A New Perspective on Boosting in Linear Regression via Subgradient Optimization and Relatives

Paul Grigas[*]

May 25, 2016

1 Boosting Algorithms in Linear Regression

Boosting [6,9,12,15,16] is an extremely successful and popular supervised learning technique that combines multiple "weak" learners into a more powerful "committee." AdaBoost [7,12,16], developed in the context of classification, is one of the earliest and most influential boosting algorithms. In our paper [5], we analyze boosting algorithms in linear regression [3,8,9] from the perspective of modern first-order methods in convex optimization. This perspective has two primary upshots: *(i)* it leads to first-ever computational guarantees for existing boosting algorithms, and *(ii)* it leads to new boosting algorithms with novel connections to the Lasso [18].

Notation We use the usual linear regression 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 $\beta \in \mathbb{R}^p$. Each column of $\mathbf{X}$ corresponds to a particular feature or predictor variable, and each row corresponds to a particular observed sample. We assume herein that the features $\mathbf{X}_i$ have been centered to have zero mean and unit ℓ_2 norm, i.e., $\|\mathbf{X}_i\|_2 = 1$ for $i = 1, \ldots, p$, and $\mathbf{y}$ is also centered to have zero mean. For a regression coefficient vector β, the predicted value of the response is given by $\mathbf{X}\beta$ and $r = \mathbf{y} - \mathbf{X}\beta$ denotes the residuals. Let e_j denote the j^{th} unit vector in $\mathbb{R}^p$ and let $\|v\|_0$ denote the number of nonzero coefficients in the vector v. Denote the empirical least squares loss function by $L_n(\beta) := \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\beta\|_2^2$, let $L_n^* := \min_{\beta \in \mathbb{R}^p} L_n(\beta)$, and let $\hat{\beta}_{\text{LS}}$ denote an arbitrary minimizer of $L_n(\beta)$, i.e., $\hat{\beta}_{\text{LS}} \in \arg\min_{\beta \in \mathbb{R}^p} L_n(\beta)$. Finally, let $\lambda_{\text{pmin}}(\mathbf{X}^T\mathbf{X})$ denote the smallest nonzero (and hence positive) eigenvalue of $\mathbf{X}^T\mathbf{X}$.

Boosting and Implicit Regularization The first boosting algorithm we consider is the Incremental Forward Stagewise algorithm [3,12] presented below, which we refer to as FS_ε.

Algorithm: Incremental Forward Stagewise Regression – FS_ε

Fix the learning rate $\varepsilon > 0$ and number of iterations M.

[*] MIT Operations Research Center, 77 Massachusetts Avenue, Cambridge, MA 02139 (mailto: pgrigas@mit.edu).

```
Initialize at $\hat{r}^0 = \mathbf{y}$, $\hat{\beta}^0 = 0$, $k = 0$ .
  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. $\hat{\beta}^{k+1}_{j_k} \leftarrow \hat{\beta}^k_{j_k} + \varepsilon \ \mathrm{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k})$ and $\hat{\beta}^{k+1}_j \leftarrow \hat{\beta}^k_j \ , j \neq j_k$
     $\hat{r}^{k+1} \leftarrow \hat{r}^k - \varepsilon \ \mathrm{sgn}((\hat{r}^k)^T \mathbf{X}_{j_k}) \mathbf{X}_{j_k}$ .
```

At the k^{th} iteration, FS$_\varepsilon$ chooses a column $\mathbf{X}_{j_k}$, corresponding to a particular feature that is the most correlated (in absolute value) with the current residuals and then updates the corresponding regression coefficient by an amount $\varepsilon > 0$, called the learning rate (or shrinkage factor).

A close cousin of FS$_\varepsilon$ is the least squares boosting algorithm, or LS-Boost(ε), proposed in [8]. The LS-Boost(ε) algorithm is identical to FS$_\varepsilon$ except that LS-Boost(ε) changes the amount by which the selected coefficient is updated at each iteration – at the k^{th} iteration, LS-Boost(ε) updates:

$$\hat{\beta}^{k+1}_{j_k} \leftarrow \hat{\beta}^k_{j_k} + \varepsilon \left((\hat{r}^k)^T \mathbf{X}_{j_k}\right) \text{ and } \hat{\beta}^{k+1}_j \leftarrow \hat{\beta}^k_j \ , j \neq j_k$$

$$\hat{r}^{k+1} \leftarrow \hat{r}^k - \varepsilon \left((\hat{r}^k)^T \mathbf{X}_{j_k}\right) \mathbf{X}_{j_k} \ ,$$

where now $\varepsilon \in (0, 1]$.

