The Annals of Statistics
2012, Vol. 40, No. 2, 1074–1101
DOI: 10.1214/12-AOS997
© Institute of Mathematical Statistics, 2012

CHARACTERIZING L_2BOOSTING

BY JOHN EHRLINGER AND HEMANT ISHWARAN

Cleveland Clinic and University of Miami

We consider L_2Boosting, a special case of Friedman's generic boosting algorithm applied to linear regression under L_2-loss. We study L_2Boosting for an arbitrary regularization parameter and derive an exact closed form expression for the number of steps taken along a fixed coordinate direction. This relationship is used to describe L_2Boosting's solution path, to describe new tools for studying its path, and to characterize some of the algorithm's unique properties, including active set cycling, a property where the algorithm spends lengthy periods of time cycling between the same coordinates when the regularization parameter is arbitrarily small. Our fixed descent analysis also reveals a *repressible condition* that limits the effectiveness of L_2Boosting in correlated problems by preventing desirable variables from entering the solution path. As a simple remedy, a data augmentation method similar to that used for the elastic net is used to introduce L_2-penalization and is shown, in combination with decorrelation, to reverse the repressible condition and circumvents L_2Boosting's deficiencies in correlated problems. In itself, this presents a new explanation for why the elastic net is successful in correlated problems and why methods like LAR and lasso can perform poorly in such settings.

1. Introduction. Given data $\{y_i, \mathbf{x}_i\}_1^n$, where y_i is the response and $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,p}) \in \mathbb{R}^p$ is the p-dimensional covariate, the goal in many analyses is to approximate the unknown function $F(\mathbf{x}) = \mathbb{E}(y|\mathbf{x})$ by minimizing a specified loss function $L(y, F)$ [a common choice is L_2-loss, $L(y, F) = (y - F)^2/2$]. In trying to estimate F, one strategy is to make use of a large system of possibly redundant functions $\mathcal{H}$. If $\mathcal{H}$ is rich enough, then it is reasonable to expect F to be well approximated by an additive expansion of the form

$$F(\mathbf{x}; \{\beta_k, \alpha_k\}_1^K) = \sum_{k=1}^{K} \beta_k h(\mathbf{x}; \alpha_k),$$

where $h(\mathbf{x}; \alpha) \in \mathcal{H}$ are base learners parameterized by $\alpha \in \Theta$. To estimate F, a joint multivariable optimization over $\{\beta_k, \alpha_k\}_1^K$ may be used. But such an optimization may be computationally slow or even infeasible for large dictionaries. Overfitting may also result. To circumvent this problem, iterative descent algorithms are often used.

Received October 2011; revised March 2012.

MSC2010 subject classifications. Primary 62J05; secondary 62J99.

Key words and phrases. Critical direction, gradient-correlation, regularization, repressibility, solution path.

One popular method is the gradient descent algorithm described by Friedman (2001), closely related to the method of "matching pursuit" used in the signal processing literature [Mallat and Zhang (1993)]. This algorithm is applicable to a wide range of problems and loss functions, and is now widely perceived to be a generic form of boosting. For the mth step, $m = 1, \dots, M$, one solves

$$\rho_m = \arg\min_{\rho \in \mathbb{R}} \sum_{i=1}^{n} L\big(y_i, F_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \alpha_m)\big), \tag{1.1}$$

where

$$\alpha_m = \arg\min_{\alpha \in \Theta} \sum_{i=1}^{n} [g_m(\mathbf{x}_i) - h(\mathbf{x}_i; \alpha)]^2 \tag{1.2}$$

identifies the closest base learner to the gradient $\mathbf{g}_m = (g_m(\mathbf{x}_1), \dots, g_m(\mathbf{x}_n))^T$ in L_2-distance, where $g_m(\mathbf{x}_i)$ is the gradient evaluated at the current value $F_{m-1}(\mathbf{x}_i)$, and is defined by

$$g_m(\mathbf{x}_i) = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F_{m-1}(\mathbf{x}_i)} = -L'(y_i, F_{m-1}(\mathbf{x}_i)).$$

The mth update for the predictor of F is

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \nu \rho_m h(\mathbf{x}; \alpha_m),$$

where $0 < \nu \leq 1$ is a regularization (learning) parameter.

