of R-FS$_{\varepsilon,\delta}$ for the Lasso)** *Consider the R-FS$_{\varepsilon,\delta}$ algorithm with learning rate ε and regularization parameter $\delta \in (0,\infty)$, where $\varepsilon \leq \delta$. Then the regression coefficient $\hat{\beta}^k$ is feasible for the* LASSO *problem (1.1) for all $k \geq 0$. Let $k \geq 0$ denote a specific iteration counter. Then there exists an index $i \in \{0,\ldots,k\}$ for which the following bounds hold:*

(i) (training error): $L_n(\hat{\beta}^i) - L^*_{n,\delta} \ \leq \frac{\delta}{n}\left[\frac{\|\mathbf{X}\hat{\beta}_{LS}\|_2^2}{2\varepsilon(k+1)} + 2\varepsilon\right]$

(ii) (predictions): for every LASSO *solution $\hat{\beta}^*_\delta$ it holds that*

$$\|\mathbf{X}\hat{\beta}^i - \mathbf{X}\hat{\beta}^*_\delta\|_2 \ \leq \sqrt{\frac{\delta\|\mathbf{X}\hat{\beta}_{LS}\|_2^2}{\varepsilon(k+1)} + 4\delta\varepsilon}$$

(iii) (ℓ_1-shrinkage of coefficients): $\|\hat{\beta}^i\|_1 \leq \ \delta\left[1 - \left(1 - \frac{\varepsilon}{\delta}\right)^k\right] \leq \ \delta$

(iv) (sparsity of coefficients): $\|\hat{\beta}^i\|_0 \leq k$. □

The proof of Theorem 4.1 is presented in Supplementary Material [19] Section A.3.3.

Interpreting the Computational Guarantees. The statistical interpretations implied by the computational guarantees presented in Theorem 4.1 are analogous to those previously discussed for LS-BOOST(ε) (Theorem 2.1) and FS$_\varepsilon$ (Theorem 3.1). These guarantees inform us about the data-fidelity *vis-à-vis* shrinkage tradeoffs as a function of the number of boosting iterations, as nicely demonstrated in Figure 7. There is, however, an important differentiation between the properties of R-FS$_{\varepsilon,\delta}$ and the properties of LS-BOOST(ε) and FS$_\varepsilon$, namely:

• For LS-BOOST(ε) and FS$_\varepsilon$, the computational guarantees (Theorems 2.1 and 3.1) describe how the estimates make their way to a unregularized ($O(\varepsilon)$-approximate) least squares solution as a function of the number of boosting iterations.

• For R-FS$_{\varepsilon,\delta}$, our results (Theorem 4.1) characterize how the estimates approach a ($O(\varepsilon)$-approximate) LASSO solution.

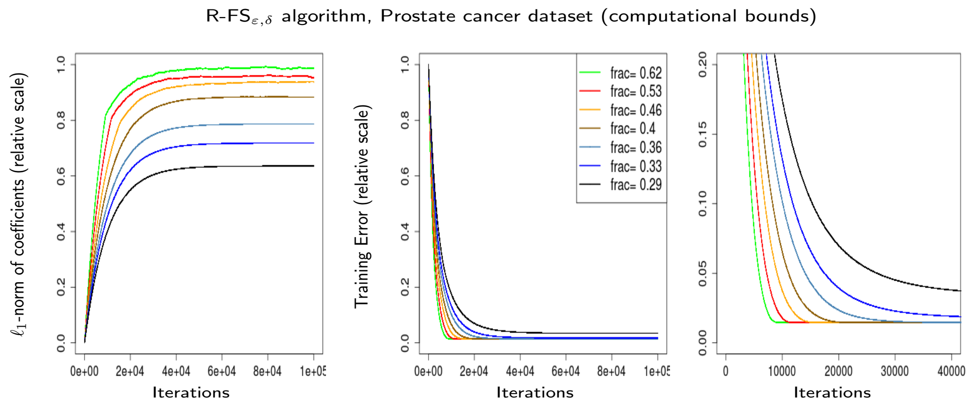

FIG 7. *Figure showing the evolution of the R-FS$_{\varepsilon,\delta}$ algorithm (with $\varepsilon = 10^{-4}$) for different values of δ, as a function of the number of boosting iterations for the Prostate cancer dataset, with $n = 10, p = 44$ (see also Figure 8). [Left panel] shows the change of the ℓ_1-norm of the regression coefficients. [Middle panel] shows the evolution of the training errors, and [Right panel] is a zoomed-in version of the middle panel. Here we took different values of δ given by $\delta = frac \times \delta_{\max}$, where, $\delta_{\max}$ denotes the ℓ_1-norm of the minimum ℓ_1-norm least squares solution, for 7 different values of frac.*

Notice that like FS$_\varepsilon$, R-FS$_{\varepsilon,\delta}$ traces out a profile of regression coefficients. This is reflected in item *(iii)* of Theorem 4.1 which bounds the ℓ_1-shrinkage of the coefficients as a function of the number of boosting iterations k. Due to the rescaling of the coefficients, the ℓ_1-shrinkage may be bounded by a geometric series that approaches δ as k grows. Thus, there are two important aspects of the bound in item *(iii)*: *(a)* the dependence on the number of boosting iterations k which characterizes model complexity during early iterations, and *(b)* the uniform bound of δ which applies even in the limit as $k \to \infty$ and implies that all regression coefficient iterates $\hat{\beta}^k$ are feasible for the LASSO problem (1.1).

On the other hand, item *(i)* characterizes the quality of the coefficients with respect to the LASSO solution, as opposed to the unregularized least squares problem as in FS$_\varepsilon$. In the limit as $k \to \infty$, item *(i)* implies that R-FS$_{\varepsilon,\delta}$ identifies a model with training error at most $L^*_{n,\delta} + \frac{2\delta\varepsilon}{n}$. This upper bound on the training error may be set to any prescribed error level by appropriately tuning ε; in particular, for $\varepsilon \approx 0$ and fixed $\delta > 0$ this limit is essentially $L^*_{n,\delta}$. Thus, combined with the uniform bound of δ on the ℓ_1-shrinkage, we see that the R-FS$_{\varepsilon,\delta}$ algorithm delivers the LASSO solution in the limit as $k \to \infty$.

It is important to emphasize that R-FS$_{\varepsilon,\delta}$ should not just be interpreted as an algorithm to solve the LASSO. Indeed, like FS$_\varepsilon$, the trajectory of the algorithm is important and R-FS$_{\varepsilon,\delta}$ may identify a more statistically interesting model in the interior of its profile. Thus, even if the LASSO solution for δ leads to overfitting, the R-FS$_{\varepsilon,\delta}$ updates may visit a model with better predictive performance by trading off bias and variance in a more desirable fashion suitable for the particular problem at hand.

Figure 8 shows the profiles of R-FS$_{\varepsilon,\delta}$ for different values of $\delta \leq \delta_{\max}$, where $\delta_{\max}$ is the ℓ_1-norm of the minimum ℓ_1-norm least squares solution. Curiously enough, Figure 8 shows that in some cases, the profile of R-FS$_{\varepsilon,\delta}$ bears a lot of similarities with that of the LASSO (as presented in Figure 2). However, the profiles are in general different. Indeed, R-FS$_{\varepsilon,\delta}$ imposes a uniform bound of δ on the ℓ_1-shrinkage, and so for values larger than δ we cannot possibly expect R-FS$_{\varepsilon,\delta}$ to approximate the LASSO path. However, even if δ is taken to be sufficiently large (but finite) the profiles may be different. In this connection it is helpful to draw the analogy between the curious similarities between the FS$_\varepsilon$ (i.e., R-FS$_{\varepsilon,\delta}$ with $\delta = \infty$) and LASSO coefficient profiles, even though the profiles are different in general.

Readers familiar with convex optimization methods will notice that R-FS$_{\varepsilon,\delta}$ bears a striking resemblance to another notable optimization algorithm: the Frank-Wolfe method [15, 35, 18]. Indeed, the update for the coefficients in R-FS$_{\varepsilon,\delta}$ is of the form:

$$\hat{\beta}^{k+1} \leftarrow (1-\bar{\alpha})\,\hat{\beta}^k + \bar{\alpha}\tilde{\beta}^k, \quad \text{where} \quad \tilde{\beta}^k := \delta \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) e_{j_k} \tag{4.3}$$

and $\bar{\alpha} := \frac{\varepsilon}{\delta} \in (0,1]$, where $\tilde{\beta}^k \in \underset{\beta:\|\beta\|_1 \leq \delta}{\arg\min} \left\{ \nabla L_n(\hat{\beta}^k)^T \beta \right\}$, which is *exactly* of the form of the Frank-Wolfe method applied to the LASSO, with constant step-sizes $\bar{\alpha} := \frac{\varepsilon}{\delta}$. The fact that R-FS$_{\varepsilon,\delta}$ is equivalent to both the subgradient descent and Frank-Wolfe methods is no coincidence; indeed, this is a special case of a more general primal-dual equivalence between certain subgradient algorithms and the Frank-Wolfe method developed in [1]. It should be noted, however, that despite this additional equivalence, the use of constant step-sizes in Frank-Wolfe is typically not very sensible – yet turns out to be relevant in the boosting context. We refer the reader to Supplementary Material [19] Section A.3.5 for further discussion regarding this connection with the Frank-Wolfe method.

5. A Modified Forward Stagewise Algorithm for Computing the Lasso Path.

In Section 4 we introduced the boosting algorithm R-FS$_{\varepsilon,\delta}$ (which is a very close cousin of FS$_\varepsilon$) that delivers solutions to the LASSO problem (1.1) for a fixed but arbitrary δ, in the limit as $k \to \infty$ with $\varepsilon \approx 0$. Furthermore, our experiments in Section 6 suggest that R-FS$_{\varepsilon,\delta}$ may lead to

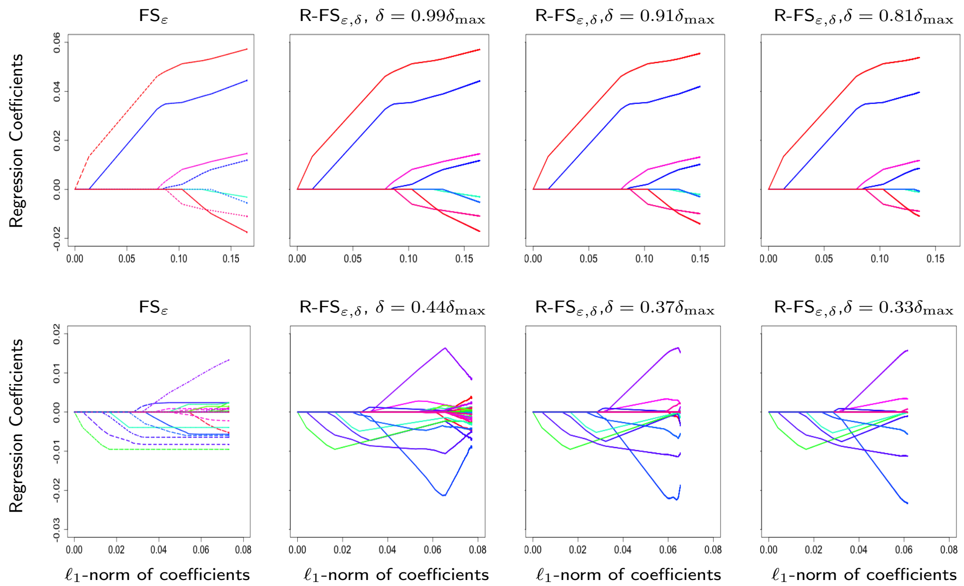

FIG 8. *Coefficient profiles for R-$FS_{\varepsilon,\delta}$ as a function of the ℓ_1-norm of the regression coefficients, for the same datasets appearing in Figure 2. For each example, different values of δ have been considered. The left panel corresponds to the choice $\delta = \infty$, i.e., FS_ε. In all the above cases, the algorithms were run for a maximum of 100,000 boosting iterations with $\varepsilon = 10^{-4}$. [Top Panel] Corresponds to the Prostate cancer dataset with $n = 98$ and $p = 8$. All the coefficient profiles look similar, and they all seem to coincide with the* LASSO *profile (see also Figure 2). [Bottom Panel] Shows the Prostate cancer dataset with a subset of samples $n = 10$ with all interactions included with $p = 44$. The coefficient profiles in this example are sensitive to the choice of δ and are seen to be more constrained towards the end of the path, for decreasing δ values. The profiles are different than the* LASSO *profiles, as seen in Figure 2. The regression coefficients at the end of the path correspond to approximate* LASSO *solutions, for the respective values of δ.*

estimators with good statistical properties for a wide range of values of δ, provided that the value of δ is not too small. While R-$\mathrm{FS}_{\varepsilon,\delta}$ by itself may be considered as a regularization scheme with excellent statistical properties, the boosting profile delivered by R-$\mathrm{FS}_{\varepsilon,\delta}$ might in some cases be different from the LASSO coefficient profile, as we saw in Figure 8. Therefore in this section we investigate the following question: is it possible to modify the R-$\mathrm{FS}_{\varepsilon,\delta}$ algorithm, while still retaining its basic algorithmic characteristics, so that it delivers an approximate LASSO coefficient profile for any dataset? We answer this question in the affirmative herein.

To fix ideas, let us consider producing the (approximate) LASSO path by producing a sequence of (approximate) LASSO solutions on a predefined grid of regularization parameter values δ in the interval $(0, \bar{\delta}]$ given by $0 <$

$\bar{\delta}_0 < \bar{\delta}_1 < \ldots < \bar{\delta}_K = \bar{\delta}$. (A standard method for generating the grid points is to use a geometric sequence such as $\bar{\delta}_i = \eta^{-i} \cdot \bar{\delta}_0$ for $i = 0, \ldots, K$, for some $\eta \in (0,1)$.) Motivated by the notion of warm-starts popularly used in the statistical computing literature in the context of computing a path of LASSO solutions via coordinate descent methods [27], we propose here a slight modification of the R-FS$_{\varepsilon,\delta}$ algorithm that sequentially updates the value of δ according to the predefined grid values $\bar{\delta}_0, \bar{\delta}_1, \cdots, \bar{\delta}_K = \bar{\delta}$, and does so prior to each update of $\hat{r}^i$ and $\hat{\beta}^i$. We call this method PATH-R-FS$_\varepsilon$, whose complete description is as follows:

Algorithm: PATH-R-FS$_\varepsilon$

Fix the learning rate $\varepsilon > 0$, choose values $\bar{\delta}_i$, $i = 0, \ldots, K$, satisfying $0 < \bar{\delta}_0 \leq \bar{\delta}_1 \leq \cdots \leq \bar{\delta}_K \leq \bar{\delta}$ such that $\varepsilon \leq \bar{\delta}_0$. Initialize at $\hat{\beta}^0 = 0$.

For $0 \leq k \leq K$ select coefficient index $j_k \in \arg\max_{j \in \{1,\ldots,p\}} |(\hat{r}^k)^T \mathbf{X}_j|$ and perform the update:

$$\begin{aligned}
&\hat{r}^{k+1} \leftarrow \hat{r}^k - \varepsilon \left[\mathrm{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k} + (\hat{r}^k - \mathbf{y})/\bar{\delta}_k\right] \\
&\hat{\beta}^{k+1}_{j_k} \leftarrow \left(1 - \varepsilon/\bar{\delta}_k\right) \hat{\beta}^k_{j_k} + \varepsilon\ \mathrm{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \text{ and } \hat{\beta}^{k+1}_j \leftarrow \left(1 - \varepsilon/\bar{\delta}_k\right) \hat{\beta}^k_j\ , j \neq j_k
\end{aligned}$$

Notice that PATH-R-FS$_\varepsilon$ retains the identical structure of a forward stagewise regression type of method, and uses the same essential update structure (3.1) of R-FS$_{\varepsilon,\delta}$. Indeed, the updates of $\hat{r}^{k+1}$ and $\hat{\beta}^{k+1}$ in PATH-R-FS$_\varepsilon$ are identical to those in (3.1) of R-FS$_{\varepsilon,\delta}$ except that they use the regularization value $\bar{\delta}_k$ at iteration k instead of the constant value of δ as in R-FS$_{\varepsilon,\delta}$.

Theoretical Guarantees for PATH-R-FS$_\varepsilon$. Analogous to Theorem 4.1 for R-FS$_{\varepsilon,\delta}$, the following theorem describes properties of the PATH-R-FS$_\varepsilon$ algorithm. In particular, the theorem provides rigorous guarantees about the distance between the PATH-R-FS$_\varepsilon$ algorithm and the LASSO coefficient profiles – which apply to any general dataset.

THEOREM 5.1. **(Computational Guarantees of PATH-R-FS$_\varepsilon$)** *Consider the PATH-R-FS$_\varepsilon$ algorithm with the given learning rate ε and regularization parameter sequence $\{\bar{\delta}_k\}$. Let $k \geq 0$ denote the total number of iterations. Then the following holds:*

(i) (LASSO feasibility and average training error): for each $i = 0, \ldots, k$, $\hat{\beta}^i$ provides an approximate solution to the LASSO problem for $\delta = \bar{\delta}_i$. More specifically, $\hat{\beta}^i$ is feasible for the LASSO problem for $\delta = \bar{\delta}_i$, and satisfies the following suboptimality bound with respect to the entire

boosting profile:

$$\frac{1}{k+1}\sum_{i=0}^{k}\left(L_n(\hat{\beta}^i) - L^*_{n,\bar{\delta}_i}\right) \;\leq\; \frac{\bar{\delta}\|\mathbf{X}\hat{\beta}_{LS}\|_2^2}{2n\varepsilon(k+1)} + \frac{2\bar{\delta}\varepsilon}{n}$$

(ii) (ℓ_1-shrinkage of coefficients): $\|\hat{\beta}^i\|_1 \leq \bar{\delta}_i$ *for* $i = 0, \ldots, k$.
(iii) (sparsity of coefficients): $\|\hat{\beta}^i\|_0 \leq i$ *for* $i = 0, \ldots, k$. □

COROLLARY 5.1. **(PATH-R-FS$_\varepsilon$ approximates the Lasso path)** *For every fixed* $\varepsilon > 0$ *and* $k \to \infty$ *it holds that:*

$$\limsup_{k\to\infty}\;\frac{1}{k+1}\sum_{i=0}^{k}\left(L_n(\hat{\beta}^i) - L^*_{n,\bar{\delta}_i}\right) \leq \frac{2\bar{\delta}\varepsilon}{n}\;,$$

(and the quantity on the right side of the above bound goes to zero as $\varepsilon \to 0$*).* □

The proof of Theorem 5.1 is presented in Supplementary Material [19] Section A.4.1.