Note that both algorithms often lead to models with attractive statistical properties [1,2,8,12]. In this linear regression setting, while there may be several important concerns, it is often of primary importance to produce a parsimonious model with good out of sample predictive performance. When p is small relative to n, minimizing the empirical least squares loss function $L_n(\beta)$ usually achieves this goal. On the other hand, when $n, p \gg 0$ (and particularly when $p > n$), $\hat{\beta}_{\text{LS}}$ often has poor predictive performance; in other words, $\hat{\beta}_{\text{LS}}$ *overfits* the training data. Additionally $\hat{\beta}_{\text{LS}}$ is almost always fully dense. Regularization techniques enable one to find a model with better predictive performance by balancing two competing objectives: *(i)* data fidelity, or how well the model fits the training data, and *(ii)* "shrinkage," or a measure of model simplicity. Shrinkage is often measured using $\|\beta\|$ for some appropriate norm $\|\cdot\|$, whereby a coefficient vector with a relatively small value of $\|\beta\|$ exhibits more shrinkage. The FS$_\varepsilon$ and LS-Boost(ε) algorithms are effective, even in settings where $n, p \gg 0$ and/or $p > n$, because they each impart a type of *implicit regularization* by tracing out a path of models with varying levels of data fidelity and shrinkage.

For both FS$_\varepsilon$ and LS-Boost(ε), the choices of ε and M play crucial roles in the statistical behavior of the algorithm. Let us consider LS-Boost(ε) alone for now. Setting $\varepsilon = 1$ corresponds to minimizing the empirical least squares loss function $L_n(\beta)$ along the direction of the selected feature, i.e., it holds that $(\hat{r}^k)^T \mathbf{X}_{j_k} = \arg\min_{u \in \mathbb{R}} L_n(\beta^k + u e_{j_k})$. Qualitatively speaking, LS-Boost(ε) does eventually minimize the empirical least squares loss function as long as $\varepsilon > 0$, but a small value of ε (for example, $\varepsilon = 0.001$) slows down the rate of convergence as compared to the choice $\varepsilon = 1$. Thus it may seem counterintuitive to set $\varepsilon < 1$; however with a small value of ε it is possible to explore a larger class of models, with varying degrees of shrinkage. It has been observed empirically that small values of ε often lead to models with better predictive power [8]. In practice, one might set ε relatively small and use a holdout dataset to select the best performing model found throughout the course of the algorithm; in many instances the selected model is found long before convergence to the empirical least squares solution. The role of M and ε in FS_ε is very similar. In short, both M and ε together control the training error (data fidelity) and the amount of shrinkage (regularization) for both LS-Boost(ε) and FS_ε. We refer the reader to Figure 1, depicting the evolution of the algorithmic properties of FS_ε and LS-Boost(ε) as a function of M and ε.