In this paper, we study Friedman's algorithm under L_2-loss in linear regression settings assuming an $n \times p$ design matrix $\mathbf{X} = [\mathbf{X}_1, \dots, \mathbf{X}_p]$, where $\mathbf{X}_k = (x_{1,k}, \dots, x_{n,k})^T$ denotes the kth column. Here $\mathbf{X}_k$ represents the kth base learner; that is, $h(\mathbf{x}_i; k) = x_{i,k}$ where $k = \alpha$ and $\Theta = \{1, \dots, p\}$. It is well known that under L_2-loss the gradient simplifies to the residual $g_m(\mathbf{x}_i) = y_i - F_{m-1}(\mathbf{x}_i)$. This is particularly attractive for a theoretical treatment as it allows one to combine the line-search (1.1) and the learner-search (1.2) into a single step because the L_2-loss function can be expressed as $L(y_i, F_{m-1}(\mathbf{x}_i) + \rho x_{i,k}) = (g_m(\mathbf{x}_i) - \rho x_{i,k})^2$. The optimization problem becomes

$$\{\rho_m, k_m\} = \arg\min_{\rho \in \mathbb{R}, 1 \leq k \leq p} \|\mathbf{g}_m - \rho \mathbf{X}_k\|^2.$$

It is common practice to standardize the response by removing its mean which eliminates the issue of whether an intercept should be included as a column of $\mathbf{X}$. It is also common to standardize the columns of $\mathbf{X}$ to have a mean of zero and squared-length of one. Thus, throughout, we assume the data is standardized according to

$$\sum_{i=1}^{n} y_i = 0, \qquad \sum_{i=1}^{n} x_{i,k} = 0, \qquad \sum_{i=1}^{n} x_{i,k}^2 = 1, \qquad k = 1, \dots, p. \tag{1.3}$$

Algorithm 1 L_2Boosting

```
1: Initialize $F_{0,i} = 0$ for $i = 1, \dots, n$
2: for $m = 1$ to $M$ do
3:     $k_m = \arg\max_{1 \le k \le p} |\mathbf{X}_k^T \mathbf{g}_m|$, where $\mathbf{g}_m = \mathbf{y} - \mathbf{F}_{m-1}$
4:     $\mathbf{F}_m = \mathbf{F}_{m-1} + \nu \rho_m \mathbf{X}_{k_m}$, where $\rho_m = \mathbf{X}_{k_m}^T \mathbf{g}_m$
5: end for
```

The condition $\sum_{i=1}^n x_{i,k}^2 = 1$ leads to a particularly useful simplification:

$$\rho_m = \mathbf{X}_{k_m}^T \mathbf{g}_m, \qquad k_m = \underset{1 \le k \le p}{\arg\max} |\mathbf{X}_k^T \mathbf{g}_m|.$$

Thus, the search for the most favorable direction is equivalent to determining the largest absolute value $|\mathbf{X}_k^T \mathbf{g}_m|$. We refer to $\mathbf{X}_k^T \mathbf{g}_m$ as the *gradient-correlation* for k. We shall refer to Friedman's algorithm under the above settings as L_2Boosting. Algorithm 1 provides a formal description of the algorithm [we use $\mathbf{F}_{m-1} = (F_{m-1}(\mathbf{x}_1), \dots, F_{m-1}(\mathbf{x}_n))^T$ for notational convenience].