Interpreting the computational guarantees. Let us now provide some interpretation of the results stated in Theorem 5.1. Recall that Theorem 4.1 presented bounds on the distance between the training errors achieved by the boosting algorithm R-FS$_{\varepsilon,\delta}$ and LASSO training errors for a *fixed* but arbitrary δ that is specified *a priori*. The message in Theorem 5.1 generalizes this notion to a *family* of LASSO solutions corresponding to a *grid* of δ values. The theorem thus quantifies how the boosting algorithm PATH-R-FS$_\varepsilon$ *simultaneously* approximates a path of LASSO solutions.

Part *(i)* of Theorem 5.1 first implies that the sequence of regression coefficient vectors $\{\hat{\beta}^i\}$ is feasible along the LASSO path, for the LASSO problem (1.1) for the sequence of regularization parameter values $\{\bar{\delta}_i\}$. In considering guarantees with respect to the training error, we would ideally like guarantees that hold across the entire spectrum of $\{\bar{\delta}_i\}$ values. While part *(i)* does not provide such strong guarantees, part *(i)* states that these quantities will be sufficiently small *on average*. Indeed, for a fixed ε and as $k \to \infty$, part *(i)* states that the average of the differences between the training errors produced by the algorithm and the optimal training errors is at most $\frac{2\bar{\delta}\varepsilon}{n}$. This non-vanishing bound (for $\varepsilon > 0$) is a consequence of the fixed learning rate ε used in PATH-R-FS$_\varepsilon$ – such bounds were also observed for R-FS$_{\varepsilon,\delta}$ and FS$_\varepsilon$.

On average, the training error of the PATH-R-FS$_\varepsilon$ solutions will be sufficiently close (as controlled by the learning rate ε) to the LASSO training error for the corresponding regularization parameter grid values $\{\bar{\delta}_i\}$. And while PATH-R-FS$_\varepsilon$ provides the most amount of flexibility in terms of controlling for model complexity since it allows for *any* (monotone) sequence of regularization parameter values in the range $(0, \bar{\delta}]$, this freedom comes at the cost of weaker training error guarantees with respect to any particular $\bar{\delta}_i$ value (as opposed to R-FS$_{\varepsilon,\delta}$ which provides strong guarantees with respect to the fixed value δ). Nevertheless, part *(i)* of Theorem 5.1 guarantees that the training errors will be sufficiently small on average across the entire path of regularization parameter values explored by the algorithm.

It is interesting that PATH-R-FS$_\varepsilon$ approximates the LASSO path training errors, with associated shrinkage and sparsity bounds – all the while performing only boosting steps. In a sense, the price it pays for being a boosting method is that the approximation to the LASSO path is only on average over the chosen grid points (as opposed to holding simultaneously over all grid points). In contrast, classic algorithms that exactly track the piecewise-linear LASSO path require an exponential number of iterations in the worst-case, see [28] for a very general result in this regard. [49] proposes "shrunken stagewise" which is shown to deliver LASSO solutions in a limiting sense as certain parameters go to zero and under technical assumptions. On the other hand, there are several efficient LASSO path algorithms with computational guarantees that hold approximately over the *entire* LASSO path; such methods typically choose the $\{\bar{\delta}_i\}$ values adaptively – see [29] and [30] for some general results in this regard, and in particular [31] for optimal general complexity results in this context. Similar adaptive choices of $\{\bar{\delta}_i\}$ values are also studied by [49] for computing an approximate LASSO path.

6. Some Computational Experiments.

We consider an array of examples exploring statistical properties of the different boosting algorithms studied herein. We consider different types of synthetic and real datasets, which are briefly described here.

Synthetic datasets. We considered synthetically generated datasets of the following types:

• **Eg-A.** Here the data matrix $\mathbf{X}$ is generated from a multivariate normal distribution, i.e., for each $i = 1, \ldots, n$, $\mathbf{x}_i \sim \mathrm{MVN}(0, \Sigma)$. Here $\mathbf{x}_i$ denotes the i^{th} row of $\mathbf{X}$ and $\Sigma = (\sigma_{ij}) \in \mathbb{R}^{p \times p}$ has all off-diagonal entries equal to ρ and all diagonal entries equal to one. The response $\mathbf{y} \in \mathbb{R}^n$ is generated as

$\mathbf{y} = \mathbf{X}\beta^{\text{pop}} + \epsilon$, where $\epsilon_i \stackrel{\text{iid}}{\sim} N(0, \sigma^2)$. The underlying regression coefficient was taken to be sparse with $\beta_i^{\text{pop}} = 1$ for all $i \leq 5$ and $\beta_i^{\text{pop}} = 0$ otherwise. σ^2 is chosen so as to control the signal to noise ratio SNR $:= \text{Var}(\mathbf{x}'\beta)/\sigma^2$.

Different values of SNR, n, p and ρ were taken and they have been specified in our results when and where appropriate.

• **Eg-B.** Here the datasets are generated similar to above, with $\beta_i^{\text{pop}} = 1$ for $i \leq 10$ and $\beta_i^{\text{pop}} = 0$ otherwise. We took SNR=1 in this example.

Real datasets. We considered four different publicly available microarray datasets as described below.

• **Leukemia dataset.** This dataset, taken from [11], was processed to have $n = 72$ and $p = 500$. $\mathbf{y}$ was created as $\mathbf{y} = \mathbf{X}\beta^{\text{pop}} + \epsilon$; with $\beta_i^{\text{pop}} = 1$ for all $i \leq 10$ and zero otherwise.

• **Golub dataset.** This dataset, taken from the R package `mpm`, was processed to have $n = 73$ and $p = 500$, with artificial responses generated as above.

• **Khan dataset.** This dataset, taken from the website of [33], was processed to have $n = 73$ and $p = 500$, with artificial responses generated as above.

• **Prostate dataset.** This dataset, analyzed in [14], was processed to create three types of different datasets: (a) the original dataset with $n = 97$ and $p = 8$, (b) a dataset with $n = 97$ and $p = 44$, formed by extending the covariate space to include second order interactions, and (c) a third dataset with $n = 10$ and $p = 44$, formed by subsampling the previous dataset.

For more detail on the above datasets, we refer the reader to the Supplementary Material [19] Section A.5.

Note that in all the examples we standardized $\mathbf{X}$ such that the columns have unit ℓ_2 norm, before running the different algorithms studied herein.

6.1. *Statistical properties of boosting algorithms: an empirical study.* We performed some experiments to better understand the statistical behavior of the different boosting methods described in this paper. We summarize our findings here; for details (including tables, figures and discussions) we refer the reader to Supplementary Material [19], Section A.5.

Sensitivity of the Learning Rate in LS-Boost(ε) *and* FS_ε. We explored how the training and test errors for LS-Boost(ε) and FS_ε change as a function of the number of boosting iterations and the learning rate. We observed that the best predictive models were sensitive to the choice of ε – the best models were obtained at values larger than zero and smaller than one. When compared to Lasso solutions, stepwise regression [14] and FS_0 [14]; FS_ε and

LS-Boost(ε) were found to be as good as the others, and in some cases were better than the rest.

Statistical properties of R-FS$_{\varepsilon,\delta}$, Lasso *solutions, and FS$_\varepsilon$: an empirical study.* We performed some experiments to evaluate the performance of R-FS$_{\varepsilon,\delta}$, in terms of predictive accuracy and sparsity of the optimal model, versus the more widely known methods FS$_\varepsilon$ and (solving the) Lasso. We found that when δ was larger than the best δ for the Lasso (in terms of obtaining a model with the best predictive performance), R-FS$_{\varepsilon,\delta}$ delivered a model with excellent statistical properties – R-FS$_{\varepsilon,\delta}$ led to sparse solutions and the predictive performance was as good as, and in some cases better than, the Lasso solution. We observed that the choice of δ does not play a very crucial role in the R-FS$_{\varepsilon,\delta}$ algorithm, once it is chosen to be reasonably large; indeed the number of boosting iterations play a more important role. The best models delivered by R-FS$_{\varepsilon,\delta}$ were more sparse than FS$_\varepsilon$.

Acknowledgements. The authors will like to thank Alexandre Belloni, Jerome Friedman, Trevor Hastie, Arian Maleki and Tomaso Poggio for helpful discussions and encouragement. A preliminary unpublished version of some of the results herein was posted on the ArXiv [17]. The authors would like to thank the Editor, Associate Editor and anonymous referees for helpful comments that have improved the paper.

SUPPLEMENTARY MATERIAL

Supplement A: Title
(doi: COMPLETED BY THE TYPESETTER; .pdf). Some text

References.

[1] Bach, F. (2015). Duality Between Subgradient and Conditional Gradient Methods. *SIAM Journal on Optimization* **25** 115–129.
[2] Bertsekas, D. (1999). *Nonlinear Programming*. Athena Scientific, Belmont, MA.
[3] Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, Cambridge.
[4] Breiman, L. (1998). Arcing classifiers (with discussion). *Annals of Statistics* **26** 801-849.
[5] Breiman, L. (1999). Prediction Games and Arcing Algorithms. *Neural Computation* **11** 1493-1517.
[6] Bühlmann, P. (2006). Boosting for high-dimensional linear models. *Annals of Statistics* 559–583.
[7] Bühlmann, P. and Hothorn, T. (2008). Boosting Algorithms: regularization, Prediction and Model Fitting (with discussion). *Statistical Science* **22** 477-505.
[8] Bühlmann, P. and Yu, B. (2003). Boosting with the L2 loss: regression and classification. *Journal of the American Statistical Association* **98** 324–339.

[9] Bühlmann, P. and Yu, B. (2006). Sparse boosting. *The Journal of Machine Learning Research* **7** 1001–1024.

[10] Clarkson, K. L. (2008). Coresets, Sparse Greedy Approximation, and the Frank-Wolfe Algorithm. *19th ACM-SIAM Symposium on Discrete Algorithms* 922-931.

[11] Dettling, M. and Bühlmann, P. (2003). Boosting for tumor classification with gene expression data. *Bioinformatics* **19** 1061–1069.

[12] Donoho, D. L., Johnstone, I. M., Kerkyacharian, G. and Picard, D. (1995). Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society. Series B (Methodological)* 301–369.

[13] Duchi, J. and Singer, Y. (2009). Boosting with structural sparsity. In *Proceedings of the 26th Annual International Conference on Machine Learning* 297–304. ACM.

[14] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression (with discussion). *Annals of Statistics* **32** 407–499. MR2060166 (2005d:62116)

[15] Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval Research Logistics Quarterly* **3** 95–110.

[16] Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Information and computation* **121** 256–285.

[17] Freund, R. M., Grigas, P. and Mazumder, R. (2013). AdaBoost and Forward Stagewise Regression are First-Order Convex Optimization Methods. *CoRR* **abs/1307.1192**.

[18] Freund, R. M. and Grigas, P. (2014). New Analysis and Results for the Frank-Wolfe Method. *to appear in Mathematical Programming*.

[19] Freund, R., Grigas, P. and Mazumder, R. (2017). Supplement to "A New Perspective on Boosting in Linear Regression via Subgradient Optimization and Relatives".

[20] Freund, Y. and Schapire, R. (1996). Experiments with a New Boosting Algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference* 148-156. Morgan Kauffman, San Francisco.

[21] Friedman, J. H. (2000). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* **29** 1189–1232.

[22] Friedman, J. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* **29** 1189–1232.

[23] Friedman, J. (2008). Fast Sparse Regression and Classification Technical Report, Department of Statistics, Stanford University.

[24] Friedman, J. H. (2012). Fast sparse regression and classification. *International Journal of Forecasting* **28** 722–738.

[25] Friedman, J., Hastie, T. and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion). *Annals of Statistics* **28** 337–307.

[26] Friedman, J. H. and Popescu, B. E. (2003). Importance sampled learning ensembles. *Journal of Machine Learning Research* **94305**.

[27] Friedman, J., Hastie, T., Hoefling, H. and Tibshirani, R. (2007). Pathwise coordinate optimization. *Annals of Applied Statistics* **2** 302-332.

[28] Gärtner, B., Giesen, J. and Jaggi, M. (2012). An Exponential Lower Bound on the Complexity of Regularization Paths. *Journal of Computational Geometry* **3** 168-195.

[29] Giesen, J., Jaggi, M. and Laue, S. (2012a). *Optimizing over the Growing Spectrahedron* In *Algorithms – ESA 2012: 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings* 503–514. Springer Berlin Heidelberg, Berlin, Heidelberg.

[30] Giesen, J., Jaggi, M. and Laue, S. (2012b). Approximating Parameterized Convex Optimization Problems. *ACM Trans. Algorithms* **9** 10:1–10:17.
[31] Giesen, J., Mueller, J., Laue, S. and Swiercy, S. (2012). Approximating Concavely Parameterized Optimization Problems. In *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou and K. Q. Weinberger, eds.) 2105–2113. Curran Associates, Inc.
[32] Hastie, T. J. and Tibshirani, R. J. (1990). *Generalized additive models* **43**. CRC Press.
[33] Hastie, T., Tibshirani, R. and Friedman, J. (2009). *Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Verlag, New York.
[34] Hastie, T., Taylor, J., Tibshirani, R. and Walther, G. (2007). Forward Stagewise Regression and the Monotone Lasso. *Electronic Journal of Statistics* **1** 1–29.
[35] Jaggi, M. (2013). Revisiting Frank-Wolfe: Projection-Free Sparse Convex Optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* 427–435.
[36] Mallat, S. G. and Zhang, Z. (1993). Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on* **41** 3397–3415.
[37] Mason, L., Baxter, J., Bartlett, P. and Frean, M. (2000). Boosting algorithms as gradient descent. **12** 512–518.
[38] Miller, A. (2002). *Subset selection in regression.* CRC Press Washington.
[39] Nesterov, Y. E. (2003). *Introductory lectures on convex optimization: a basic course. Applied Optimization* **87**. Kluwer Academic Publishers, Boston.
[40] Pati, Y. C., Rezaiifar, R. and Krishnaprasad, P. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on* 40–44. IEEE.
[41] Polyak, B. (1987). *Introduction to Optimization.* Optimization Software, Inc., New York.
[42] Rätsch, G., Onoda, T. and Müller, K.-R. (2001). Soft margins for AdaBoost. *Machine learning* **42** 287–320.
[43] Rosset, S., Zhu, J. and Hastie, T. (2004). Boosting as a Regularized Path to a Maximum Margin Classifier. *Journal of Machine Learning Research* **5** 941-973.
[44] Rosset, S., Swirszcz, G., Srebro, N. and Zhu, J. (2007). ℓ_1 regularization in infinite dimensional feature spaces. In *Conference on Learning theory* 544–558. Springer.
[45] Schapire, R. (1990). The strength of weak learnability. *Machine Learning* **5** 197-227.
[46] Schapire, R. E. and Freund, Y. (2012). *Boosting: Foundations and Algorithms. Adaptive computation and machine learning.* Mit Press.
[47] Shor, N. Z. (1985). *Minimization Methods for Non-Differentiable Functions. Springer Series in Computational Mathematics* **3**. Springer, Berlin.
[48] Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, Series B* **58** 267–288.
[49] Tibshirani, R. J. (2015). A General Framework for Fast Stagewise Algorithms. *Journal of Machine Learning Research* **16** 2543-2588.
[50] Tukey, J. (1977). *Exploratory data analysis.* Addison-Wesley,Massachusetts.
[51] Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027.*
[52] Weisberg, S. (1980). *Applied Linear Regression.* Wiley, New York.
[53] Zhao, P. and Yu, B. (2007). Stagewise lasso. *The Journal of Machine Learning Research* **8** 2701–2726.

SUPPLEMENTARY MATERIAL OF "A NEW PERSPECTIVE ON BOOSTING IN LINEAR REGRESSION VIA SUBGRADIENT OPTIMIZATION AND RELATIVES"

BY ROBERT M. FREUND, PAUL GRIGAS AND RAHUL MAZUMDER

A.1. Additional Details for Section 1.

A.1.1. *Figure showing training error versus ℓ_1-shrinkage bounds.* Figure A.1 shows profiles of ℓ_1 norm of the regression coefficients versus training error for LS-BOOST(ε) , FS_ε and LASSO.

ℓ_1 shrinkage versus data-fidelity tradeoffs: LS-BOOST(ε) , FS_ε, and LASSO

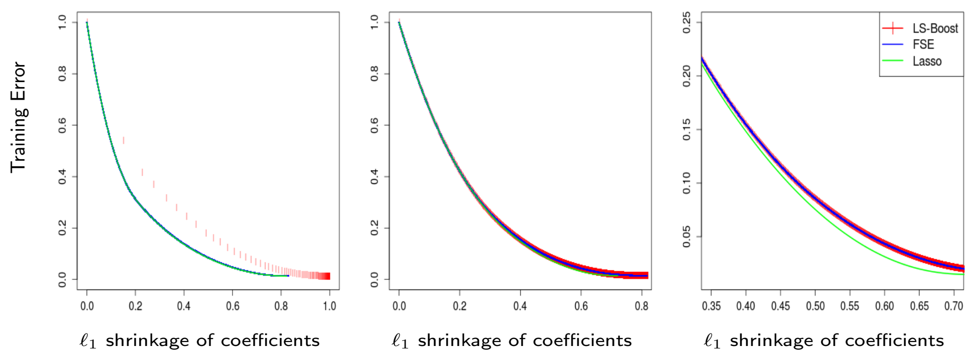

FIG A.1. *Figure showing profiles of ℓ_1 norm of the regression coefficients versus training error for* LS-BOOST(ε) *, FS_ε and* LASSO*. [Left panel] Shows profiles for a synthetic dataset where the covariates are drawn from a Gaussian distribution with pairwise correlations $\rho = 0.5$. The true β has ten non-zeros with $\beta_i = 1$ for $i = 1, \ldots, 10$, and SNR = 1. Here we ran* LS-BOOST(ε) *with $\varepsilon = 1$ and ran FS_ε with $\varepsilon = 10^{-2}$. The middle (and right) panel profiles corresponds to the Prostate cancer dataset (described in Section 6). Here we ran* LS-BOOST(ε) *with $\varepsilon = 0.01$ and we ran FS_ε with $\varepsilon = 10^{-5}$. The right panel figure is a zoomed-in version of the middle panel in order to highlight the difference in profiles between* LS-BOOST(ε) *, FS_ε and* LASSO*. The vertical axes have been normalized so that the training error at $k = 0$ is one, and the horizontal axes have been scaled to the unit interval (to express the ℓ_1-norm of $\hat{\beta}^k$ as a fraction of the maximum).*

A.2. Additional Details for Section 2.