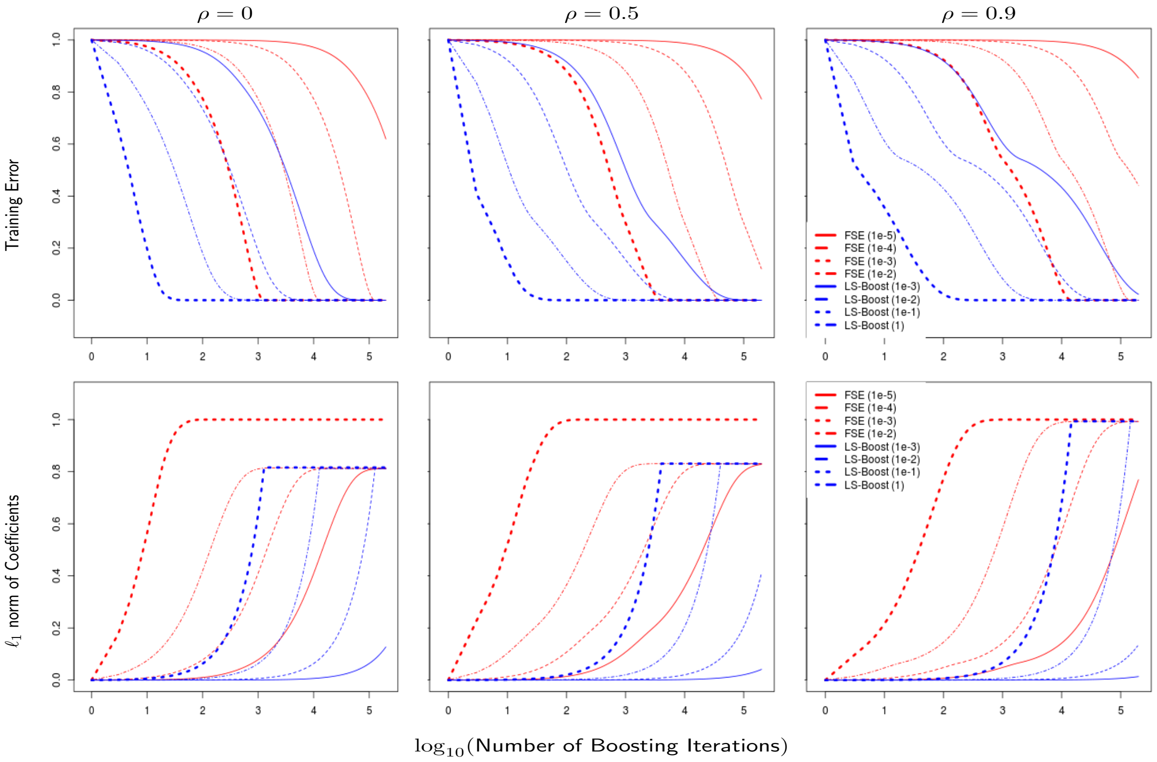

Figure 1: Evolution of LS-Boost(ε) and FS_ε versus iterations (in the log-scale), run on a synthetic dataset with $n = 50$, $p = 500$; the features are drawn from a Gaussian distribution with pairwise correlations ρ. The true β has ten nonzeros with $\beta_i = 1, i \leq 10$ and $\text{SNR} = 1$. Three different values of ρ have been considered ($\rho = 0, 0.5$ and 0.9) and ε varies from $\varepsilon = 10^{-5}$ to $\varepsilon = 1$. The top row shows the training errors for different learning rates, and the bottom row shows the ℓ_1 norm of the coefficients produced by the different algorithms for different learning rates. (Here the values have all been re-scaled so that the y-axis lies in $[0, 1]$).

2 Computational Guarantees for FS_ε and LS-Boost(ε) Through the Lens of Subgradient Descent

Up until the present work, and as pointed out by [12], the understanding of how the algorithmic parameters ε and M control the tradeoffs between data fidelity and shrinkage in FS_ε and LS-Boost(ε) has been rather qualitative. One of the contributions of the full paper is a precise quantification of this tradeoff, for both FS_ε and LS-Boost(ε). Indeed, the paper presents, for the first time, precise descriptions of how the quantities ε and M control the amount of training error and regularization in FS_ε and LS-Boost(ε). These precise computational guarantees are enabled by new connections to first-order methods in convex optimization. In particular, the paper presents a new unifying framework for interpreting FS_ε and LS-Boost(ε) as instances of the subgradient descent method of convex optimization, applied to the problem of minimizing the largest correlation between residuals and predictors.

Boosting as Subgradient Descent 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:

$$\begin{array}{lllll} \text{Correlation Minimization (CM)}: & f^* := & \min\limits_{r} & f(r) & := \ \|\mathbf{X}^T r\|_\infty \\ & & \text{s.t.} & r \in P_{\text{res}} \ , & \end{array} \tag{1}$$

which we dub the "Correlation Minimization" problem, or CM for short, since $f(r)$ is the largest absolute correlation between the residual vector r and the predictors. 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.

The subgradient descent method (see [17], for example) is a simple generalization of the method of gradient descent to the case when $f(\cdot)$ is not differentiable. As applied to the CM problem (1), the subgradient descent method has the following update scheme:

$$\begin{array}{lll} \text{Compute a subgradient of } f(\cdot) \text{ at } r^k & : & g^k \in \partial f(r^k) \\ \text{Peform update at } r^k & : & r^{k+1} \leftarrow \Pi_{P_{\text{res}}}(r^k - \alpha_k g^k) \ , \end{array} \tag{2}$$

where $\partial f(r)$ denotes the set of subgradients of $f(\cdot)$ at r and $\Pi_{P_{\text{res}}}$ denotes the (Euclidean) projection operator onto P_{res}, namely $\Pi_{P_{\text{res}}}(\bar{r}) := \arg\min\limits_{r \in P_{\text{res}}} \|r - \bar{r}\|_2$.