Properties of stagewise algorithms similar to L_2Boosting have been studied extensively under the assumption of an infinitesimally small regularization parameter. Efron et al. (2004) considered a forward stagewise algorithm FS_ε, and showed under a convex cone condition that the Least Angle Regression (LAR) algorithm yields the solution path for FS_0, the limit of FS_ε as $\varepsilon \to 0$. This shows that FS_ε, a variant of boosting, and the lasso [Tibshirani (1996)] are related in some settings. Hastie et al. (2007) showed in general that the solution path of FS_0 is equivalent to the path of the monotone lasso.

However, much less work has focused on stagewise algorithms assuming an arbitrary learning parameter $0 < \nu \le 1$. An important exception is Bühlmann (2006) who studied L_2Boosting with componentwise linear least squares, the same algorithm studied here, and proved consistency for arbitrary ν under a sparsity assumption where p can increase at an exponential rate relative to n. As pointed out in Bühlmann (2006), the FS_ε algorithm studied by Efron et al. (2004) bears similarities to L_2Boosting. It is identical to Algorithm 1, except for line 4, where ε is used in place of ν and

$$\mathbf{F}_m = \mathbf{F}_{m-1} + \varepsilon \delta_m \mathbf{X}_{k_m}, \qquad \text{where } \delta_m = \text{sgn}[\text{corr}(\mathbf{g}_m, \mathbf{X}_{k_m})].$$

Thus, FS_ε replaces the gradient-correlation ρ_m with the sign of the gradient-correlation δ_m. For infinitesimally small ν this difference appears to be inconsequential, and it is generally believed that the two limiting solution paths are equal [Hastie (2007)]. In general, however, for arbitrary $0 < \nu \le 1$, the two solution paths are different. Indeed, Bühlmann (2006) indicated certain unique advantages possessed by L_2Boosting. Other related work includes Bühlmann and Yu (2003), who described a bias-variance decomposition of the mean-squared-error of a variant of L_2Boosting.

1.1. *Outline and contributions.* In this paper, we investigate the properties of L_2Boosting assuming an arbitrary learning parameter $0 < \nu \leq 1$. During L_2Boosting's descent along a fixed coordinate direction, a new coordinate becomes more favorable when it becomes closest to the current gradient. But when does this actually occur? We provide an exact simple closed form expression for this quantity: the number of iterations to favorability (Theorem 2 of Section 2). This core identity is used to describe L_2Boosting's solution path (Theorem 3), to introduce new tools for studying its path and to study and characterize some of the algorithm's unique properties. One of these is active set cycling, a property where the algorithm spends lengthy periods of time cycling between the same coordinates when ν is small (Section 3).

Our fixed descent identity also reveals how correlation affects L_2Boosting's ability to select variables in highly correlated problems. We identify a *repressible condition* that prevents a new variable from entering the active set, even though that variable may be highly desirable (Section 4). Using a data augmentation approach, similar to that used for calculating the elastic net [Zou and Hastie (2005)], we describe a simple method for adding L_2-penalization to L_2Boosting (Section 5). In combination with decorrelation, this reverses the repressible condition and improves L_2Boosting's performance in correlated problems. Because L_2Boosting is known to approximate forward stagewise algorithms for arbitrarily small ν, it is natural to expect these results to apply to such algorithms like LAR and lasso, and thus our results provide a new explanation for why these algorithms may perform poorly in correlated settings and why methods like the elastic net, which makes use of L_2-penalization, are more adept in such settings. All proofs in this manuscript can be found in the supplemental article [Ehrlinger and Ishwaran (2012)].

2. Fixed descent analysis. To analyze L_2Boosting we introduce the following notation useful for describing its solution path. Let $\{l_1, \dots, l_{M^*}\}$ be the $M^* \leq M$ nonduplicated values in order of appearance of the selected coordinate directions $\mathcal{B}_M = \{k_1, \dots, k_M\}$. We refer to these ordered, nonduplicated values as *critical directions* of the path. For example, if $\mathcal{B}_M = \{5, 5, 5, 3, 5, 1, 4, 4, 5\}$, the critical directions are $\{5, 3, 5, 1, 4, 5\}$ and $M^* = 6$. To formally describe the solution path we introduce the following nomenclature.