A.2.1. *Properties of Convex Quadratic Functions.* Consider the following quadratic optimization problem (QP) defined as:

$$h^* := \min_{x \in \mathbb{R}^n} h(x) := \tfrac{1}{2} x^T Q x + q^T x + q^o \ ,$$

where Q is a symmetric positive semi-definite matrix, whereby $h(\cdot)$ is a convex function. We assume that $Q \neq 0$, and recall that $\lambda_{\text{pmin}}(Q)$ denotes the smallest nonzero (and hence positive) eigenvalue of Q.

PROPOSITION A.2.1. *If $h^* > -\infty$, then for any given x, there exists an optimal solution x^* of (QP) for which*

$$\|x - x^*\|_2 \leq \sqrt{\frac{2(h(x) - h^*)}{\lambda_{\text{pmin}}(Q)}} \ .$$

Also, it holds that

$$\|\nabla h(x)\|_2 \geq \sqrt{\frac{\lambda_{\text{pmin}}(Q) \cdot (h(x) - h^*)}{2}} \ .$$

Proof: The result is simply manipulation of linear algebra. Let us assume without loss of generality that $q^o = 0$. If $h^* > -\infty$, then (QP) has an optimal solution x^*, and the set of optimal solutions are characterized by the gradient condition

$$0 = \nabla h(x) = Qx + q \ .$$

Now let us write the sparse eigendecomposition of Q as $Q = PDP^T$ where D is a diagonal matrix of non-zero eigenvalues of Q and the columns of P are orthonormal, namely $P^T P = I$. Because (QP) has an optimal solution, the system of equations $Qx = -q$ has a solution, and let $\tilde{x}$ denote any such solution. Direct manipulation establishes:

$$PP^T q = -PP^T Q\tilde{x} = -PP^T PDP^T \tilde{x} = -PDP^T \tilde{x} = -Q\tilde{x} = q \ .$$

Furthermore, let $\hat{x} := -PD^{-1}P^T q$. It is then straightforward to see that $\hat{x}$ is an optimal solution of (QP) since in particular:

$$Q\hat{x} = -PDP^T PD^{-1}P^T q = -PP^T q = -q \ ,$$

and hence

$$h^* = \tfrac{1}{2}\hat{x}^T Q\hat{x} + q^T \hat{x} = -\tfrac{1}{2}\hat{x}^T Q\hat{x} = -\tfrac{1}{2} q^T PD^{-1}P^T PDP^T PD^{-1}P^T q = -\tfrac{1}{2} q^T PD^{-1}P^T q \ .$$

Now let x be given, and define $x^* := [I - PP^T]x - PD^{-1}P^T q$. Then just as above it is straightforward to establish that $Qx^* = -q$ whereby x^* is an optimal solution. Furthermore, it holds that:

$$\begin{aligned}
\|x - x^*\|_2^2 &= (q^T PD^{-1} + x^T P)P^T P(D^{-1}P^T q + P^T x) \\
&= (q^T PD^{-\frac{1}{2}} + x^T PD^{\frac{1}{2}})D^{-1}(D^{-\frac{1}{2}}P^T q + D^{\frac{1}{2}}P^T x) \\
&\leq \tfrac{1}{\lambda_{\mathrm{pmin}}(Q)}(q^T PD^{-\frac{1}{2}} + x^T PD^{\frac{1}{2}})(D^{-\frac{1}{2}}P^T q + D^{\frac{1}{2}}P^T x) \\
&= \tfrac{1}{\lambda_{\mathrm{pmin}}(Q)}(q^T PD^{-1}P^T q + x^T PDP^T x + 2x^T PP^T q) \\
&= \tfrac{1}{\lambda_{\mathrm{pmin}}(Q)}(-2h^* + x^T Qx + 2x^T q) \\
&= \tfrac{2}{\lambda_{\mathrm{pmin}}(Q)}(h(x) - h^*) \ ,
\end{aligned}$$

and taking square roots establishes the first inequality of the proposition.

Using the gradient inequality for convex functions, it holds that:

$$\begin{aligned}
h^* = h(x^*) &\geq h(x) + \nabla h(x)^T(x^* - x) \\
&\geq h(x) - \|\nabla h(x)\|_2 \|x^* - x\|_2 \\
&\geq h(x) - \|\nabla h(x)\|_2 \sqrt{\tfrac{2(h(x) - h^*)}{\lambda_{\mathrm{pmin}}(Q)}} \ ,
\end{aligned}$$

and rearranging the above proves the second inequality of the proposition. $\square$

A.2.2. *Proof of Theorem 2.1.* We first prove part *(i)*. Utilizing (2.9), which states that $\hat{r}^{k+1} = \hat{r}^k - \varepsilon\left((\hat{r}^k)^T \mathbf{X}_{j_k}\right)\mathbf{X}_{j_k}$, we have:

$$\begin{aligned}
L_n(\hat{\beta}^{k+1}) &= \tfrac{1}{2n}\|\hat{r}^{k+1}\|_2^2 \\
&= \tfrac{1}{2n}\|\hat{r}^k - \varepsilon\left((\hat{r}^k)^T \mathbf{X}_{j_k}\right)\mathbf{X}_{j_k}\|_2^2 \\
&= \tfrac{1}{2n}\|\hat{r}^k\|_2^2 - \tfrac{1}{n}\varepsilon\left((\hat{r}^k)^T \mathbf{X}_{j_k}\right)^2 + \tfrac{1}{2n}\varepsilon^2\left((\hat{r}^k)^T \mathbf{X}_{j_k}\right)^2 \qquad \text{(A.2.1)} \\
&= L_n(\hat{\beta}^k) - \tfrac{1}{2n}\varepsilon(2-\varepsilon)\left((\hat{r}^k)^T \mathbf{X}_{j_k}\right)^2 \\
&= L_n(\hat{\beta}^k) - \tfrac{1}{2n}\varepsilon(2-\varepsilon)n^2\|\nabla L_n(\hat{\beta}^k)\|_\infty^2 \ ,
\end{aligned}$$

(where the last equality above uses (2.8)), which yields:

$$L_n(\hat\beta^{k+1}) - L_n^* = L_n(\hat\beta^k) - L_n^* - \tfrac{n}{2}\varepsilon(2-\varepsilon)\|\nabla L_n(\hat\beta^k)\|_\infty^2 \ . \tag{A.2.2}$$

We next seek to bound the right-most term above. We will do this by invoking Proposition A.2.1, which presents two important properties of convex quadratic functions. Because $L_n(\cdot)$ is a convex quadratic function of the same format as Proposition A.2.1 with $h(\cdot) \leftarrow L_n(\cdot)$, $Q \leftarrow \frac{1}{n}\mathbf{X}^T\mathbf{X}$, and $h^* \leftarrow L_n^*$, it follows from the second property of Proposition A.2.1 that

$$\|\nabla L_n(\beta)\|_2 \geq \sqrt{\frac{\lambda_{\text{pmin}}(\frac{1}{n}\mathbf{X}^T\mathbf{X})(L_n(\beta) - L_n^*)}{2}} = \sqrt{\frac{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})(L_n(\beta) - L_n^*)}{2n}} \ .$$

Therefore

$$\|\nabla L_n(\beta)\|_\infty^2 \geq \tfrac{1}{p}\|\nabla L_n(\beta)\|_2^2 \geq \frac{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})(L_n(\beta) - L_n^*)}{2np} \ .$$

Substituting this inequality into (A.2.2) yields after rearranging:

$$L_n(\hat\beta^{k+1}) - L_n^* \leq (L_n(\hat\beta^k) - L_n^*)\left(1 - \frac{\varepsilon(2-\varepsilon)\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}{4p}\right) = (L_n(\hat\beta^k) - L_n^*)\cdot\gamma \ . \tag{A.2.3}$$

Now note that $L_n(\hat\beta^0) = L_n(0) = \frac{1}{2n}\|\mathbf{y}\|_2^2$ and

$$\begin{aligned} L_n(\hat\beta^0) - L_n^* &= \tfrac{1}{2n}\|\mathbf{y}\|_2^2 - \tfrac{1}{2n}\|\mathbf{y} - \mathbf{X}\hat\beta_{\text{LS}}\|_2^2 \\ &= \tfrac{1}{2n}\|\mathbf{y}\|_2^2 - \tfrac{1}{2n}(\|\mathbf{y}\|_2^2 - 2\mathbf{y}^T\mathbf{X}\hat\beta_{\text{LS}} + \|\mathbf{X}\hat\beta_{\text{LS}}\|_2^2) \\ &= \tfrac{1}{2n}\|\mathbf{X}\hat\beta_{\text{LS}}\|_2^2 \ , \end{aligned}$$

where the last equality uses the normal equations (2.3). Then *(i)* follows by using elementary induction and combining the above with (A.2.3):

$$L_n(\hat\beta^k) - L_n^* \leq (L_n(\hat\beta^0) - L_n^*)\cdot\gamma^k = \tfrac{1}{2n}\|\mathbf{X}\hat\beta_{\text{LS}}\|_2^2\cdot\gamma^k \ .$$

To prove *(ii)*, we invoke the first inequality of Proposition A.2.1, which in this context states that

$$\|\hat\beta^k - \hat\beta_{\text{LS}}\|_2 \leq \frac{\sqrt{2(L_n(\hat\beta^k) - L_n^*)}}{\sqrt{\lambda_{\text{pmin}}(\frac{1}{n}\mathbf{X}^T\mathbf{X})}} = \frac{\sqrt{2n(L_n(\hat\beta^k) - L_n^*)}}{\sqrt{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}} \ .$$

Part *(ii)* then follows by substituting the bound on $(L_n(\hat\beta^k) - L_n^*)$ from *(i)* and simplifying terms. Similarly, the proof of *(iii)* follows from the observation that $\|\mathbf{X}\hat\beta^k - \mathbf{X}\hat\beta_{\text{LS}}\|_2 = \sqrt{2n(L_n(\hat\beta^k) - L_n^*)}$ and then substituting the bound on $(L_n(\hat\beta^k) - L_n^*)$ from *(i)* and simplifying terms.

To prove *(iv)*, define the point $\tilde{\beta}^k := \hat{\beta}^k + \tilde{u}_{j_k} e_{j_k}$. Then using similar arithmetic as in (A.2.1) one obtains:

$$L_n^* \le L_n(\tilde{\beta}^k) = L_n(\hat{\beta}^k) - \tfrac{1}{2n}\tilde{u}_{j_k}^2 \ ,$$

where we recall that $\tilde{u}_{j_k} = (\hat{r}^k)^T \mathbf{X}_{j_k}$. This inequality then rearranges to

$$|\tilde{u}_{j_k}| \ \le\ \sqrt{2n(L_n(\hat{\beta}^k) - L_n^*)} \ \le\ \|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2 \cdot \gamma^{k/2} \ , \tag{A.2.4}$$

where the second inequality follows by substituting the bound on $(L_n(\hat{\beta}^i) - L_n^*)$ from *(i)*. Recalling (2.4) and (2.8), the above is exactly part *(iv)*.

Part *(v)* presents two distinct bounds on $\|\hat{\beta}^k\|_1$, which we prove independently. To prove the first bound, let $\hat{\beta}_{\mathrm{LS}}$ be any least-squares solution, which therefore satisfies (2.3). It is then elementary to derive using similar manipulation as in (A.2.1) that for all i the following holds:

$$\|\mathbf{X}(\hat{\beta}^{i+1} - \hat{\beta}_{\mathrm{LS}})\|_2^2 = \|\mathbf{X}(\hat{\beta}^i - \hat{\beta}_{\mathrm{LS}})\|_2^2 - (2\varepsilon - \varepsilon^2)\tilde{u}_{j_i}^2 \tag{A.2.5}$$

which implies that

$$(2\varepsilon - \varepsilon^2)\sum_{i=0}^{k-1} \tilde{u}_{j_i}^2 = \|\mathbf{X}(\hat{\beta}^0 - \hat{\beta}_{\mathrm{LS}})\|_2^2 - \|\mathbf{X}(\hat{\beta}^k - \hat{\beta}_{\mathrm{LS}})\|_2^2 = \|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2^2 - \|\mathbf{X}(\hat{\beta}^k - \hat{\beta}_{\mathrm{LS}})\|_2^2 \ . \tag{A.2.6}$$

Then note that

$$\begin{aligned} \|\hat{\beta}^k\|_1 &\le \|(\varepsilon\tilde{u}_{j_0}, \ldots, \varepsilon\tilde{u}_{j_{k-1}})\|_1 \\ &\le \sqrt{k}\varepsilon\|(\tilde{u}_{j_0}, \ldots, \tilde{u}_{j_{k-1}})\|_2 \\ &= \sqrt{k}\sqrt{\tfrac{\varepsilon}{2-\varepsilon}}\sqrt{\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2^2 - \|\mathbf{X}\hat{\beta}_{\mathrm{LS}} - \mathbf{X}\hat{\beta}^k\|_2^2} \ , \end{aligned} \tag{A.2.7}$$

where the last equality is from (A.2.6).

To prove the second bound in *(v)*, noting that $\hat{\beta}^k = \sum_{i=0}^{k-1} \varepsilon\tilde{u}_{j_i} e_{j_i}$, we bound $\|\hat{\beta}^k\|_1$ as follows:

$$\begin{aligned} \|\hat{\beta}^k\|_1 \le \varepsilon\sum_{i=0}^{k-1} |\tilde{u}_{j_i}| \ &\le\ \varepsilon\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2 \sum_{i=0}^{k-1} \gamma^{i/2} \\ &= \frac{\varepsilon\|\mathbf{X}\hat{\beta}_{LS}\|_2}{1 - \sqrt{\gamma}}\left(1 - \gamma^{k/2}\right) \ , \end{aligned}$$

where the second inequality uses (A.2.4) for each $i \in \{0, \ldots, k-1\}$ and the final equality is a geometric series, which completes the proof of *(v)*. Part *(vi)* is simply the property of LS-Boost(ε) that derives from the fact that $\hat{\beta}^0 := 0$ and at every iteration at most one coordinate of β changes status from a zero to a non-zero value. □

A.2.3. *Additional properties of* LS-BOOST(ε). We present two other interesting properties of the LS-BOOST(ε) algorithm, namely an additional bound on the correlation between residuals and predictors, and a bound on the ℓ_2-shrinkage of the regression coefficients. Both are presented in the following proposition.

PROPOSITION A.2.2. **(Two additional properties of LS-Boost(ε))** *Consider the iterates of the* LS-BOOST(ε) *algorithm after k iterations and consider the linear convergence rate coefficient γ:*

$$\gamma := \left(1 - \frac{\varepsilon(2-\varepsilon)\lambda_{\mathrm{pmin}}(\mathbf{X}^T\mathbf{X})}{4p}\right).$$

(i) There exists an index $i \in \{0,\ldots,k\}$ for which the ℓ_∞ norm of the gradient vector of the least squares loss function evaluated at $\hat{\beta}^i$ satisfies:

$$\begin{array}{rl} & \|\nabla L_n(\hat{\beta}^i)\|_\infty \\ = & \frac{1}{n}\|\mathbf{X}^T\hat{r}^i\|_\infty \\ \le & \min\left\{ \dfrac{\sqrt{\|\mathbf{X}\hat{\beta}_{LS}\|_2^2 - \|\mathbf{X}\hat{\beta}_{LS} - \mathbf{X}\hat{\beta}^{k+1}\|_2^2}}{n\sqrt{\varepsilon(2-\varepsilon)(k+1)}} \ , \frac{1}{n}\|\mathbf{X}\hat{\beta}_{LS}\|_2 \cdot \gamma^{k/2} \right\} . \end{array} \tag{A.2.8}$$

(ii) Let J_ℓ denote the number of iterations of LS-BOOST(ε) *, among the first k iterations, where the algorithm takes a step in coordinate ℓ, for $\ell = 1,\ldots,p$, and let $J_{\max} := \max\{J_1,\ldots,J_p\}$. Then the following bound on the shrinkage of $\hat{\beta}^k$ holds:*

$$\|\hat{\beta}^k\|_2 \le \sqrt{J_{\max}}\sqrt{\frac{\varepsilon}{2-\varepsilon}}\sqrt{\|\mathbf{X}\hat{\beta}_{LS}\|_2^2 - \|\mathbf{X}\hat{\beta}_{LS} - \mathbf{X}\hat{\beta}^k\|_2^2} \ . \tag{A.2.9}$$

□

PROOF. We first prove part *(i)*. The first equality of (A.2.8) is a restatement of (2.8). For each $i \in \{0,\ldots,k\}$, recall that $\tilde{u}_{j_i} = (\hat{r}^i)^T\mathbf{X}_{j_i}$ and that $|\tilde{u}_{j_i}| = |(\hat{r}^i)^T\mathbf{X}_{j_i}| = \|\mathbf{X}^T\hat{r}^i\|_\infty$, from (2.8). Therefore:

$$\left(\min_{i\in\{0,\ldots,k\}} |\tilde{u}_{j_i}|\right)^2 = \min_{i\in\{0,\ldots,k\}} \tilde{u}_{j_i}^2 \le \frac{1}{k+1}\sum_{i=0}^{k} \tilde{u}_{j_i}^2 \le \frac{\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2^2 - \|\mathbf{X}(\hat{\beta}^{k+1} - \hat{\beta}_{\mathrm{LS}})\|_2^2}{\varepsilon(2-\varepsilon)(k+1)} \ , \tag{A.2.10}$$

where the final inequality follows from (A.2.6) in the proof of Theorem 2.1. Now letting i be an index achieving the minimum in the left hand side of the above and taking square roots implies that

$$\|\mathbf{X}^T \hat{r}^i\|_\infty = |\tilde{u}_{j_i}| \le \frac{\sqrt{\|\mathbf{X}\hat{\beta}_{\rm LS}\|_2^2 - \|\mathbf{X}\hat{\beta}_{\rm LS} - \mathbf{X}\hat{\beta}^{k+1}\|_2^2}}{\sqrt{\varepsilon(2-\varepsilon)(k+1)}} \ ,$$

which is equivalent to the inequality in (A.2.8) for the first right-most term therein. Directly applying (A.2.4) from the proof of Theorem 2.1 and using the fact that i is an index achieving the minimum in the left hand side of (A.2.10) yields:

$$\|\mathbf{X}^T \hat{r}^i\|_\infty = |\tilde{u}_{j_i}| \le |\tilde{u}_{j_k}| \le \|\mathbf{X}\hat{\beta}_{\rm LS}\|_2 \left(1 - \frac{\varepsilon(2-\varepsilon)\lambda_{\rm pmin}(\mathbf{X}^T\mathbf{X})}{4p}\right)^{k/2} \ ,$$

which is equivalent to the inequality in (A.2.8) for the second right-most term therein.