The following proposition states that the boosting algorithms FS_ε and LS-Boost(ε) can be viewed as instantiations of the subgradient descent method to solve the CM problem (1).

Proposition 2.1. *Consider the subgradient descent method* (2) *with step-size sequence* $\{\alpha_k\}$ *to solve the correlation minimization (CM) problem* (1), *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 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* $\tilde{u}_{j_k} := (\hat{r}^k)^T \mathbf{X}_{j_k} = \arg\min_u \|\hat{r}^k - \mathbf{X}_{j_k} u\|_2^2$.

Some Computational Guarantees for FS_ε Proposition 2.1 is interesting especially since FS_ε and LS-Boost(ε) have been traditionally interpreted as greedy coordinate descent or steepest descent type procedures [10,12]. Furthermore the following theorem presents relevant convergence properties of FS_ε, some of which are direct consequences of Proposition 2.1 based on well-known computational guarantees associated with the subgradient descent method [13,14].

Theorem 2.1. (Some Convergence Properties of FS_ε) *Consider the* FS_ε *algorithm with learning rate* ε. *Let* $M \geq 0$ *be the total number of iterations. Then there exists an index* $i \in \{0, \ldots, M\}$ *for which the following bounds hold:*

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

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

(iii) (sparsity of coefficients): $\|\hat{\beta}^i\|_0 \leq M$.

Theorem 2.1 gives a flavor of some of the computational guarantees included in the full paper; the paper includes additional results regarding convergence of regression coefficients, prediction distances, and correlation values. Furthermore, the paper also includes an analogous theorem for LS-Boost(ε), which highlights the differences in convergence patterns between the two algorithms. Theorem 2.1 (and related results included in the paper) provides, for the first time, a precise theoretical description of the amount of data fidelity and shrinkage/regularization imparted by running FS$_\varepsilon$ for a fixed but arbitrary number of iterations, for *any* dataset. Moreover, this result sheds light on the data fidelity *vis-à-vis* shrinkage characteristics of FS$_\varepsilon$. In particular, Theorem 2.1 demonstrates explicitly how (bounds on) the training error and ℓ_1-shrinkage depend on the algorithmic parameters ε and M, which implies an explicit tradeoff between data fidelity and shrinkage that is controlled by these parameters. Indeed, let TBound and SBound denote the training error bound and shrinkage bound in parts *(i)* and *(ii)* of Theorem 2.1, respectively. Then simple manipulation of the arithmetic in these two bounds yields the following tradeoff equation:

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

In the full paper, we extensively discuss the consequences of Theorem 2.1 and related results in terms of improved understanding of the behavior of FS$_\varepsilon$ and LS-Boost(ε).

3 Boosting and Lasso

As mentioned previously, FS$_\varepsilon$ and LS-Boost(ε) are effective even in high-dimensional settings where $p > n$ since they implicitly deliver regularized models. An alternative and very popular approach in such settings is based on an explicit regularization scheme, namely ℓ_1-regularized regression, i.e., Lasso [18]. The constraint version of Lasso with regularization parameter $\delta \geq 0$ is given by the following convex quadratic optimization problem:

$$\text{Lasso} \ : \quad L^*_{n,\delta} := \begin{array}{ll} \min\limits_{\beta} & \frac{1}{2n}\|\mathbf{y}-\mathbf{X}\beta\|_2^2 \\ \text{s.t.} & \|\beta\|_1 \leq \delta \ . \end{array} \tag{3}$$

Although Lasso and the previously discussed boosting methods originate from different perspectives, there are interesting similarities between the two, as is nicely explored in [3,11,12]. Figure 2 (top panel) shows an example where the Lasso profile/path (the set of solutions of (3) as δ varies) is similar to the trajectories of FS$_\varepsilon$ and LS-Boost(ε) (for small values of ε). Although they are different in general (Figure 2, bottom panel), [3,11] explores the connection more deeply.