DEFINITION 1. The descent length along a critical direction l_r is denoted by L_r. The critical point S_r is the step number at which the descent along l_r ends. Thus, following step S_{r-1}, the descent is along l_r for a total of L_r steps, ending at step S_r.

The set of values $(l_r, L_r, S_r)_1^{M^*}$ can be used to formally describe the solution path of L_2Boosting: the algorithm begins by descending along direction l_1 (the first critical direction) for L_1 steps, after which it switches to a descent along direction l_2 (the second critical direction) for a total of L_2 steps. This continues with the last

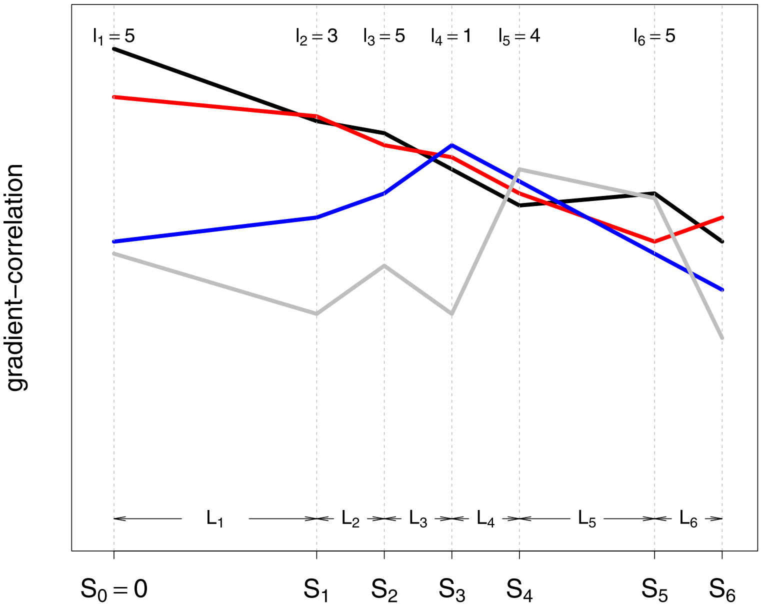

FIG. 1. *Solution path for L_2Boosting where $\mathcal{B}_M = \{5, 5, 5, 3, 5, 1, 4, 4, 5\}$. The $M^* = 6$ critical directions are $(l_r)_1^6 = (5, 3, 5, 1, 4, 5)$ with critical descent step lengths $(L_r)_1^6 = (3, 1, 1, 1, 2, 1)$ and critical points $(S_r)_1^6 = (3, 4, 5, 6, 8, 9)$.*

descent along l_{M^*} (the final critical direction) for a total of L_{M^*} steps. See Figure 1 for illustration of the notation.

A key observation is that L_2Boosting's behavior along a given descent is deterministic except for its descent length L_r (number of steps). If we could determine the descent length, a quantity we show is highly amenable to analysis, then an exact description of the solution path becomes possible as L_2Boosting can be conceptualized as collection of such fixed paths.

Imagine then that we are at step m' of the algorithm and that in the following step a new critical direction k is formed. Let us study the descent along k for the next $m = 1, \dots, M'$ steps. Thus, in the mth step of the descent along k, the predictor is

$$\mathbf{F}_{k,m} = \mathbf{F}_{k,m-1} + \nu\rho_{k,m}\mathbf{X}_k, \qquad \text{where } \rho_{k,m} = \mathbf{X}_k^T(\mathbf{y} - \mathbf{F}_{k,m-1}).$$

Consider then Algorithm 2 which repeatedly boosts the predictor along the kth direction for a total of M' steps.

The following result states a closed form solution for the m-step predictor of Algorithm 2 and will be crucial to our characterization of L_2Boosting.