We now prove part *(ii)*. For fixed $k > 0$, let $\mathcal{J}(\ell)$ denote the set of iteration counters where LS-Boost(ε) modifies coordinate ℓ of β, namely

$$\mathcal{J}(\ell) := \{ i : i < k \text{ and } j_i = \ell \text{ in Step (2.) of Algorithm LS-Boost}(\varepsilon) \} \ ,$$

for $\ell = 1, \ldots, p$. Then $J_\ell = |\mathcal{J}(\ell)|$, and the sets $\mathcal{J}(1), \ldots, \mathcal{J}(p)$ partition the iteration index set $\{0, 1, \ldots, k-1\}$. We have:

(A.2.11)

$$\begin{aligned}
\|\hat{\beta}^k\|_2 &\le \|(\textstyle\sum_{i \in \mathcal{J}(1)} \varepsilon \tilde{u}_{j_i}, \ldots, \sum_{i \in \mathcal{J}(p)} \varepsilon \tilde{u}_{j_i})\|_2 \\
&\le \left\| \left(\sqrt{J(1)} \sqrt{\textstyle\sum_{i \in \mathcal{J}(1)} \varepsilon^2 \tilde{u}_{j_i}^2}, \ldots, \sqrt{J(p)} \sqrt{\textstyle\sum_{i \in \mathcal{J}(p)} \varepsilon^2 \tilde{u}_{j_i}^2} \right) \right\|_2 \\
&\le \varepsilon \sqrt{J_{\max}} \left\| \left(\sqrt{\textstyle\sum_{i \in \mathcal{J}(1)} \tilde{u}_{j_i}^2}, \ldots, \sqrt{\textstyle\sum_{i \in \mathcal{J}(p)} \tilde{u}_{j_i}^2} \right) \right\|_2 \\
&= \varepsilon \sqrt{J_{\max}} \sqrt{\left(\tilde{u}_{j_0}^2 + \ldots + \tilde{u}_{j_{k-1}}^2 \right)} \ ,
\end{aligned}$$

and the proof is completed by applying inequality (A.2.6). □

Part *(i)* of Proposition A.2.2 describes the behavior of the gradient of the least squares loss function — indeed, recall that the dynamics of the gradient are closely linked to that of the LS-Boost(ε) algorithm and, in particular, to the evolution of the loss function values. To illustrate this connection, let us recall two simple characteristics of the LS-Boost(ε) algorithm:

$$\begin{aligned}
L_n(\hat{\beta}^k) - L_n(\hat{\beta}^{k+1}) &= \tfrac{n}{2}\varepsilon(2-\varepsilon)\|\nabla L_n(\hat{\beta}^k)\|_\infty^2 \\
\hat{r}^{k+1} - \hat{r}^k &= -\varepsilon \left((\hat{r}^k)^T \mathbf{X}_{j_k} \right) \mathbf{X}_{j_k} \ ,
\end{aligned}$$

which follow from (A.2.2) and Step (3.) of the LS-Boost(ε) algorithm, respectively. The above updates of the LS-Boost(ε) algorithm clearly show that smaller values of the ℓ_∞ norm of the gradient slows down the "progress" of the residuals and thus the overall algorithm. Larger values of the norm of the gradient, on the other hand, lead to rapid "progress" in the algorithm. Here, we use the term "progress" to measure the amount of decrease in training error and the norm of the changes in successive residuals. Informally speaking, the LS-Boost(ε) algorithm operationally works towards minimizing the unregularized least squares loss function — and the gradient of the least squares loss function is simultaneously shrunk towards zero. Equation (A.2.8) precisely quantifies the rate at which the ℓ_∞ norm of the gradient converges to zero. Observe that the bound is a minimum of two distinct rates: one which decays as $O(\frac{1}{\sqrt{k}})$ and another which is a linear rate of convergence with parameter $\sqrt{\gamma}$. This is similar to item *(v)* of Theorem 2.1. For small values of k the first rate will dominate, until a point is reached where the linear rate begins to dominate. Note that the dependence on the linear rate γ suggests that for large values of correlations among the samples, the gradient decays slower than for smaller pairwise correlations among the samples.

The behavior of the LS-Boost(ε) algorithm described above should be contrasted with the FS$_\varepsilon$ algorithm. In view of Step (3.) of the FS$_\varepsilon$ algorithm, the successive differences of the residuals in FS$_\varepsilon$ are indifferent to the magnitude of the gradient of the least squares loss function — as long as the gradient is non-zero, then for FS$_\varepsilon$ it holds that $\|\hat{r}^{k+1} - \hat{r}^k\|_2 = \varepsilon$. Thus FS$_\varepsilon$ undergoes a more erratic evolution, unlike LS-Boost(ε) where the convergence of the residuals is much more "smooth."

A.2.4. *Concentration Results for $\lambda_{\rm pmin}(\mathbf{X}^T\mathbf{X})$ in the High-dimensional Case.*

Proposition A.2.3. *Suppose that $p > n$, let $\tilde{\mathbf{X}} \in \mathbb{R}^{n\times p}$ be a random matrix whose entries are i.i.d. standard normal random variables, and define $\mathbf{X} := \frac{1}{\sqrt{n}}\tilde{\mathbf{X}}$. Then it holds that:*

$$\mathbb{E}[\lambda_{\rm pmin}(\mathbf{X}^T\mathbf{X})] \geq \frac{1}{n}\left(\sqrt{p} - \sqrt{n}\right)^2 \ .$$

Furthermore, for every $t \in [0, \sqrt{p}-\sqrt{n}]$, with probability at least $1-2\exp(-t^2/2)$ it holds that:

$$\lambda_{\rm pmin}(\mathbf{X}^T\mathbf{X}) \geq \frac{1}{n}\left(\sqrt{p} - \sqrt{n} - t\right)^2 \ .$$

PROOF. Let $\sigma_1(\tilde{\mathbf{X}}^T) \geq \sigma_2(\tilde{\mathbf{X}}^T) \geq \ldots \geq \sigma_n(\tilde{\mathbf{X}}^T)$ denote the ordered singular values of $\tilde{\mathbf{X}}^T$ (equivalently of $\tilde{\mathbf{X}}$). Then, Theorem 5.32 of [51] states that:

$$\mathbb{E}[\sigma_n(\tilde{\mathbf{X}}^T)] \geq \sqrt{p} - \sqrt{n} \ ,$$

which thus implies:

$$\begin{aligned}\mathbb{E}[\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})] =& \mathbb{E}[(\sigma_n(\mathbf{X}^T))^2] \\ \geq& (\mathbb{E}[\sigma_n(\mathbf{X}^T)])^2 \\ =& \frac{1}{n}(\mathbb{E}[\sigma_n(\tilde{\mathbf{X}}^T)])^2 \\ \geq& \frac{1}{n}\left(\sqrt{p} - \sqrt{n}\right)^2 \ ,\end{aligned}$$

where the first inequality is Jensen's inequality.

Corollary 5.35 of [51] states that for every $t \geq 0$, with probability at least $1 - 2\exp(-t^2/2)$ it holds that:

$$\sigma_n(\tilde{\mathbf{X}}^T) \geq \sqrt{p} - \sqrt{n} - t \ ,$$

which implies for $t \in [0, \sqrt{p} - \sqrt{n}]$ that:

$$\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X}) = (\sigma_n(\mathbf{X}^T))^2 = \frac{1}{n}(\sigma_n(\tilde{\mathbf{X}}^T))^2 \geq \frac{1}{n}\left(\sqrt{p} - \sqrt{n} - t\right)^2 \ .$$

□

Note that in practice we standardize the model matrix $\mathbf{X}$ so that its columns have unit ℓ_2 norm. Supposing that the entries of $\mathbf{X}$ did originate from an i.i.d. standard normal matrix $\tilde{\mathbf{X}}$, standardizing the columns of $\tilde{\mathbf{X}}$ is not equivalent to setting $\mathbf{X} := \frac{1}{\sqrt{n}}\tilde{\mathbf{X}}$. But, for large enough n, standardizing is a valid approximation to normalizing by $\frac{1}{\sqrt{n}}$, i.e., $\mathbf{X} \approx \frac{1}{\sqrt{n}}\tilde{\mathbf{X}}$, and we may thus apply the above results.

Additional Details for Section 3.

A.2.5. *An Elementary Sequence Process Result, and a Proof of Proposition 3.1.* Consider the following elementary sequence process: $x^0 \in \mathbb{R}^n$ is given, and $x^{i+1} \leftarrow x^i - \alpha_i g^i$ for all $i \geq 0$, where $g^i \in \mathbb{R}^n$ and α_i is a nonnegative scalar, for all i. For this process there are *no* assumptions on how the vectors g^i might be generated.

PROPOSITION A.2.4. *For the elementary sequence process described above, suppose that the $\{g^i\}$ are uniformly bounded, namely $\|g^i\|_2 \le G$ for all $i \ge 0$. Then for all $k \ge 0$ and for any $x \in \mathrm{R}^n$ it holds that:*

$$\frac{1}{\sum_{i=0}^k \alpha_i} \sum_{i=0}^k \alpha_i (g^i)^T (x^i - x) \quad \le \quad \frac{\|x^0 - x\|_2^2 + G^2 \sum_{i=0}^k \alpha_i^2}{2 \sum_{i=0}^k \alpha_i} \ . \tag{A.2.12}$$

Indeed, in the case when $\alpha_i = \varepsilon$ for all i, it holds that:

$$\frac{1}{k+1} \sum_{i=0}^k (g^i)^T (x^i - x) \quad \le \quad \frac{\|x^0 - x\|_2^2}{2(k+1)\varepsilon} + \frac{G^2 \varepsilon}{2} \ . \tag{A.2.13}$$

PROOF. Elementary arithmetic yields the following:

$$\begin{aligned}
\|x^{i+1} - x\|_2^2 &= \|x^i - \alpha_i g^i - x\|_2^2 \\
&= \|x^i - x\|_2^2 + \alpha_i^2 \|g^i\|_2^2 + 2\alpha_i (g^i)^T (x - x^i) \\
&\le \|x^i - x\|_2^2 + G^2 \alpha_i^2 + 2\alpha_i (g^i)^T (x - x^i) \ .
\end{aligned}$$

Rearranging and summing these inequalities for $i = 0, \ldots, k$ then yields:

$$2 \sum_{i=0}^k \alpha_i (g^i)^T (x^i - x) \le G^2 \sum_{i=0}^k \alpha_i^2 + \|x^0 - x\|_2^2 - \|x^{k+1} - x\|_2^2 \le G^2 \sum_{i=0}^k \alpha_i^2 + \|x^0 - x\|_2^2 \ ,$$

which then rearranges to yield (A.2.12). (A.2.13) follows from (A.2.12) by direct substitution. □

Proof of Proposition 3.1: Consider the subgradient descent method (3.5) with arbitrary step-sizes α_i for all i. We will prove the following inequality:

$$\min_{i \in \{0,\ldots,k\}} f(x^i) \quad \le \quad f^* \ + \ \frac{\|x^0 - x^*\|_2^2 + G^2 \sum_{i=0}^k \alpha_i^2}{2 \sum_{i=0}^k \alpha_i} \ , \tag{A.2.14}$$

from which the proof of Proposition 3.1 follows by substituting $\alpha_i = \alpha$ for all i and simplifying terms. Let us now prove (A.2.14). The subgradient descent method (3.5) is applied to instances of problem (3.3) where $f(\cdot)$ is convex, and where g^i is a subgradient of $f(\cdot)$ at x^i, for all i. If x^* is an optimal solution of (3.3), it therefore holds from the subgradient inequality that

$$f^* = f(x^*) \geq f(x^i) + (g^i)^T(x^* - x^i) \ .$$

Substituting this inequality in (A.2.12) for the value of $x = x^*$ yields:

$$\begin{aligned}\frac{\|x^0 - x^*\|_2^2 + G^2 \sum_{i=0}^k \alpha_i^2}{2\sum_{i=0}^k \alpha_i} &\geq \frac{1}{\sum_{i=0}^k \alpha_i} \sum_{i=0}^k \alpha_i (g^i)^T(x^i - x^*) \\ &\geq \frac{1}{\sum_{i=0}^k \alpha_i} \sum_{i=0}^k \alpha_i (f(x^i) - f^*) \geq \min_{i \in \{0,\ldots,k\}} f(x^i) - f^* \ .\end{aligned}$$

□

A.2.6. *Proof of Theorem 3.1.* We first prove part *(i)*. Note that item *(i)* of Proposition 3.2 shows that FS_ε is a specific instance of subgradient descent to solve problem (3.7), using the constant step-size ε. Therefore we can apply the computational guarantees associated with the subgradient descent method, particularly Proposition 3.1, to the FS_ε algorithm. Examining Proposition 3.1, we need to work out the corresponding values of f^*, $\|x^0 - x^*\|_2$, α, and G in the context of FS_ε for solving the CM problem (3.7). Note that $f^* = 0$ for problem (3.7). We bound the distance from the initial residuals to the optimal least-squares residuals as follows:

$$\|\hat{r}^0 - r^*\|_2 = \|\hat{r}^0 - \hat{r}_{LS}\|_2 = \|\mathbf{y} - (\mathbf{y} - \mathbf{X}\hat{\beta}_{\mathrm{LS}})\|_2 = \|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2 \ .$$

From Proposition 3.2 part *(i)* we have $\alpha = \varepsilon$. Last of all, we need to determine an upper bound G on the norms of subgradients. We have:

$$\|g^k\|_2 = \|\mathrm{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k})\mathbf{X}_{j_k}\|_2 = \|\mathbf{X}_{j_k}\|_2 = 1 \ ,$$

since the covariates have been standardized, so we can set $G = 1$. Now suppose algorithm FS_ε is run for k iterations. Proposition 3.1 then implies that:

(A.2.15)

$$\min_{i \in \{0,\ldots,k\}} \|\mathbf{X}^T \hat{r}^i\|_\infty = \min_{i \in \{0,\ldots,k\}} f(\hat{r}^i) \leq f^* + \frac{\|\hat{r}^0 - r^*\|_2^2}{2\alpha(k+1)} + \frac{\alpha G^2}{2} = \frac{\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2^2}{2\varepsilon(k+1)} + \frac{\varepsilon}{2} \ .$$

The above inequality provides a bound on the best (among the first k residual iterates) empirical correlation between the residuals $\hat{r}^i$ and each predictor variable, where the bound depends explicitly on the learning rate ε and the number of iterations k. Furthermore, invoking (2.4), the above inequality implies the following upper bound on the norm of the gradient of the least squares loss $L_n(\cdot)$ for the model iterates $\{\hat{\beta}^i\}$ generated by FS_ε:

$$\min_{i\in\{0,\ldots,k\}} \|\nabla L_n(\hat{\beta}^i)\|_\infty \leq \frac{\|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2^2}{2n\varepsilon(k+1)} + \frac{\varepsilon}{2n} \ . \tag{A.2.16}$$

Let i be the index where the minimum is attained on the left side of the above inequality. In a similar vein as in the analysis in Section 2, we now use Proposition A.2.1 which presents two important properties of convex quadratic functions. Because $L_n(\cdot)$ is a convex quadratic function of the same format as Proposition A.2.1 with $h(\cdot) \leftarrow L_n(\cdot)$, $Q \leftarrow \frac{1}{n}\mathbf{X}^T\mathbf{X}$, and $h^* \leftarrow L_n^*$, it follows from the second property of Proposition A.2.1 that

$$\|\nabla L_n(\hat{\beta}^i)\|_2 \geq \sqrt{\frac{\lambda_{\text{pmin}}(\frac{1}{n}\mathbf{X}^T\mathbf{X})(L_n(\hat{\beta}^i) - L_n^*)}{2}} = \sqrt{\frac{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})(L_n(\hat{\beta}^i) - L_n^*)}{2n}} \ ,$$

where recall that $\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})$ denotes the smallest non-zero (hence positive) eigenvalue of $\mathbf{X}^T\mathbf{X}$. Therefore

$$\|\nabla L_n(\hat{\beta}^i)\|_\infty^2 \geq \tfrac{1}{p}\|\nabla L_n(\hat{\beta}^i)\|_2^2 \geq \frac{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})(L_n(\hat{\beta}^i) - L_n^*)}{2np} \ .$$

Substituting this inequality into (A.2.16) for the index i where the minimum is attained yields after rearranging:

$$L_n(\hat{\beta}^i) - L_n^* \leq \frac{p}{2n\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}\left[\frac{\|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2^2}{\varepsilon(k+1)} + \varepsilon\right]^2 \ , \tag{A.2.17}$$

which proves part *(i)*. The proof of part *(ii)* follows by noting from the first inequality of Proposition A.2.1 that there exists a least-squares solution $\hat{\beta}^*$ for which:

$$\begin{aligned}
\|\hat{\beta}^* - \hat{\beta}^i\|_2 &\leq \sqrt{\frac{2(L_n(\hat{\beta}^i) - L_n^*)}{\lambda_{\text{pmin}}\left(\frac{1}{n}\mathbf{X}^T\mathbf{X}\right)}} \\
&= \sqrt{\frac{2n(L_n(\hat{\beta}^i) - L_n^*)}{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}} \\
&\leq \frac{\sqrt{p}}{\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})}\left[\frac{\|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2^2}{\varepsilon(k+1)} + \varepsilon\right] \ ,
\end{aligned}$$

where the second inequality in the above chain follows using (A.2.17). The proof of part *(iii)* follows by first observing that $\|\mathbf{X}(\hat{\beta}^i - \hat{\beta}_{\rm LS})\|_2 = \sqrt{2n(L_n(\hat{\beta}^i) - L_n^*)}$ and then substituting the bound on $(L_n(\hat{\beta}^i) - L_n^*)$ from part *(i)* and simplifying terms. Part *(iv)* is a restatement of inequality (A.2.15). Finally, parts *(v)* and *(vi)* are simple and well-known structural properties of FS_ε that are re-stated here for completeness. □

A.2.7. *A deeper investigation of the computational guarantees for* LS-Boost(ε) *and* FS_ε. Here we show that, at least in theory, LS-Boost(ε) is much more efficient than FS_ε if the primary goal is to obtain a model with a certain (pre-specified) data-fidelity. To formalize this notion, we consider a parameter $\tau \in (0, 1]$. We say that $\bar{\beta}$ is at a τ-relative distance to the least squares predictions if $\bar{\beta}$ satisfies:

$$\|\mathbf{X}\bar{\beta} - \mathbf{X}\hat{\beta}_{\rm LS}\|_2 \le \tau \|\mathbf{X}\hat{\beta}_{\rm LS}\|_2 \ . \tag{A.2.18}$$

Now let us pose the following question: if both LS-Boost(ε) and FS_ε are allowed to run with an appropriately chosen learning rate ε for each algorithm, which algorithm will satisfy (A.2.18) in fewer iterations? We will answer this question by studying closely the computational guarantees of Theorems 2.1 and 3.1. Since our primary goal is to compute $\bar{\beta}$ satisfying (A.2.18), we may optimize the learning rate ε, for each algorithm, to achieve this goal with the smallest number of boosting iterations.