One of the aims of our work is to contribute some substantial further understanding of the relationship between Lasso, FS$_\varepsilon$, and LS-Boost(ε), particularly for *arbitrary* datasets where such understanding is still fairly limited. Motivated thusly, we introduce a new boosting algorithm, called R-FS$_{\varepsilon,\delta}$ (regularized FS$_\varepsilon$), that includes an additional shrinkage step as compared to FS$_\varepsilon$. That is R-FS$_{\varepsilon,\delta}$ first shrinks all of the coefficients, then adds ε to the selected coefficient; R-FS$_{\varepsilon,\delta}$ replaces Step 3 of FS$_\varepsilon$ by:

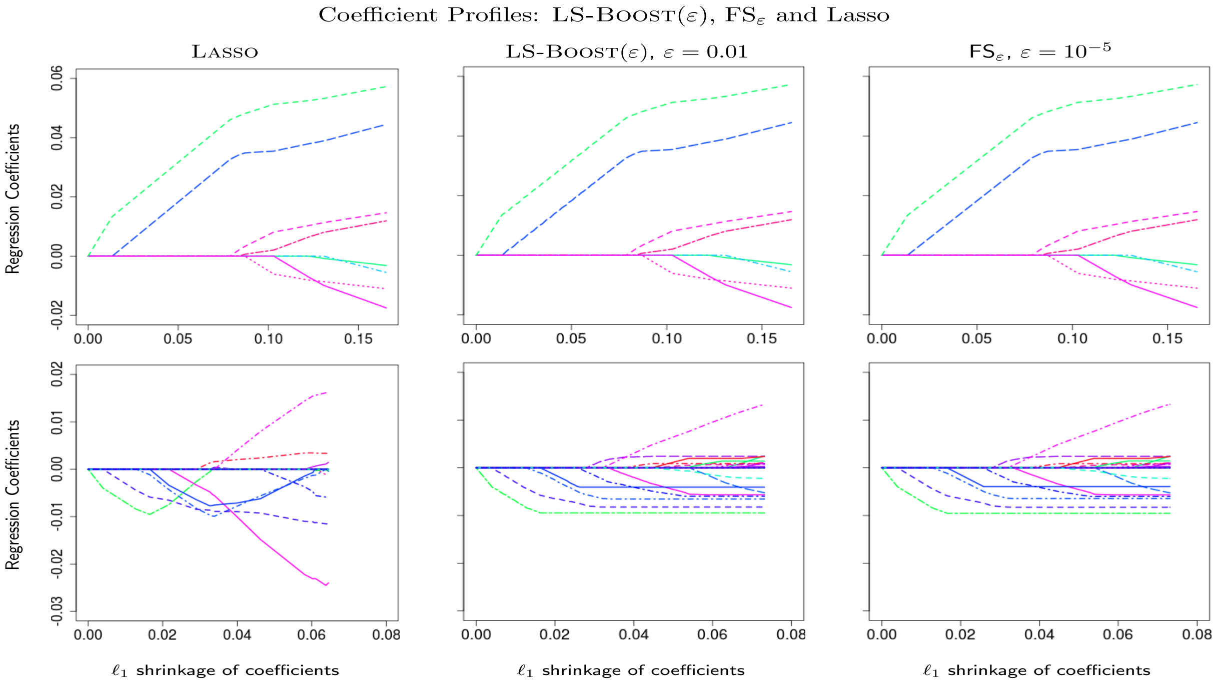

Figure 2: Coefficient Profiles for different algorithms as a function of the ℓ_1 norm of the regression coefficients on two different datasets. The top row corresponds to a dataset where the coefficient profiles look very similar, and the bottom row corresponds to a dataset where the coefficient profile of Lasso is seen to be different from FS$_\varepsilon$ and LS-Boost(ε).

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

where $\delta > 0$ is an additional algorithmic parameter. Note that one can easily verify the formula for updating the residuals based on the coefficient update. Furthermore, R-FS$_{\varepsilon,\delta}$ with $\delta = +\infty$ is exactly FS$_\varepsilon$.

It turns out that R-FS$_{\varepsilon,\delta}$ is precisely related to the Lasso problem through duality. Consider the following parametric family of optimization problems indexed by $\delta \in (0, \infty]$:

$$
\begin{array}{llcll}
\mathrm{RCM}_\delta : & f^*_\delta := \min\limits_{r} & f_\delta(r) & := & \|\mathbf{X}^T r\|_\infty + \frac{1}{2\delta}\|r - \mathbf{y}\|_2^2 \\
& \text{s.t.} & r \in P_{\text{res}} & := & \{r \in \mathbb{R}^n : r = \mathbf{y} - \mathbf{X}\beta \text{ for some } \beta \in \mathbb{R}^p\} \ ,
\end{array}
\tag{4}
$$

where “RCM” connotes Regularlized Correlation Minimization.