THEOREM 1. *$\mathbf{F}_{k,m} = \mathbf{F}_{k,0} + \nu_m\rho_{k,1}\mathbf{X}_k$, where $\nu_m = 1 - (1 - \nu)^m$ and $\rho_{k,1} = \mathbf{X}_k^T(\mathbf{y} - \mathbf{F}_{k,0})$.*

Algorithm 2 L_2Boosting (Fixed direction, k)

```
$\mathbf{F}_{k,0} = \mathbf{F}_{m'}$
for $m = 1$ to $M'$ do
    $\mathbf{F}_{k,m} = \mathbf{F}_{k,m-1} + \nu\rho_{k,m}\mathbf{X}_k$, where $\rho_{k,m} = \mathbf{X}_k^T(\mathbf{y} - \mathbf{F}_{k,m-1})$
end for
```

Theorem 1 shows that taking a single step with learning parameter ν_m yields the same limit as taking m steps with the smaller learning parameter ν. The result also sheds insight into how ν slows the descent relative to stagewise regression. Notice that the m-step predictor can be written as

$$\mathbf{F}_{k,m} = \underbrace{\mathbf{F}_{k,0} + \rho_{k,1}\mathbf{X}_k}_{\text{stagewise}} - \underbrace{(1-\nu)^m \rho_{k,1}\mathbf{X}_k}_{\text{slow learning}}.$$

The first term on the right is the predictor from a greedy stagewise step, while the second term represents the effect of slow-learning. This latter term is what slows the descent relative to a greedy step. When $m \to \infty$ this term vanishes, and we end up with stagewise fitting, $\nu = 1$.

2.1. *Directional change in the descent.* Theorem 1 shows how to take a large boosting step in place of many small steps, but it does not indicate how many steps must be taken along k before a new variable enters the solution path. If this were known, then the entire k-descent could be characterized in terms of a single step.

To determine the descent length, suppose that L_2Boosting has descended along k for a total of m steps. At step $m+1$ the algorithm must decide whether to continue along k or to select a new direction j. To determine when to switch directions, we introduce the following definition.

DEFINITION 2. A direction j is said to be more favorable than k at step $m+1$ if $|\rho_{k,m}| \geq |\rho_{j,m}|$ and $|\rho_{k,m+1}| < |\rho_{j,m+1}|$. Thus, if j is more favorable at $m+1$, the descent switches to j for step $m+1$.

To determine when j becomes more favorable, it will be useful to have a closed form expression for $\rho_{k,m+1}$ and $\rho_{j,m+1}$. By Theorem 1,

$$\begin{aligned}\rho_{j,m+1} &= \mathbf{X}_j^T(\mathbf{y} - \mathbf{F}_{k,m})\\ &= \mathbf{X}_j^T[(\mathbf{y} - \mathbf{F}_{k,0}) - \nu_m\rho_{k,1}\mathbf{X}_k]\\ &= \rho_{j,1} - \nu_m\rho_{k,1}R_{j,k},\end{aligned}$$

where $R_{j,k} = \mathbf{X}_j^T\mathbf{X}_k$. Setting $j = k$ yields $\rho_{k,m+1} = (1-\nu)^m\rho_{k,1}$. Therefore, $|\rho_{k,m+1}| < |\rho_{j,m+1}|$ if and only if

$$(1-\nu)^{2m}\rho_{k,1}^2 < (\rho_{j,1} - \nu_m\rho_{k,1}R_{j,k})^2.$$

Dividing throughout by $\rho_{k,1}$, with a little bit of rearrangement, this becomes

$$(1-\nu)^{2m} < [(1-\nu)^m R_{j,k} + (d_{j,k} - R_{j,k})]^2, \tag{2.1}$$

where $d_{j,k} = \rho_{j,1}/\rho_{k,1}$. Notice importantly that $|d_{j,k}| \le 1$ because k is the direction with maximal gradient-correlation at the start of the descent. It is also useful to keep in mind that $R_{j,k}$ is the sample correlation of $\mathbf{X}_j$ and $\mathbf{X}_k$ due to (1.3), and thus $|R_{j,k}| \le 1$. The following result states the number of steps taken along k before j becomes more favorable.