Let us first study LS-Boost(ε) . As we have seen, a learning rate of $\varepsilon = 1$ achieves the fastest rate of linear convergence for LS-Boost(ε) and is thus optimal with regard to the bound in part *(iii)* of Theorem 2.1. If we run LS-Boost(ε) with $\varepsilon = 1$ for $k^{\text{LS-Boost}(\varepsilon)} := \left\lceil \frac{4p}{\lambda_{\rm pmin}(\mathbf{X}^T\mathbf{X})} \ln\left(\frac{1}{\tau^2}\right) \right\rceil$ iterations, then it follows from part *(iii)* of Theorem 2.1 that we achieve (A.2.18). Furthermore, it follows from (3.9) that the resulting ℓ_1-shrinkage bound will satisfy:

$$\text{Sbound}^{\text{LS-Boost}(\varepsilon)} \le \|\mathbf{X}\hat{\beta}_{\rm LS}\|_2 \sqrt{k^{\text{LS-Boost}(\varepsilon)}} \ .$$

For FS_ε, if one works out the arithmetic based on part *(iii)* of Theorem 3.1, the optimal number of boosting iterations to achieve (A.2.18) is given by: $k^{\text{FS}_\varepsilon} := \left\lceil \frac{4p}{\lambda_{\rm pmin}(\mathbf{X}^T\mathbf{X})} \left(\frac{1}{\tau^2}\right) \right\rceil - 1$ using the learning rate $\varepsilon = \frac{\|\mathbf{X}\hat{\beta}_{\rm LS}\|_2}{\sqrt{k^{\text{FS}_\varepsilon}+1}}$. Also, it follows from part *(v)* of Theorem 3.1 that the resulting shrinkage bound will satisfy:

$$\text{Sbound}^{\text{FS}_\varepsilon} \le \varepsilon \cdot k^{\text{FS}_\varepsilon} \approx \|\mathbf{X}\hat{\beta}_{\rm LS}\|_2 \cdot \sqrt{k^{\text{FS}_\varepsilon}} \ .$$

Observe that $k^{\text{LS-Boost}(\varepsilon)} < k^{\text{FS}_\varepsilon}$, whereby LS-Boost$(\varepsilon)$ is able to achieve (A.2.18) in fewer iterations than FS_ε. Indeed, if we let η denote the ratio $k^{\text{LS-Boost}(\varepsilon)}/k^{\text{FS}_\varepsilon}$, then it holds that

$$\eta := \frac{k^{\text{LS-Boost}(\varepsilon)}}{k^{\text{FS}_\varepsilon}} \approx \frac{\ln\left(\frac{1}{\tau^2}\right)}{\frac{1}{\tau^2}} \leq \frac{1}{e} \ < \ 0.368 \ . \tag{A.2.19}$$

The left panel of Figure A.2 shows the value of η as a function of τ. For small values of the tolerance parameter τ we see that η is itself close to zero, which means that LS-Boost(ε) will need significantly fewer iterations than FS_ε to achieve the condition (A.2.18).

We can also examine the ℓ_1-shrinkage bounds similarly. If we let ϑ denote the ratio of

SBound$^{\text{LS-Boost}(\varepsilon)}$ to SBound$^{\text{FS}_\varepsilon}$, then it holds that

(A.2.20)

$$\vartheta := \frac{\text{SBound}^{\text{LS-Boost}(\varepsilon)}}{\text{SBound}^{\text{FS}_\varepsilon}} = \sqrt{\frac{k^{\text{LS-Boost}(\varepsilon)}}{k^{\text{FS}_\varepsilon}}} = \frac{\sqrt{\ln\left(\frac{1}{\tau^2}\right)}}{\frac{1}{\tau}} \leq \frac{1}{\sqrt{e}} \ < \ 0.607 \ .$$

This means that if both bounds are relatively tight, then the ℓ_1-shrinkage of the final model produced by LS-Boost(ε) is smaller than that of the final model produced by FS_ε, by at least a factor of 0.607. The right panel of Figure A.2 shows the value of ϑ as a function of τ. For small values of the relative predication error constant τ we see that ϑ is itself close to zero.

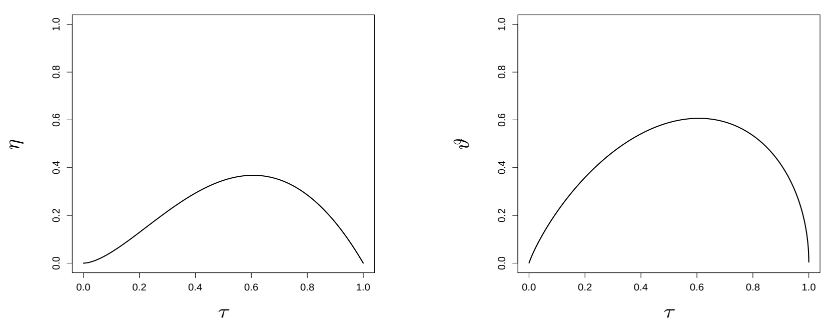

Fig A.2. *Plot showing the value of the ratio η of iterations of* LS-Boost(ε) *to FS_ε (equation (A.2.19)) versus the target relative prediction error τ [left panel], and the ratio ϑ of shrinkage bounds of* LS-Boost(ε) *to FS_ε (equation (A.2.20)) versus the target relative prediction error τ [right panel].*

We summarize the above analysis in the following remark.

REMARK A.2.1. **(Comparison of efficiency of LS-Boost(ε) and FS$_\varepsilon$)** *Suppose that the primary goal is to achieve a τ-relative prediction error as defined in (A.2.18), and that* LS-BOOST(ε) *and FS_ε are run with appropriately determined learning rates for each algorithm. Then the ratio of required number of iterations of these methods to achieve (A.2.18) satisfies*

$$\eta := \frac{k^{\text{LS-BOOST}(\varepsilon)}}{k^{FS_\varepsilon}} < 0.368 \ .$$

Also, the ratio of the shrinkage bounds from running these numbers of iterations satisfies

$$\vartheta := \frac{\text{SBOUND}^{\text{LS-BOOST}(\varepsilon)}}{\text{SBOUND}^{FS_\varepsilon}} < 0.607 \ ,$$

where all of the analysis derives from the bounds in Theorems 2.1 and 3.1.

We caution the reader that the analysis leading to Remark A.2.1 is premised on the singular goal of achieving (A.2.18) in as few iterations as possible. As mentioned previously, the models produced in the interior of the boosting profile are more statistically interesting than those produced at the end. Thus for both algorithms it may be beneficial, and may lessen the risk of over-fitting, to trace out a smoother profile by selecting the learning rate ε to be smaller than the prescribed values in this subsection ($\varepsilon = 1$ for LS-BOOST(ε) and $\varepsilon = \frac{\|\mathbf{X}\hat{\beta}_{\text{LS}}\|_2}{\sqrt{k^{\text{FS}_\varepsilon}+1}}$ for FS$_\varepsilon$). Indeed, considering just LS-BOOST(ε) for simplicity, if our goal is to produce a τ-relative prediction error according to (A.2.18) with the largest value of the ℓ_1 norm of the coefficients, then Figure 3 suggests that this should be accomplished by selecting ε as small as possible (essentially very slightly larger than 0).

A.2.8. *Tightness of bounds.* Herein we show the results of some numerical experiments to assess the tightness of the bounds provided in Theorem 2.1. We considered two different classes of examples, Type 1 and Type 2, as described below.

Type 1: In this example, we set $\mathbf{X}$ such that $\mathbf{X}^T\mathbf{X} = \mathbf{\Sigma}_{p\times p}$ with the (i,j)-entry of $\mathbf{\Sigma}$ being given by: $\sigma_{ij} = \rho$ for $i \neq j$ and $\sigma_{ii} = 1$ otherwise. The response was generated as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^0$, without any noise. Here $\beta_1^0 = 3$ and $\beta_i^0 = 0$ otherwise.

Type 2: This example is similar to Type 1, with the exception of $\boldsymbol{\beta}^0$. Here $\beta_i^0 = 3$ for all $i = 1,\ldots,p$.

LS-BOOST(ε) was allowed to run for different values of ε for different datasets generated as per Types 1 and 2, for different values of ρ and p. Let $\hat{\gamma}_k := (L_n(\hat{\beta}^{k+1}) - L_n^*)/(L_n(\hat{\beta}^k) - L_n^*)$ denote the estimated rate of change of the loss function optimality gap at iteration k. If the upper bounds implied by Theorem 2.1 are tight, the quantity $\hat{\gamma}_k \equiv \gamma$ for all values of k. We considered the evolution of LS-Boost(ε) for a fixed ε for up to $20,000$ iterations. We set $q_{10\%}, q_{50\%}$ and $q_{90\%}$ as the 10th, 50th and 90th quantiles of the observed values $\hat{\gamma}_k$ for $k \geq 1$, and we show the values of their deviations from γ in Table A.1. Table A.1 shows results for $\varepsilon \in \{0.1, 1\}$ — we see that the theoretical values and the observed values are quite close. The theoretical and empirical values are in good agreement for all values of ε. (Note that for smaller values of ε, the value of γ becomes progressively closer to one.)

We also study the evolution of the ℓ_1-shrinkage of the coefficients as per Theorem 2.1, part (v). We considered the following modified upper bound:

$$\|\hat{\beta}^k\|_1 \leq \min\left\{\sqrt{k}\sqrt{\tfrac{\varepsilon}{2-\varepsilon}}\sqrt{\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2^2} \ \ , \ \frac{\varepsilon\|\mathbf{X}\hat{\beta}_{LS}\|_2}{1-\sqrt{\gamma}}\left(1-\gamma^{k/2}\right)\right\},$$

by dropping the term $(-\|\mathbf{X}\hat{\beta}_{\mathrm{LS}} - \mathbf{X}\hat{\beta}^k\|_2^2)$ from the first component. Figure A.3 shows the results for LS-Boost(ε) for a fixed $\varepsilon \in \{0.001, 0.01\}$. The figure shows that the bounds on the ℓ_1-shrinkage of the coefficients are tight for early iterations k, but are not tight for large values of k. We observe empirically that the theoretical and empirical bounds are in good agreement for small/moderate values of ε – but the agreement deteriorates with larger values of ε that are close to one.

A.3. Additional Details for Section 4.

A.3.1. *Duality Between Regularized Correlation Minimization and the* Lasso. In this section, we precisely state the duality relationship between the RCM problem (4.2) and the Lasso optimization problem (1.1). We first prove the following property of the least squares loss function that will be useful in our analysis.

Proposition A.3.1. *The least squares loss function $L_n(\cdot)$ has the following max representation:*

$$L_n(\beta) = \max_{\tilde{r}\in P_{res}} \left\{-\tilde{r}^T(\tfrac{1}{n}\mathbf{X})\beta - \tfrac{1}{2n}\|\tilde{r}-\mathbf{y}\|_2^2 + \tfrac{1}{2n}\|\mathbf{y}\|_2^2\right\} \ , \tag{A.3.1}$$

where $P_{\mathrm{res}} := \{r \in \mathbb{R}^n : r = \mathbf{y} - \mathbf{X}\beta$ for some $\beta \in \mathbb{R}^p\}$. Moreover, the unique optimal solution (as a function of β) to the subproblem in (A.3.1) is $\bar{r} := \mathbf{y} - \mathbf{X}\beta$.

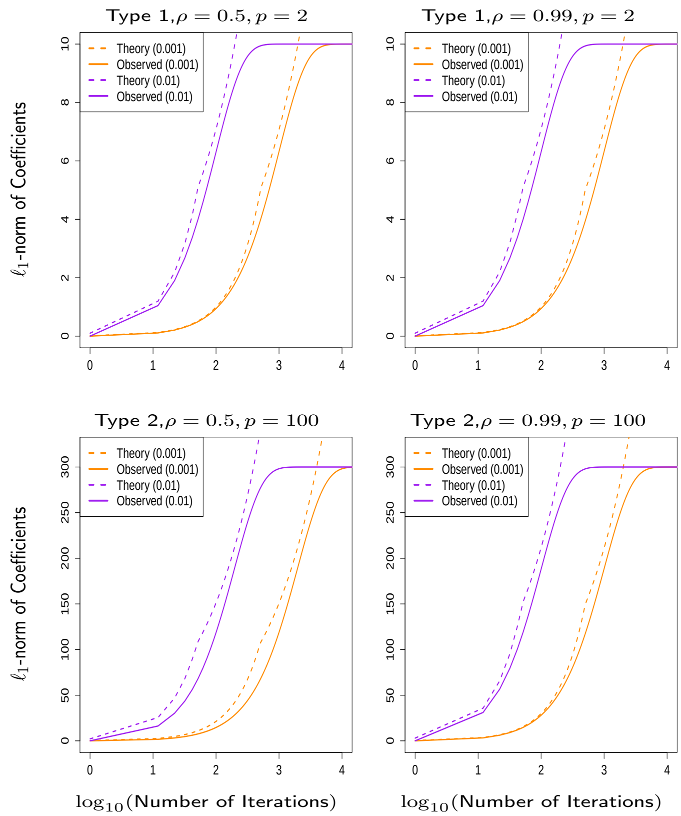

FIG A.3. *Figure showing observed versus theoretical bounds (as implied by Theorem 2.1) of the ℓ_1-norm of the regression coefficients obtained from* LS-BOOST(ε) *, for different values of ε and different datasets. The horizontal axis is the number of* LS-BOOST(ε) *iterations (in the log-scale). The top panels show results for data of Type 1, and the bottom panels show results for data of Type 2.*

PROOF. For any $\beta \in \mathbb{R}^p$, it is easy to verify through optimality conditions (setting the gradient with respect to $\tilde{r}$ equal to 0) that $\bar{r}$ solves the subproblem in (A.3.1), i.e.,

$$\bar{r} = \arg\max_{\tilde{r} \in P_{\text{res}}} \left\{ -\tilde{r}^T (\tfrac{1}{n}\mathbf{X})\beta - \tfrac{1}{2n}\|\tilde{r} - \mathbf{y}\|_2^2 + \tfrac{1}{2n}\|\mathbf{y}\|_2^2 \right\} \ .$$

Thus, we have

$$\begin{aligned}\max_{\tilde{r} \in P_{\text{res}}} \left\{ -\tilde{r}^T (\tfrac{1}{n}\mathbf{X})\beta - \tfrac{1}{2n}\|\tilde{r} - \mathbf{y}\|_2^2 + \tfrac{1}{2n}\|\mathbf{y}\|_2^2 \right\} &= \tfrac{1}{n}\left(\tfrac{1}{2}\|\mathbf{y}\|_2^2 - \mathbf{y}^T\mathbf{X}\beta + \tfrac{1}{2}\|\mathbf{X}\beta\|_2^2 \right) \\ &= \tfrac{1}{2n}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 \ .\end{aligned}$$

$\square$

The next result demonstrates that RCM (4.2) has a direct interpretation as a (scaled) dual of the LASSO problem (1.1). Moreover, in part *(iii)* of the result below, we present a bound on the optimality gap for the LASSO problem in terms of a quantity that is closely related to the objective function of RCM.