In the full paper, we establish the following connections between R-FS$_{\varepsilon,\delta}$, the RCM problem, and the Lasso problem:

1. The RCM problem (4) is equivalent to the dual problem of the Lasso (3).
2. R-FS$_{\varepsilon,\delta}$ is an instance of subgradient descent applied to the RCM problem (4).

The R-FS$_{\varepsilon,\delta}$ algorithm is also related to a variant of the Frank-Wolfe method in convex optimization [4], applied directly to Lasso.

Furthermore, we show the following properties of the new algorithm R-FS$_{\varepsilon,\delta}$:

- As the number of iterations become large, R-FS$_{\varepsilon,\delta}$ delivers an approximate Lasso solution.
- R-FS$_{\varepsilon,\delta}$ has computational guarantees analogous to Theorem 2.1 that provide a precise description of data-fidelity *vis-à-vis* ℓ_1 shrinkage.
- R-FS$_{\varepsilon,\delta}$ specializes to FS$_{\varepsilon}$, LS-Boost(ε) and the Lasso depending on the parameter value δ and the learning rates (step-sizes) used therein.
- An adaptive version of R-FS$_{\varepsilon,\delta}$, which we call PATH-R-FS$_{\varepsilon}$, is shown to approximate the path of Lasso solutions with precise bounds that quantify the approximation error over the path.
- In our computational experiments, we observe that R-FS$_{\varepsilon,\delta}$ leads to models with statistical properties that compare favorably with the Lasso and FS$_{\varepsilon}$. R-FS$_{\varepsilon,\delta}$ also leads to models that are sparser than FS$_{\varepsilon}$.

In total, we establish that FS$_{\varepsilon}$, LS-Boost(ε) and Lasso can be viewed as special instances of one "grand" algorithm: the subgradient descent method applied to the RCM problem (4).

4 Summary

We analyze boosting algorithms in linear regression from the perspective modern first-order methods in convex optimization. We show that classic boosting algorithms in linear regression, FS$_{\varepsilon}$ and LS-Boost(ε), can be viewed as subgradient descent to minimize the maximum absolute correlation between features and residuals. We also propose a modification of FS$_{\varepsilon}$ that yields an algorithm for the Lasso, and that computes the Lasso path. Our perspective leads to first-ever comprehensive computational guarantees for all of these boosting algorithms, which provide a precise theoretical description of the amount of data-fidelity and regularization imparted by running a boosting algorithm with a pre-specified learning rate for a fixed but arbitrary number of iterations, for any dataset.

5 Acknowledgments

The results described here and in the full paper is joint work with Robert Freund and Rahul Mazumder, and I am grateful for their collaboration and their general support. I would also like to thank the prize committee – Mohit Tawarmalani, Fatma Kılınç-Karzan, Warren Powell, and Uday Shanbhag – for choosing to honor this work, and more importantly for their service to the optimization community in general.

References

[1] P. Bühlmann. Boosting for high-dimensional linear models. *Annals of Statistics*, pages 559–583, 2006.

[2] P. Bühlmann and T. Hothorn. Boosting algorithms: regularization, prediction and model fitting (with discussion). *Statistical Science*, 22(4):477–505, 2008.

[3] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression (with discussion). *Annals of Statistics*, 32(2):407–499, 2004.

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

[5] R. M. Freund, P. Grigas, and R. Mazumder. A new perspective on boosting in linear regression via subgradient optimization and relatives. *arXiv preprint arXiv:1505.04243*, 2015.

[6] Y. Freund. Boosting a weak learning algorithm by majority. *Information and computation*, 121(2):256–285, 1995.

[7] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156. Morgan Kauffman, San Francisco, 1996.

[8] J. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

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

[10] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29:1189–1232, 2000.

[11] T. Hastie, J. Taylor, R. Tibshirani, and G. Walther. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1:1–29, 2007.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer Verlag, New York, 2009.

[13] Y. E. Nesterov. *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied Optimization.* Kluwer Academic Publishers, Boston, 2003.

[14] B. Polyak. *Introduction to Optimization.* Optimization Software, Inc., New York, 1987.

[15] R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[16] R. Schapire and Y. Freund. *Boosting: Foundations and Algorithms.* Adaptive computation and machine learning. Mit Press, 2012.

[17] N. Z. Shor. *Minimization Methods for Non-Differentiable Functions*, volume 3 of *Springer Series in Computational Mathematics.* Springer, Berlin, 1985.

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