THEOREM 2. *The number of steps $m_{j,k}$ taken along k so that j becomes more favorable than k at $m_{j,k}+1$ is the largest integer m such that*

$$(1-\nu)^{m-1} \ge \frac{|d_{j,k} - R_{j,k}|}{1 - R_{j,k}\operatorname{sgn}(d_{j,k} - R_{j,k})}. \tag{2.2}$$

It follows that for $0 < \nu < 1$

$$m_{j,k} = \operatorname{floor}\left[1 + \frac{\log|d_{j,k} - R_{j,k}| - \log(1 - R_{j,k}\operatorname{sgn}(d_{j,k} - R_{j,k}))}{\log(1-\nu)}\right], \tag{2.3}$$

where $\operatorname{floor}(z)$ *is the largest integer less than or equal to* z.

REMARK 1. In particular, notice that $m_{j,k} = \infty$ when $d_{j,k} = R_{j,k}$ [adopting the standard convention that $\operatorname{sgn}(0) = 0$ and assuming that $\nu < 1$]. We call $d_{j,k} = R_{j,k}$ the repressible condition. Section 4 will show that repressibility plays a key role in L_2Boosting's behavior in correlated settings.

REMARK 2. When $\nu = 1$ we obtain $m_{j,k} = 1$ from (2.2) which corresponds to greedy stagewise fitting. Because this makes the $\nu = 1$ case uninteresting, we shall hereafter assume that $0 < \nu < 1$.

2.2. *Defining the solution path.* Theorem 2 immediately shows that the problem of determining the next variable to enter the solution path can be recast as finding the direction requiring the fewest number of steps $m_{j,k}$ to favorability. When combined with Theorem 1, this characterizes the entire descent and can be used to characterize L_2Boosting's solution path.

As before, assume that k corresponds to the first critical direction of the path, that is, $l_1 = k$. By Theorem 2, L_2Boosting descends along k for a total of $S_1 = L_1$ steps, where $L_1 = m_{l_2,k}$ and l_2 is the coordinate requiring the smallest number of steps to become more favorable than k. By Theorem 1, the predictor at step S_1 is

$$\mathbf{F}_{S_1} = \mathbf{F}_0 + \nu_{L_1}\rho_{l_1}^{(1)}\mathbf{X}_{l_1}, \qquad \text{where } \rho_{l_1}^{(1)} = \mathbf{X}_{l_1}^T(\mathbf{y} - \mathbf{F}_0).$$

Applying Theorem 1 once again, but now using a descent along l_2 initialized at $\mathbf{F}_{S_1}$, and continuing this argument recursively, as well as using the representation for the number of steps from Theorem 2, yields Theorem 3, which presents a recursive description of L_2Boosting's solution path.

THEOREM 3. $\mathbf{F}_{S_r} = \mathbf{F}_{S_{r-1}} + \nu_{L_r}\rho_{l_r}^{(r)}\mathbf{X}_{l_r}$, *where* $\{(l_r, L_r, S_r, \rho_{l_r}^{(r)})\}_1^{M^*}$ *are determined recursively from*

$$l_1 = \arg\max_{1\le j\le p} |\mathbf{X}_j^T(\mathbf{y} - \mathbf{F}_0)|, \qquad l_{r+1} = \arg\max_{j\ne l_r} |\rho_j^{(r+1)}|,$$

$$M_j^{(r)} = \operatorname{floor}\left[1 + \frac{\log|D_j^{(r)} - R_{j,l_r}| - \log(1 - R_{j,l_r}\operatorname{sgn}(D_j^{(r)} - R_{j,l_r}))}{\log(1-\nu)}\right],$$

$$L_r = M_{l_{r+1}}^{(r)}, \qquad S_r = S_{r-1} + L_r, \qquad S_0 = 0,$$

$$D_j^{(r)} = \frac{\rho_j^{(r)}}{\rho_{l_r}^{(r)}}, \qquad \rho_j^{(r+1)} = \mathbf{X}_j^T(\mathbf{y} - \mathbf{F}_{S_r}) = \rho_j^{(r)} - \nu_{L_r}\rho_{l_r}^{(r)} R_{j,l_r}.$$