PROPOSITION A.3.2. **(Duality Equivalence of Lasso and RCM$_\delta$, and Optimality Bounds)** *The* LASSO *problem* (1.1) *and the regularized correlation minimization problem* RCM$_\delta$ (4.2) *are dual optimization problems modulo the scaling factor* $\frac{n}{\delta}$. *In particular:*

(i) *(Weak Duality) If β is feasible for the* LASSO *problem* (1.1), *and if $\tilde{r}$ is feasible for the regularized correlation minimization problem* RCM$_\delta$ (4.2), *then*
$$L_n(\beta) + \tfrac{\delta}{n} f_\delta(\tilde{r}) \ \geq \ \tfrac{1}{2n}\|\mathbf{y}\|_2^2 \ .$$
(ii) *(Strong Duality) It holds that:*
$$L_{n,\delta}^* + \tfrac{\delta}{n} f_\delta^* = \tfrac{1}{2n}\|\mathbf{y}\|_2^2 \ .$$
Moreover, for any given parameter value $\delta \geq 0$, there is a unique vector of residuals $\hat{r}_\delta^*$ associated with every LASSO *solution $\hat{\beta}_\delta^*$, i.e., $\hat{r}_\delta^* = \mathbf{y} - \mathbf{X}\hat{\beta}_\delta^*$, and $\hat{r}_\delta^*$ is the unique optimal solution to the RCM$_\delta$ problem* (4.2).

(iii) *(Optimality Condition for* LASSO*) If β is feasible for the* LASSO *problem* (1.1) *and $r = \mathbf{y} - \mathbf{X}\beta$, then*
$$\omega_\delta(\beta) := \|\mathbf{X}^T r\|_\infty - \frac{r^T \mathbf{X}\beta}{\delta} \ \geq 0 \ , \tag{A.3.2}$$
and
$$L_n(\beta) - L_{n,\delta}^* \ \leq \ \tfrac{\delta}{n} \cdot \omega_\delta(\beta) \ .$$
Hence, if $\omega_\delta(\beta) = 0$, then β is an optimal solution of the LASSO *problem* (1.1). □

PROOF. Let us first construct the problem RCM$_\delta$ using basic constructs of minmax duality. As demonstrated in Proposition A.3.1, the least-squares loss function $L_n(\cdot)$ has the following max representation:
$$L_n(\beta) = \max_{\tilde{r} \in P_{\text{res}}} \left\{ -\tilde{r}^T(\tfrac{1}{n}\mathbf{X})\beta - \tfrac{1}{2n}\|\tilde{r} - \mathbf{y}\|_2^2 + \tfrac{1}{2n}\|\mathbf{y}\|_2^2 \right\} \ .$$
Therefore the LASSO problem (1.1) can be written as
$$\min_{\beta \in B_\delta} \ \max_{\tilde{r} \in P_{\text{res}}} \left\{ -\tilde{r}^T(\tfrac{1}{n}\mathbf{X})\beta - \tfrac{1}{2n}\|\tilde{r} - \mathbf{y}\|_2^2 + \tfrac{1}{2n}\|\mathbf{y}\|_2^2 \right\}$$

where $B_\delta := \{\beta \in \mathbb{R}^p : \|\beta\|_1 \leq \delta\}$. We construct a dual of the above problem by interchanging the min and max operators above, yielding the following dual optimization problem:

$$\max_{\tilde{r}\in P_{\text{res}}} \min_{\beta\in B_\delta} \left\{-\tilde{r}^T(\tfrac{1}{n}\mathbf{X})\beta - \tfrac{1}{2n}\|\tilde{r}-\mathbf{y}\|_2^2 + \tfrac{1}{2n}\|\mathbf{y}\|_2^2\right\} \ .$$

After negating, and dropping the constant term $\frac{1}{2n}\|\mathbf{y}\|_2^2$, the above dual problem is equivalent to:

$$\min_{\tilde{r}\in P_{\text{res}}} \max_{\beta\in B_\delta} \left\{\tilde{r}^T(\tfrac{1}{n}\mathbf{X})\beta\right\} + \tfrac{1}{2n}\|\tilde{r}-\mathbf{y}\|_2^2 \ . \tag{A.3.3}$$

Now notice that

$$\max_{\beta\in B_\delta} \left\{\tilde{r}^T(\tfrac{1}{n}\mathbf{X})\beta\right\} = \tfrac{\delta}{n}\left(\max_{j\in\{1,\ldots,p\}} |\tilde{r}^T\mathbf{X}_j|\right) = \tfrac{\delta}{n}\|\mathbf{X}^T\tilde{r}\|_\infty \ , \tag{A.3.4}$$

from which it follows after scaling by $\frac{n}{\delta}$ that (A.3.3) is equivalent to (4.2).

Let us now prove item (i). Let β be feasible for the Lasso problem (1.1) and $\tilde{r}$ be feasible for the regularized correlation minimization problem RCM$_\delta$ (4.2), and let $r = \mathbf{y} - \mathbf{X}\beta$ and let $\tilde{\beta}$ be such that $\tilde{r} = \mathbf{y} - \mathbf{X}\tilde{\beta}$. Then direct arithmetic manipulation yields the following equality:

$$L_n(\beta) + \tfrac{\delta}{n} f_\delta(\tilde{r}) = \tfrac{1}{2n}\|\mathbf{y}\|_2^2 + \tfrac{1}{2n}\|r-\tilde{r}\|_2^2 + \tfrac{\delta}{n}\left(\|\mathbf{X}^T\tilde{r}\|_\infty - \frac{\tilde{r}^T\mathbf{X}\beta}{\delta}\right) \ , \tag{A.3.5}$$

from which the result follows since $\|r-\tilde{r}\|_2^2 \geq 0$ and $\tilde{r}^T\mathbf{X}\beta \leq \|\mathbf{X}^T\tilde{r}\|_\infty\|\beta\|_1 \leq \delta\|\mathbf{X}^T\tilde{r}\|_\infty$ which implies that the last term above is also nonnegative.

To prove item (ii), notice that both the Lasso and RCM$_\delta$ can be re-cast as optimization problems with a convex quadratic objective function and with linear inequality constraints. That being the case, the classical strong duality results for linearly-constrained convex quadratic optimization apply, see [3] for example.

We now prove (iii). Since β is feasible for the Lasso problem, it follows from the Holder inequality that $r^T\mathbf{X}\beta \leq \|\mathbf{X}^T r\|_\infty\|\beta\|_1 \leq \delta\|\mathbf{X}^T r\|_\infty$, from which it then follows that $\omega_\delta(\beta) \geq 0$. Invoking (A.3.5) with $\tilde{r} \leftarrow r = \mathbf{y} - \mathbf{X}\beta$ yields:

$$L_n(\beta) + \tfrac{\delta}{n} f_\delta(r) = \tfrac{1}{2n}\|\mathbf{y}\|_2^2 + \tfrac{\delta}{n}\cdot\omega_\delta(\beta) \ .$$

Combining the above with strong duality (ii) yields:

$$L_n(\beta) + \tfrac{\delta}{n} f_\delta(r) = L^*_{n,\delta} + \tfrac{\delta}{n} f^*_\delta + \tfrac{\delta}{n}\cdot\omega_\delta(\beta) \ .$$

After rearranging we have:

$$L_n(\beta) - L^*_{n,\delta} \leq \tfrac{\delta}{n} f^*_\delta - \tfrac{\delta}{n} f_\delta(r) + \tfrac{\delta}{n}\cdot\omega_\delta(\beta) \leq \tfrac{\delta}{n}\cdot\omega_\delta(\beta) \ ,$$

where the last inequality follows since $f^*_\delta \leq f_\delta(r)$. □

A.3.2. *Proof of Proposition 4.1.* Recall the update formula for the residuals in R-FS$_{\varepsilon,\delta}$:

$$\hat{r}^{k+1} \leftarrow \hat{r}^k - \varepsilon \left[\operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k} + \tfrac{1}{\delta}(\hat{r}^k - \mathbf{y}) \right] \ . \tag{A.3.6}$$

We first show that $g^k := \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k} + \frac{1}{\delta}(\hat{r}^k - \mathbf{y})$ is a subgradient of $f_\delta(\cdot)$ at $\hat{r}^k$. Recalling the proof of Proposition 3.2, we have that $\operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k}$ is a subgradient of $f(r) := \|\mathbf{X}^T r\|_\infty$ at $\hat{r}^k$ since $j_k \in \arg\max_{j \in \{1,\ldots,p\}} |(\hat{r}^k)^T \mathbf{X}_j|$. Therefore, since $f_\delta(r) = f(r) + \frac{1}{2\delta}\|r - \mathbf{y}\|_2^2$, it follows from the additive property of subgradients (and gradients) that $g^k = \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k} + \frac{1}{\delta}(\hat{r}^k - \mathbf{y})$ is a subgradient of $f_\delta(r)$ at $r = \hat{r}^k$. Therefore the update (A.3.6) is of the form $\hat{r}^{k+1} = \hat{r}^k - \varepsilon g^k$ where $g^k \in \partial f_\delta(\hat{r}^k)$. Finally note that $\hat{r}^k - \varepsilon g^k = \hat{r}^{k+1} = \mathbf{y} - \mathbf{X}\beta^{k+1} \in P_{\text{res}}$, hence $\Pi_{P_{\text{res}}}(\hat{r}^k - \varepsilon g^k) = \hat{r}^k - \varepsilon g^k$, i.e., the projection step is superfluous here. Therefore $\hat{r}^{k+1} = \Pi_{P_{\text{res}}}(\hat{r}^k - \varepsilon g^k)$, which shows that (A.3.6) is precisely the update for the subgradient descent method with step-size $\alpha_k := \varepsilon$. □

A.3.3. *Proof of Theorem 4.1.* Let us first use induction to demonstrate that the following inequality holds:

$$\|\hat{\beta}^k\|_1 \le \varepsilon \sum_{j=0}^{k-1} \left(1 - \tfrac{\varepsilon}{\delta}\right)^j \quad \text{for all } k \ge 0 \ . \tag{A.3.7}$$

Clearly, (A.3.7) holds for $k = 0$ since $\hat{\beta}^0 = 0$. Assuming that (A.3.7) holds for k, then the update for $\hat{\beta}^{k+1}$ in step (3.) of algorithm R-FS$_{\varepsilon,\delta}$ can be written as $\hat{\beta}^{k+1} = (1 - \frac{\varepsilon}{\delta})\hat{\beta}^k + \varepsilon \cdot \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) e_{j_k}$, from which it holds that

$$\begin{aligned}
\|\hat{\beta}^{k+1}\|_1 &= \|(1 - \tfrac{\varepsilon}{\delta})\hat{\beta}^k + \varepsilon \cdot \operatorname{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) e_{j_k}\|_1 \\
&\le (1 - \tfrac{\varepsilon}{\delta})\|\hat{\beta}^k\|_1 + \varepsilon \|e_{j_k}\|_1 \\
&\le (1 - \tfrac{\varepsilon}{\delta})\varepsilon \sum_{j=0}^{k-1} \left(1 - \tfrac{\varepsilon}{\delta}\right)^j \ + \ \varepsilon \\
&= \varepsilon \sum_{j=0}^{k} \left(1 - \tfrac{\varepsilon}{\delta}\right)^j \ ,
\end{aligned}$$

which completes the induction. Now note that (A.3.7) is a geometric series and we have:

$$\text{(A.3.8)}\quad \|\hat\beta^k\|_1 \le \ \varepsilon\sum_{j=0}^{k-1}\left(1-\tfrac{\varepsilon}{\delta}\right)^j = \ \delta\left[1-\left(1-\tfrac{\varepsilon}{\delta}\right)^k\right] \le \ \delta \quad \text{for all } k\ge 0\ .$$

Recall that we developed the algorithm R-FS$_{\varepsilon,\delta}$ in such a way that it corresponds exactly to an instantiation of the subgradient descent method applied to the RCM problem (4.2). Indeed, the update rule for the residuals given in Step (3.) of R-FS$_{\varepsilon,\delta}$ is: $\hat r^{k+1}\leftarrow \hat r^k-\varepsilon g^k$ where $g^k=\left[\mathrm{sgn}((\hat r^k)^T\mathbf{X}_{j_k})\mathbf{X}_{j_k}+\frac{1}{\delta}(\hat r^k-\mathbf{y})\right]$. We therefore can apply Proposition A.2.4, and more specifically the inequality (A.2.13). In order to do so we need to translate the terms of Proposition A.2.4 to our setting: here the variables x are now the residuals r, the iterates x^i are now the iterates $\hat r^i$, etc. The step-sizes of algorithm R-FS$_{\varepsilon,\delta}$ are fixed at ε, so we have $\alpha_i=\varepsilon$ for all $i\ge 0$. Setting the value of x in Proposition A.2.4 to be least-squares residual value, namely $x=\hat r_{LS}$, the left side of (A.2.13) is therefore:

(A.3.9)

$$\begin{aligned}
\tfrac{1}{k+1}\textstyle\sum_{i=0}^k (g^i)^T(x^i-x) \quad &= \quad \tfrac{1}{k+1}\textstyle\sum_{i=0}^k\left(\mathbf{X}\left[\mathrm{sgn}((\hat r^i)^T\mathbf{X}_{j_i})e_{j_i}-\tfrac{1}{\delta}\hat\beta^i\right]\right)^T(\hat r^i-\hat r_{LS})\\
&= \quad \tfrac{1}{k+1}\textstyle\sum_{i=0}^k\left(\mathrm{sgn}((\hat r^i)^T\mathbf{X}_{j_i})\mathbf{X}_{j_i}-\tfrac{1}{\delta}(\mathbf{X}\hat\beta^i)\right)^T\hat r^i\\
&= \quad \tfrac{1}{k+1}\textstyle\sum_{i=0}^k\left[\|\mathbf{X}^T\hat r^i\|_\infty-\tfrac{1}{\delta}(\hat r^i)^T\mathbf{X}\hat\beta^i\right]\\
&= \quad \tfrac{1}{k+1}\textstyle\sum_{i=0}^k\omega_\delta(\hat\beta^i)\ ,
\end{aligned}$$

where the second equality uses the fact that $\mathbf{X}^T\hat r_{LS}=0$ from (2.3) and the fourth equality uses the definition of $\omega_\delta(\beta)$ from (A.3.2).

Let us now evaluate the right side of (A.2.13). We have $\|x^0-x\|_2=\|\hat r^0-\hat r_{LS}\|_2=\|\mathbf{y}-(\mathbf{y}-\mathbf{X}\hat\beta_{\mathrm{LS}})\|_2=\|\mathbf{X}\hat\beta_{\mathrm{LS}}\|_2$. Also, it holds that

$$\begin{aligned}
\|g^i\|_2 =&\|\mathrm{sgn}((\hat r^i)^T\mathbf{X}_{j_i})\mathbf{X}_{j_i}-\tfrac{1}{\delta}(\mathbf{X}\hat\beta^i)\|_2\\
\le&\|\mathbf{X}_{j_i}\|_2+\|\mathbf{X}(\tfrac{\hat\beta^i}{\delta})\|_2\\
\le&1+\tfrac{1}{\delta}\|\mathbf{X}\|_{1,2}\|\hat\beta^i\|_1\\
\le&1+\|\mathbf{X}\|_{1,2}\\
\le&2\ ,
\end{aligned}$$

where the third inequality follows since $\|\hat\beta^i\|_1\le\delta$ from (A.3.8) and the second and fourth inequalities follow from the assumption that the columns of $\mathbf{X}$ have been normalized to have unit ℓ_2 norm. Therefore $G = 2$ is a uniform bound on $\|g^i\|_2$. Combining the above, inequality (A.2.13) implies that after running R-FS$_{\varepsilon,\delta}$ for k iterations, it holds that:

(A.3.10)

$$\min_{i\in\{0,\ldots,k\}} \omega_\delta(\hat\beta^i) \ \le\ \frac{1}{k+1}\sum_{i=0}^{k}\omega_\delta(\hat\beta^i) \ \le\ \frac{\|\mathbf{X}\hat\beta_{\text{LS}}\|_2^2}{2(k+1)\varepsilon} + \frac{2^2\varepsilon}{2} \ =\ \frac{\|\mathbf{X}\hat\beta_{\text{LS}}\|_2^2}{2\varepsilon(k+1)} + 2\varepsilon \ ,$$

where the first inequality is elementary arithmetic and the second inequality is the application of (A.2.13). Now let i be the index obtaining the minimum in the left-most side of the above. Then it follows from part (iii) of Proposition A.3.2 that

$$L_n(\hat\beta^i) - L^*_{n,\delta} \ \le\ \tfrac{\delta}{n}\cdot\omega_\delta(\hat\beta^i) \le \frac{\delta\|\mathbf{X}\hat\beta_{\text{LS}}\|_2^2}{2n\varepsilon(k+1)} + \frac{2\delta\varepsilon}{n} \ , \tag{A.3.11}$$

which proves item (i) of the theorem.

To prove item (ii), note first that if $\hat\beta^*_\delta$ is a solution of the Lasso problem (1.1), then it holds that $\|\hat\beta^*_\delta\|_1 \le \delta$ (feasibility) and $\omega_\delta(\hat\beta^*_\delta) = 0$ (optimality). This latter condition follows easily from the optimality conditions of linearly constrained convex quadratic problems, see [3] for example. Setting $\hat r^*_\delta = \mathbf{y} - \mathbf{X}\hat\beta^*_\delta$, the following holds true:

$$\begin{aligned}
\|\mathbf{X}\hat\beta^i - \mathbf{X}\hat\beta^*_\delta\|_2^2 \ &=\ 2n\left(L_n(\hat\beta^i) - L_n(\hat\beta^*_\delta) + (\hat r^*_\delta)^T\mathbf{X}(\hat\beta^i - \hat\beta^*_\delta)\right)\\
&=\ 2n\left(L_n(\hat\beta^i) - L^*_{n,\delta} - \delta\|\mathbf{X}^T\hat r^*_\delta\|_\infty + (\hat r^*_\delta)^T\mathbf{X}\hat\beta^i\right)\\
&\le\ 2n\left(L_n(\hat\beta^i) - L^*_{n,\delta} - \delta\|\mathbf{X}^T\hat r^*_\delta\|_\infty + \|\mathbf{X}^T\hat r^*_\delta\|_\infty\|\hat\beta^i\|_1\right)\\
&\le\ 2n\left(L_n(\hat\beta^i) - L^*_{n,\delta} - \delta\|\mathbf{X}^T\hat r^*_\delta\|_\infty + \delta\|\mathbf{X}^T\hat r^*_\delta\|_\infty\right)\\
&=\ 2n\left(L_n(\hat\beta^i) - L^*_{n,\delta}\right)\\
&\le\ \tfrac{\delta\|\mathbf{X}\hat\beta_{\text{LS}}\|_2^2}{\varepsilon(k+1)} + 4\delta\varepsilon \ ,
\end{aligned}$$

where the first equality is from direct arithmetic substitution, the second equality uses the fact that $\omega_\delta(\hat\beta^*_\delta) = 0$ whereby $(\hat r^*_\delta)^T\mathbf{X}\hat\beta^*_\delta = \delta\|\mathbf{X}^T\hat r^*_\delta\|_\infty$, the first inequality follows by applying Holder's inequality to the last term of the second equality, and the final inequality is an application of (A.3.11). Item (ii) then follows by taking square roots of the above.

Item (iii) is essentially just (A.3.8). Indeed, since $i \leq k$ we have:

$$\|\hat{\beta}^i\|_1 \leq \ \varepsilon \sum_{j=0}^{i-1} \left(1 - \tfrac{\varepsilon}{\delta}\right)^j \leq \ \varepsilon \sum_{j=0}^{k-1} \left(1 - \tfrac{\varepsilon}{\delta}\right)^j = \ \delta \left[1 - \left(1 - \tfrac{\varepsilon}{\delta}\right)^k\right] \leq \ \delta \ .$$

(Note that we emphasize the dependence on k rather than i in the above since we have direct control over the number of boosting iterations k.) Item (iv) of the theorem is just a restatement of the sparsity property of R-FS$_{\varepsilon,\delta}$. □

A.3.4. *Regularized Boosting: Related Work and Context.* As we have already seen, the FS$_\varepsilon$ algorithm leads to models that have curious similarities with the Lasso coefficient profile, but in general the profiles are different. Sufficient conditions under which the coefficient profiles of FS$_\varepsilon$ (for $\varepsilon \approx 0$) and Lasso are equivalent have been explored in [34]. A related research question is whether there are structurally similar algorithmic variants of FS$_\varepsilon$ that lead to Lasso solutions for arbitrary datasets? In this vein [53] propose BLasso, a corrective version of the forward stagewise algorithm. BLasso, in addition to taking incremental forward steps (as in FS$_\varepsilon$), also takes backward steps, the result of which is that the algorithm approximates the Lasso coefficient profile under certain assumptions on the data. The authors observe that BLasso often leads to models that are sparser and have better predictive accuracy than those produced by FS$_\varepsilon$.

In [9], the authors point out that models delivered by boosting methods need not be adequately sparse, and they highlight the importance of obtaining models that have more sparsity, better prediction accuracy, and better variable selection properties. They propose a sparse variant of $L2$-Boost (see also Section 1) which considers a regularized version of the squared error loss, penalizing the approximate degrees of freedom of the model.

In [26], the authors also point out that boosting algorithms often lead to a large collection of nonzero coefficients. They suggest reducing the complexity of the model by some form of "post-processing" technique—one such proposal is to apply a Lasso regularization on the selected set of coefficients.

A parallel line of work in machine learning [13] explores the scope of boosting-like algorithms on ℓ_1-regularized versions of different loss functions arising mainly in the context of classification problems. The proposal of [13], when adapted to the least squares regression problem with ℓ_1-regularization penalty, leads to the penalized version of the Lasso problem:

$$\min_{\beta} \ \ \tfrac{1}{2n}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1 \ , \tag{A.3.12}$$

for which the authors [13] employ greedy coordinate descent methods. Like the boosting algorithms considered herein, at each iteration the algorithm studied by [13] selects a certain coefficient β_{j_k} to update, leaving all other coefficients β_i unchanged. The amount with which to update the coefficient β_{j_k} is determined by fully optimizing the loss function (A.3.12) with respect to β_{j_k}, again holding all other coefficients constant (note that one recovers LS-Boost(1) if $\lambda = 0$). This way of updating β_{j_k} leads to a simple soft-thresholding operation [12] and is *structurally* different from forward stagewise update rules. In contrast, the boosting algorithm R-FS$_{\varepsilon,\delta}$ that we propose here is based on subgradient descent on the dual of the Lasso problem (1.1), i.e., problem (4.2).

A.3.5. *Connecting R-FS$_{\varepsilon,\delta}$ to the Frank-Wolfe method.* Although we developed and analyzed R-FS$_{\varepsilon,\delta}$ from the perspective of subgradient descent, one can also interpret R-FS$_{\varepsilon,\delta}$ as the Frank-Wolfe algorithm in convex optimization [15, 35, 18] applied to the Lasso (1.1). This secondary interpretation can be derived directly from the structure of the updates in R-FS$_{\varepsilon,\delta}$ or as a special case of a more general primal-dual equivalence between subgradient descent and Frank-Wolfe developed in [1]. Other authors have commented on the qualitative similarities between boosting algorithms and the Frank-Wolfe method, see for instance [10] and [35].

We note that there are subtle differences between R-FS$_{\varepsilon,\delta}$ and a standard Frank-Wolfe method — the difference lies in the choice of the step-size sequence. Standard (popular) Frank Wolfe methods [35] (see also, references therein) use a step-size of $2/(k+2)$ (where k is the iteration index) or a step-size based on a line-search. The step-size in R-FS$_{\varepsilon,\delta}$, however, is *fixed* across iterations — this is done to make R-FS$_{\varepsilon,\delta}$ coincide with a boosting algorithm with fixed learning rate (and in particular, to make it coincide with FS$_\varepsilon$ when $\delta = \infty$). Analysis of this particular step-size choice in R-FS$_{\varepsilon,\delta}$ requires a different technique than that used in a more traditional analysis of Frank-Wolfe such as that presented in [35]. In particular, the analysis we present herein is based on tools from subgradient optimization, which we utilize since subgradient descent provides a natural unifying framework for a general class of boosting algorithms (including FS$_\varepsilon$ and R-FS$_{\varepsilon,\delta}$) via a single algorithm applied to a parametric class of objective functions. We note that [18] derive computational guarantees for Frank-Wolfe with a general step-size sequence – including a fixed step-size across iterations; however their proof is different than that presented herein.

A.4. Additional Details for Section 5.

A.4.1. *Proof of Theorem 5.1.* We first prove the feasibility of $\hat\beta^k$ for the Lasso problem with parameter $\bar\delta_k$. We do so by induction. The feasibility of $\hat\beta^k$ is obviously true for $k=0$ since $\hat\beta^0 = 0$ and hence $\|\hat\beta^0\|_1 = 0 < \bar\delta_0$. Now suppose it is true for some iteration k, i.e., $\|\hat\beta^k\|_1 \le \bar\delta_k$. Then the update for $\hat\beta^{k+1}$ in step (3.) of algorithm PATH-R-FS$_\varepsilon$ can be written as $\hat\beta^{k+1} = (1-\frac{\varepsilon}{\bar\delta_k})\hat\beta^k + \frac{\varepsilon}{\bar\delta_k}(\bar\delta_k \mathrm{sgn}((\hat r^k)^T \mathbf{X}_{j_k}) e_{j_k})$, from which it follows that

$$\begin{aligned}\|\hat\beta^{k+1}\|_1 &= \|(1-\tfrac{\varepsilon}{\bar\delta_k})\hat\beta^k + \tfrac{\varepsilon}{\bar\delta_k}(\bar\delta_k \mathrm{sgn}((\hat r^k)^T \mathbf{X}_{j_k}) e_{j_k})\|_1\\ &\le (1-\tfrac{\varepsilon}{\bar\delta_k})\|\hat\beta^k\|_1 + \tfrac{\varepsilon}{\bar\delta_k}\|\bar\delta_k e_{j_k}\|_1 \le (1-\tfrac{\varepsilon}{\bar\delta_k})\bar\delta_k + \tfrac{\varepsilon}{\bar\delta_k}\bar\delta_k = \bar\delta_k \le \bar\delta_{k+1}\ ,\end{aligned}$$

which completes the induction.

We now prove the bound on the average training error in part *(i)*. In fact, we will prove something stronger than this bound, namely we will prove:

$$\frac{1}{k+1}\sum_{i=0}^{k}\frac{1}{\bar\delta_i}\left(L_n(\hat\beta^i) - L^*_{n,\bar\delta_i}\right) \ \le\ \frac{\|\mathbf{X}\hat\beta_{\mathrm{LS}}\|_2^2}{2n\varepsilon(k+1)} + \frac{2\varepsilon}{n}\ , \tag{A.4.1}$$

from which average training error bound of part *(i)* follows since $\bar\delta_i \le \bar\delta$ for all i. The update rule for the residuals given in Step (3.) of R-FS$_{\varepsilon,\delta}$ is: $\hat r^{k+1} \leftarrow \hat r^k - \varepsilon g^k$ where $g^k = \left[\mathrm{sgn}((\hat r^k)^T\mathbf{X}_{j_k})\mathbf{X}_{j_k} + \frac{1}{\bar\delta_k}(\hat r^k - \mathbf{y})\right]$. This update rule is precisely in the format of an elementary sequence process, see Appendix A.2.5, and we therefore can apply Proposition A.2.4, and more specifically the inequality (A.2.13). Similar in structure to the proof of Theorem 4.1, we first need to translate the terms of Proposition A.2.4 to our setting: once again the variables x are now the residuals r, the iterates x^i are now the iterates $\hat r^i$, etc. The step-sizes of algorithm PATH-R-FS$_\varepsilon$ are fixed at ε, so we have $\alpha_i = \varepsilon$ for all $i \ge 0$. Setting the value of x in Proposition A.2.4 to be least-squares residual value, namely $x = \hat r_{LS}$, and using the exact same logic as in the equations (A.3.9), one obtains the following result about the left side of (A.2.13):

$$\frac{1}{k+1}\sum_{i=0}^{k}(g^i)^T(x^i - x) = \frac{1}{k+1}\sum_{i=0}^{k}\omega_{\bar\delta_i}(\hat\beta^i)\ .$$

Let us now evaluate the right side of (A.2.13). We have $\|x^0 - x\|_2 = \|\hat r^0 -$

$\hat{r}_{LS}\|_2 = \|\mathbf{y} - (\mathbf{y} - \mathbf{X}\hat{\beta}_{\mathrm{LS}})\|_2 = \|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2$. Also, it holds that

$$\begin{aligned}
\|g^i\|_2 =& \|\mathrm{sgn}((\hat{r}^i)^T\mathbf{X}_{j_i})\mathbf{X}_{j_i} - \tfrac{1}{\bar{\delta}_i}(\mathbf{X}\hat{\beta}^i)\|_2 \\
\leq& \|\mathbf{X}_{j_i}\|_2 + \|\mathbf{X}(\tfrac{\hat{\beta}^i}{\bar{\delta}_i})\|_2 \\
\leq& 1 + \tfrac{1}{\bar{\delta}_i}\|\mathbf{X}\|_{1,2}\|\hat{\beta}^i\|_1 \\
\leq& 1 + \|\mathbf{X}\|_{1,2} \\
\leq& 2
\end{aligned}$$

where the third inequality follows since $\|\hat{\beta}^i\|_1 \leq \bar{\delta}_i$ from the feasibility of $\hat{\beta}^i$ for the LASSO problem with parameter $\bar{\delta}_i$ proven at the outset, and the second and fourth inequalities follow from the assumption that the columns of $\mathbf{X}$ have been normalized to have unit ℓ_2 norm. Therefore $G = 2$ is a uniform bound on $\|g^i\|_2$. Combining the above, inequality (A.2.13) implies that after running PATH-R-FS$_\varepsilon$ for k iterations, it holds that:

$$\text{(A.4.2)} \qquad \frac{1}{k+1}\sum_{i=0}^{k}\omega_{\bar{\delta}_i}(\hat{\beta}^i) \;\leq\; \frac{\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2^2}{2(k+1)\varepsilon} + \frac{2^2\varepsilon}{2} \;=\; \frac{\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2^2}{2\varepsilon(k+1)} + 2\varepsilon \ ,$$

where the inequality is the application of (A.2.13). From Proposition A.3.2 we have $L_n(\hat{\beta}^i) - L^*_{n,\bar{\delta}_i} \;\leq\; \frac{\bar{\delta}_i}{n}\cdot\omega_{\bar{\delta}_i}(\hat{\beta}^i)$, which combines with (A.4.2) to yield:

$$\frac{1}{k+1}\sum_{i=0}^{k}\frac{1}{\bar{\delta}_i}\left(L_n(\hat{\beta}^i) - L^*_{n,\bar{\delta}_i}\right) \;\leq\; \frac{1}{(k+1)}\frac{1}{n}\sum_{i=0}^{k}\omega_{\bar{\delta}_i}(\hat{\beta}^i) \;\leq\; \frac{\|\mathbf{X}\hat{\beta}_{\mathrm{LS}}\|_2^2}{2n\varepsilon(k+1)} + \frac{2\varepsilon}{n} \ .$$

This proves (A.4.1) which then completes the proof of part *(i)* through the bounds $\bar{\delta}_i \leq \bar{\delta}$ for all i.

Part *(ii)* is a restatement of the feasibility of $\hat{\beta}^k$ for the LASSO problem with parameter $\bar{\delta}_k$ which was proved at the outset, and is re-written to be consistent with the format of, and for comparison with, Theorem 4.1. Last of all, part *(iii)* follows since at each iteration at most one new coefficient is introduced at a non-zero level. □

A.5. Additional Details on the Experiments. We describe here some additional details pertaining to the computational results performed in this paper. We first describe in some more detail the real datasets that have been considered in the paper.

Description of datasets considered.
We considered four different publicly available microarray datasets as described below.

Leukemia dataset. This dataset, taken from [11], has binary response with continuous covariates, with 72 samples and approximately 3500 covariates. We further processed the dataset by taking a subsample of $p = 500$ covariates, while retaining all $n = 72$ sample points. We artificially generated the response $\mathbf{y}$ via a linear model with the given covariates $\mathbf{X}$ (as described in Eg-A in Section 6). The true regression coefficient β^{pop} was taken as $\beta_i^{\text{pop}} = 1$ for all $i \leq 10$ and zero otherwise.

Golub dataset. The original dataset was taken from the R package `mpm`, which had 73 samples with approximately 5000 covariates. We reduced this to $p = 500$ covariates (all samples were retained). Responses $\mathbf{y}$ were generated via a linear model with β^{pop} as above.

Khan dataset. This dataset was taken from the dataset webpage `http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/` accompanying the book [33]. The original covariate matrix (`khan.xtest`), which had 73 samples with approximately 5000 covariates, was reduced to $p = 500$ covariates (all samples were retained). Responses $\mathbf{y}$ were generated via a linear model with β^{pop} as above.

Prostate cancer dataset. This dataset appears in [14] and is available from the R package `LARS`. The first column `lcavol` was taken as the response (no artificial response was created here). We generated multiple datasets from this dataset, as follows:

(a) One of the datasets is the original one with $n = 97$ and $p = 8$.
(b) We created another dataset, with $n = 97$ and $p = 44$ by enhancing the covariate space to include second order interactions.
(c) We created another dataset, with $n = 10$ and $p = 44$. We subsampled the dataset from (b), which again was enhanced to include second order interactions.

Note that in all the examples above we standardized $\mathbf{X}$ such that the columns have unit ℓ_2 norm, before running the different algorithms studied herein.

Sensitivity of the Learning Rate in LS-Boost(ε) *and FS$_\varepsilon$.* We performed several experiments running LS-Boost(ε) and FS$_\varepsilon$ on an array of real and synthetic datasets, to explore how the training and test errors change as a function of the number of boosting iterations and the learning rate. Some of the results appear in Figure A.4. The training errors were found to decrease with increasing number of boosting iterations. The rate of decay, however, is very sensitive to the value of ε, with smaller values of ε leading to slower

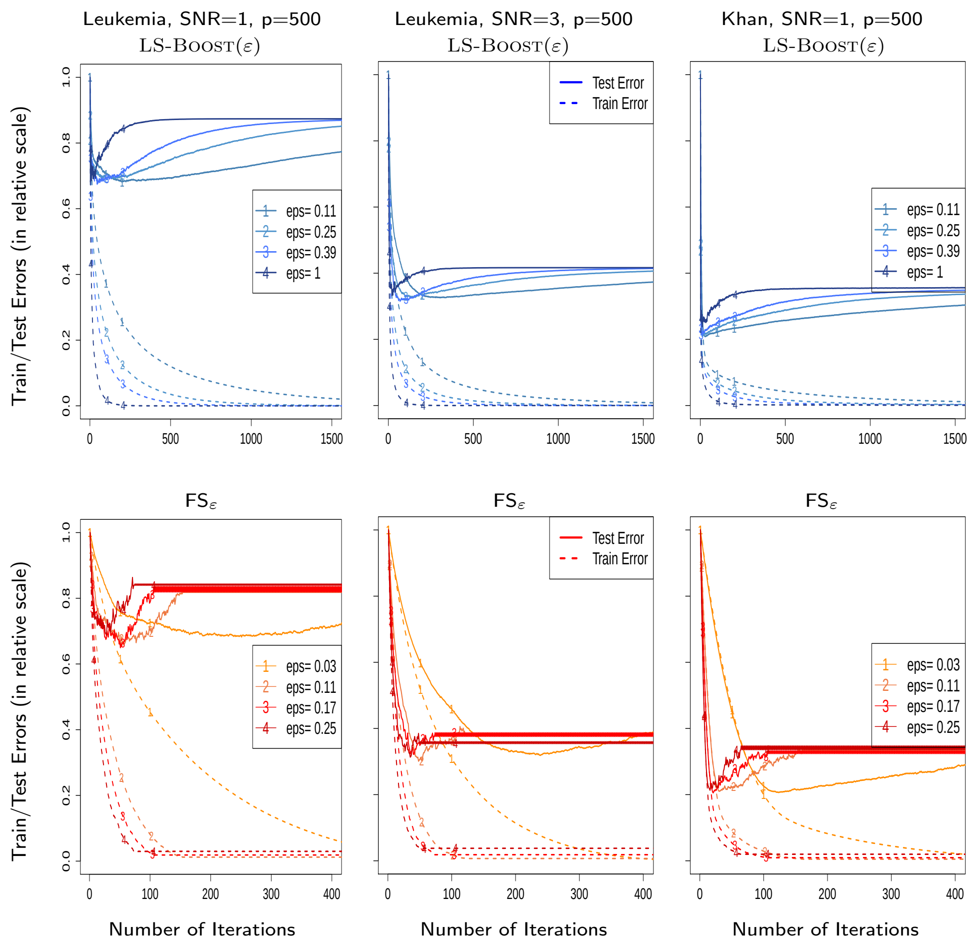

Fig A.4. *Figure showing the training and test errors (in relative scale) as a function of boosting iterations, for both* LS-Boost(ε) *(top panel) and* FS_ε *(bottom panel). As the number of iterations increases, the training error shows a global monotone pattern. The test errors however, initially decrease and then start increasing after reaching a minimum. The best test errors obtained are found to be sensitive to the choice of* ε. *Two different datasets have been considered: the Leukemia dataset (left and middle panels) and the Khan dataset (right panel), as described in Section 6.*

convergence behavior to the least squares fit, as expected. The test errors were found to decrease and then increase after reaching a minimum; furthermore, the best predictive models were found to be sensitive to the choice of ε.

In addition to the above, we also performed a series of experiments on both real and synthetic datasets comparing the performance of LS-Boost(ε) and

FS_ε to other sparse learning methods, namely solution via the LASSO, stepwise regression [14] and FS_0 [14]. Our results are presented in Table A.2. In all the cases, we found that the performance of FS_ε and LS-BOOST(ε) were at least as good as the LASSO solution. And in some cases, the performances of FS_ε and LS-BOOST(ε) were superior. The best predictive models achieved by LS-BOOST(ε) and FS_ε correspond to values of ε that are larger than zero or even close to one – this suggests that a proper choice of ε can lead to superior models.

Statistical properties of R-$FS_{\varepsilon,\delta}$, LASSO and FS_ε: an empirical study. We performed some experiments to evaluate the performance of R-$\text{FS}_{\varepsilon,\delta}$, in terms of predictive accuracy and sparsity of the optimal model, versus the more widely known methods FS_ε and solution via the LASSO. In all the cases, we took a small value of $\varepsilon = 10^{-3}$. We ran R-$\text{FS}_{\varepsilon,\delta}$ on a grid of twenty δ values, with the limiting solution corresponding to the LASSO solution estimate at the particular value of δ selected. In all cases, we found that when δ was large, i.e., larger than the best δ for the LASSO (in terms of obtaining a model with the best predictive performance), R-$\text{FS}_{\varepsilon,\delta}$ delivered a model with excellent statistical properties – R-$\text{FS}_{\varepsilon,\delta}$ led to sparse solutions (the sparsity was similar to that of the best LASSO model) and the predictive performance was as good as, and in some cases better than, the LASSO solution. This suggests that the choice of δ does not play a very crucial role in the R-$\text{FS}_{\varepsilon,\delta}$ algorithm, once it is chosen to be reasonably large; indeed the number of boosting iterations plays a more important role in obtaining good quality statistical estimates. When compared to FS_ε (i.e., the version of R-$\text{FS}_{\varepsilon,\delta}$ with $\delta = \infty$) we observed that the best models delivered by R-$\text{FS}_{\varepsilon,\delta}$ were more sparse (i.e., with fewer non-zeros) than the best FS_ε solutions. This complements a popular belief about boosting in that it delivers models that are quite dense – see the discussion herein in Section A.3.4. Furthermore, it shows that the particular form of regularized boosting that we consider, R-$\text{FS}_{\varepsilon,\delta}$, does indeed induce sparser solutions. Our detailed results are presented in Table A.3.

Comments on Table A.2. In this experiment, we ran FS_ε and LS-BOOST(ε) for thirty different values of ε in the range 0.001 to 0.8. The entire regularization paths for LASSO, FS_0, and the more aggressive Stepwise regression were computed with the `LARS` package. First, we observe that Stepwise regression, which is quite fast in reaching an unconstrained least squares solution, does not perform well in terms of obtaining a model with good predictive performance. The slowly learning boosting methods perform quite well – in fact their performances are quite similar to the best LASSO solutions. A closer inspection shows that FS_ε almost always delivers the best predictive models when ε is allowed to be flexible. While a good automated method to find the optimal value of ε is certainly worth investigating, we leave this for future work (of course, there are excellent heuristics for choosing the optimal ε in practice, such as cross validation, etc.). However, we do highlight that in practice a strictly non-zero learning rate ε may lead to better models than its limiting version $\varepsilon = 0+$.

For Eg-A ($\rho = 0.8$), both LS-Boost(ε) and FS_ε achieved the best model at $\varepsilon = 10^{-3}$. For Eg-A ($\rho = 0$), LS-Boost(ε) achieved the best model at $\varepsilon = 0.1, 0.7, 0.8$ and FS_ε achieved the best model at $\varepsilon = 10^{-3}, 0.7, 0.8$ (both for SNR values 1, 3, 10 respectively). For the Leukemia dataset, LS-Boost(ε) achieved the best model at $\varepsilon = 0.6, 0.7, 0.02$ and FS_ε achieved the best model at $\varepsilon = 0.6, 0.02, 0.02$ (both for SNR values 1, 3, 10 respectively). For the Khan dataset, LS-Boost(ε) achieved the best model at $\varepsilon = 0.001, 0.001, 0.02$ and FS_ε achieved the best model at $\varepsilon = 0.001, 0.02, 0.001$ (both for SNR values 1, 3, 10 respectively).

Type	p	ρ	ε	$\lvert q_{10\%} - \gamma \rvert$	$\lvert q_{50\%} - \gamma \rvert$	$\lvert q_{90\%} - \gamma \rvert$	γ
1	2	-0.99	0.1	2.375e-04	2.375e-04	2.375e-04	9.99762e-01
1	2	-0.99	1.0	1.250e-03	1.250e-03	1.250e-03	9.98750e-01
1	2	-0.5	0.1	1.187e-02	1.187e-02	1.187e-02	9.88125e-01
1	2	-0.5	1.0	6.250e-02	6.250e-02	6.250e-02	9.37500e-01
1	2	0.5	0.1	1.187e-02	1.187e-02	1.187e-02	9.88125e-01
1	2	0.5	1.0	6.250e-02	6.250e-02	6.250e-02	9.37500e-01
1	2	0.99	0.1	2.375e-04	2.375e-04	2.375e-04	9.99762e-01
1	2	0.99	1.0	1.250e-03	1.250e-03	1.250e-03	9.98750e-01
2	2	-0.99	0.1	8.146e-04	8.146e-04	8.146e-04	9.99762e-01
2	2	-0.99	1.0	1.865e-02	1.250e-03	1.250e-03	9.98750e-01
2	2	-0.5	0.1	1.187e-02	1.187e-02	1.187e-02	9.88125e-01
2	2	-0.5	1.0	6.250e-02	6.250e-02	6.250e-02	9.37500e-01
2	2	0.5	0.1	1.187e-02	1.187e-02	1.187e-02	9.88125e-01
2	2	0.5	1.0	6.250e-02	6.250e-02	6.250e-02	9.37500e-01
2	2	0.99	0.1	2.375e-04	2.375e-04	2.375e-04	9.99762e-01
2	2	0.99	1.0	1.865e-02	1.250e-03	1.250e-03	9.98750e-01
2	50	-0.99	0.1	2.804e-03	2.615e-03	6.518e-04	9.99348e-01
2	50	-0.99	1.0	3.431e-03	3.431e-03	3.431e-03	9.96569e-01
2	50	-0.5	0.1	4.738e-03	7.345e-04	7.345e-04	9.99265e-01
2	50	-0.5	1.0	3.866e-03	3.866e-03	3.866e-03	9.96134e-01
2	50	0.5	0.1	1.869e-03	1.750e-03	1.716e-03	9.99525e-01
2	50	0.5	1.0	2.500e-03	2.500e-03	2.500e-03	9.97500e-01
2	50	0.99	0.1	9.676e-05	6.597e-05	5.665e-05	9.99991e-01
2	50	0.99	1.0	1.013e-02	5.000e-05	5.000e-05	9.99950e-01
2	100	-0.99	0.1	1.330e-03	1.217e-03	1.153e-03	9.99679e-01
2	100	-0.99	1.0	1.692e-03	1.692e-03	1.692e-03	9.98308e-01
2	100	-0.5	0.1	1.464e-03	1.352e-03	1.270e-03	9.99638e-01
2	100	-0.5	1.0	1.904e-03	1.904e-03	1.904e-03	9.98096e-01
2	100	0.5	0.1	1.120e-03	9.784e-04	9.459e-04	9.99762e-01
2	100	0.5	1.0	1.753e-02	1.250e-03	1.250e-03	9.98750e-01
2	100	0.99	0.1	7.874e-05	5.335e-05	4.507e-05	9.99995e-01
2	100	0.99	1.0	1.008e-02	2.500e-05	2.500e-05	9.99975e-01
2	200	-0.99	0.1	8.898e-04	7.217e-04	6.814e-04	9.99840e-01
2	200	-0.99	1.0	1.509e-02	8.403e-04	8.403e-04	9.99160e-01
2	200	-0.5	0.1	8.952e-04	7.415e-04	6.965e-04	9.99821e-01
2	200	-0.5	1.0	1.523e-02	9.446e-04	9.446e-04	9.99055e-01
2	200	0.5	0.1	8.429e-04	6.466e-04	6.208e-04	9.99881e-01
2	200	0.5	1.0	1.128e-02	6.250e-04	6.250e-04	9.99375e-01
2	200	0.99	0.1	7.186e-05	4.833e-05	4.043e-05	9.99998e-01
2	200	0.99	1.0	3.417e-04	2.325e-04	2.161e-04	9.99988e-01
2	500	-0.99	0.1	6.974e-04	4.662e-04	4.331e-04	9.99936e-01
2	500	-0.99	1.0	8.227e-03	3.348e-04	3.348e-04	9.99665e-01
2	500	-0.5	0.1	6.554e-04	4.443e-04	4.135e-04	9.99929e-01
2	500	-0.5	1.0	9.550e-03	3.761e-04	3.761e-04	9.99624e-01
2	500	0.5	0.1	7.124e-04	4.565e-04	4.281e-04	9.99953e-01
2	500	0.5	1.0	5.371e-03	3.637e-03	2.500e-04	9.99750e-01
2	500	0.99	0.1	6.848e-05	4.582e-05	3.807e-05	9.99999e-01
2	500	0.99	1.0	1.914e-04	1.201e-04	1.048e-04	9.99995e-01

TABLE A.1

Table showing the observed convergence rate versus the theoretical linear convergence parameter γ (as given by Theorem 2.1) for the LS-BOOST(ε) *algorithm for different values of ε, for the datasets Types 1 and 2. The bounds are observed to be fairly tight until the ℓ_1-norm of the regression coefficients are found to stabilize, which corresponds to the unregularized fit. Since statistically interesting solutions appear typically in the interior of the* LS-BOOST(ε) *path, the agreement between the observed and theoretical bounds are encouraging.*

Dataset	SNR	n	LS-Boost(ε) $\times 10^{-2}$	FS$_\varepsilon$ $\times 10^{-2}$	FS$_0$ $\times 10^{-2}$	Stepwise $\times 10^{-2}$	Lasso $\times 10^{-2}$
Leukemia	1	72	65.9525 (1.8221)	66.7713 (1.8097)	68.1869 (1.4971)	74.5487 (2.6439)	68.3471 (1.584)
	3	72	35.4844 (1.1973)	35.5704 (0.898)	35.8385 (0.7165)	38.9429 (1.8030)	35.3673 (0.7924)
	10	72	13.5424 (0.4267)	13.3690 (0.3771)	13.6298 (0.3945)	14.8802 (0.4398)	13.4929 (0.4276)
Khan	1	63	22.3612 (1.1058)	22.6185 (1.0312)	22.9128 (1.1209)	25.2328 (1.0734)	23.5145 (1.2044)
	3	63	9.3988 (0.4856)	9.4851 (0.4721)	9.6571 (0.3813)	10.8495 (0.3627)	9.2339 (0.404)
	10	63	3.4061 (0.1272)	3.4036 (0.1397)	3.4812 (0.1093)	3.7986 (0.0914)	3.1118 (0.1229)
Eg-A, $\rho = 0.8$	1	50	53.1406 (1.5943)	52.1377 (1.6559)	53.6286 (1.4464)	60.3266 (1.9341)	53.7675 (1.2415)
	3	50	29.1960 (1.2555)	29.2814 (1.0487)	30.0654 (1.0066)	33.4318 (0.8780)	29.8000 (1.2662)
	10	50	12.2688 (0.3359)	12.0845 (0.3668)	12.6034 (0.5052)	15.9408 (0.7939)	12.4262 (0.3660)
Eg-A, $\rho = 0$	1	50	74.1228 (2.1494)	73.8503 (2.0983)	75.0705 (2.5759)	92.8779 (2.7025)	75.0852 (2.1039)
	3	50	38.1357 (2.7795)	40.0003 (1.8576)	41.0643 (1.5503)	43.9425 (3.9180)	41.4932 (2.2092)
	10	50	14.8867 (0.6994)	12.9090 (0.5553)	15.2174 (0.7086)	12.5502 (0.8256)	15.0877 (0.7142)

Table A.2

Table showing the prediction errors (in percentages) of different methods: LS-Boost(ε) *,* FS_ε *(both for different values of* ε*),* FS_0*, (forward) Stepwise regression, and* Lasso*. The numbers within parentheses denote standard errors. (The test errors were all standardized by the test error corresponding to the null model.)* LS-Boost(ε) *,* FS_ε *are found to exhibit similar statistical performances as the* Lasso*, in fact in some examples the boosting methods seem to be marginally better than* Lasso*. The predictive performance of the models were also found to be sensitive to the choice of the learning rate* ε*. For* FS_0 *and Stepwise we used the* `R` *package* `LARS` *[14] to compute the solutions. For all the cases,* $p = 500$*. For Eg-A, we took* $n = 50$*. Both* LS-Boost(ε) *and* FS_ε *were run for a few values of* ε *in the range* $[0.001 - 0.8]$ *– in all cases, the optimal models (see the text for details) for* LS-Boost(ε) *and* FS_ε *were achieved at a value of* ε *larger than its limiting version* $\varepsilon = 0+$*, thereby suggesting the sensitivity of the best predictive model to the learning rate* ε*.*

Real Data Example: Leukemia							
Method	n	p	SNR	Test Error	Sparsity	$\|\hat{\beta}^{\text{opt}}\|_1/\|\hat{\beta}^*\|_1$	$\delta/\delta_{\max}$
FS_ε	72	500	1	0.3431 (0.0087)	28	0.2339	-
R-$\text{FS}_{\varepsilon,\delta}$	72	500	1	0.3411 (0.0086)	25	0.1829	0.56
Lasso	72	500	1	0.3460 (0.0086)	30	1	0.11
FS_ε	72	500	10	0.0681 (0.0014)	67	0.7116	-
R-$\text{FS}_{\varepsilon,\delta}$	72	500	10	0.0659 (0.0014)	60	0.5323	0.56
Lasso	72	500	10	0.0677 (0.0015)	61	1	0.29

Synthetic Data Examples: Eg-B (SNR=1)							
Method	n	p	ρ	Test Error	Sparsity	$\|\hat{\beta}^{\text{opt}}\|_1/\|\hat{\beta}^*\|_1$	$\delta/\delta_{\max}$
FS_ε	50	500	0	0.19001 (0.0057)	56	0.9753	-
R-$\text{FS}_{\varepsilon,\delta}$	50	500	0	0.18692 (0.0057)	51	0.5386	0.71
Lasso	50	500	0	0.19163 (0.0059)	47	1	0.38
FS_ε	50	500	0.5	0.20902 (0.0057)	14	0.9171	-
R-$\text{FS}_{\varepsilon,\delta}$	50	500	0.5	0.20636 (0.0055)	10	0.1505	0.46
Lasso	50	500	0.5	0.21413 (0.0059)	13	1	0.07
FS_ε	50	500	0.9	0.05581 (0.0015)	4	0.9739	-
R-$\text{FS}_{\varepsilon,\delta}$	50	500	0.9	0.05507 (0.0015)	4	0.0446	0.63
Lasso	50	500	0.9	0.09137 (0.0025)	5	1	0.04

TABLE A.3

Table showing the statistical properties of R-$FS_{\varepsilon,\delta}$ as compared to Lasso *and FS_ε. Both R-$FS_{\varepsilon,\delta}$ and FS_ε use $\varepsilon = 0.001$. The model that achieved the best predictive performance (test-error) corresponds to $\hat{\beta}^{opt}$. The limiting model (as the number of boosting iterations is taken to be infinitely large) for each method is denoted by $\hat{\beta}^*$. "Sparsity" denotes the number of coefficients in $\hat{\beta}^{opt}$ larger than 10^{-5} in absolute value. $\delta_{\max}$ is the ℓ_1-norm of the least squares solution with minimal ℓ_1-norm. Both R-$FS_{\varepsilon,\delta}$ and* Lasso *were run for a few δ values of the form $\eta\delta_{\max}$, where η takes on twenty values in $[0.01, 0.8]$. For the real data instances, R-$FS_{\varepsilon,\delta}$ and* Lasso *were run for a maximum of 30,000 iterations, and FS_ε was run for 20,000 iterations. For the synthetic examples, all methods were run for a maximum of 10,000 iterations. The best models for R-$FS_{\varepsilon,\delta}$ and FS_ε were all obtained in the interior of the path. The best models delivered by R-$FS_{\varepsilon,\delta}$ are seen to be more sparse and have better predictive performance than the best models obtained by FS_ε. The performances of* Lasso *and R-$FS_{\varepsilon,\delta}$ are found to be quite similar, though in some cases R-$FS_{\varepsilon,\delta}$ is seen to be at an advantage in terms of better predictive accuracy. The test errors were all standardized by the test error corresponding to the null model.*