REMARK 3. A technical issue arises in Theorem 3 when $M_j^{(r)}$ is not unique. Non-uniqueness can occur due to rounding which is caused by the floor function used in the definition of $m_{j,k}$. This is why line 1 selects the next critical value, l_{r+1}, by maximizing the absolute gradient-correlation $|\rho_j^{(r+1)}|$ and not by minimizing the step number $M_j^{(r)}$. This definition for l_{r+1} is equivalent to the two-step solution

$$l_{r+1} \leftarrow \arg\max_{j\in l_{r+1}} |\rho_j^{(r+1)}|, \qquad \text{where } l_{r+1} = \arg\min_{j\ne l_r}\{M_j^{(r)}\}.$$

REMARK 4. Another technical issue arises when there is a tie in the absolute gradient-correlation. In line 3 of Algorithm 1 it may be possible for two coordinates, say j and k, to have equal gradient-correlations at step $m > 1$. Theorem 3 implicitly deals with such ties due to Definition 2. For example, suppose that the first $m-1$ steps are along k with the tie occurring at step m. In the language of Theorem 2, because j becomes more favorable than k at $m+1$, where $m = m_{j,k}$, we have

$$|\rho_{j,m-1}| < |\rho_{k,m-1}|, \qquad |\rho_{j,m}| = |\rho_{k,m}|, \qquad |\rho_{j,m+1}| > |\rho_{k,m+1}|.$$

In this example, Theorem 3 resolves the tie at m by continuing to descend along k, then switching to j at step $m+1$. Although Algorithm 1 does not explicitly address this issue, the potential discrepancy is minor because such ties should rarely occur in practice. This is because for $|\rho_{j,m}| = |\rho_{k,m}|$ to hold, the value inside the floor function of (2.3) used to define $m_{j,k}$ must be an integer (a careful analysis of the proof of Theorem 2 shows why). A tie can occur only when this value is an integer which is numerically unlikely to occur.

REMARK 5. Theorem 3 immediately yields a recursive solution for the coefficient vector, $\boldsymbol{\beta}$. The solution path for $\boldsymbol{\beta}$ is the piecewise solution

$$\boldsymbol{\beta}^{(r)} = \boldsymbol{\beta}^{(r-1)} + \nu_{L_r}\rho_{l_r}^{(r)}\mathbf{1}_{l_r}, \qquad \boldsymbol{\beta}^{(0)} = \mathbf{0},$$

where $\mathbf{1}_{l_r} \in \mathbb{R}^p$ is the vector with one in coordinate l_r and zero elsewhere.

Algorithm 3 L_2Boosting (Solution path)

```
$\mathbf{F}_0 = \mathbf{0}$; $S_0 = 0$; $l_1 = \arg\max_{1\leq j\leq p}|\mathbf{X}_j^T\mathbf{y}|$
for $r = 1$ to $M^*$ do
    $l_{r+1} = \arg\max_{j\neq l_r}|\rho_j^{(r+1)}|$; $\rho_j^{(r+1)} = \rho_j^{(r)} - \nu_{L_r}\rho_{l_r}^{(r)}R_{j,l_r}$
    $L_r = M_{l_{r+1}}^{(r)}$; $S_r = S_{r-1} + L_r$
    $\mathbf{F}_{S_r} = \mathbf{F}_{S_{r-1}} + \nu_{L_r}\rho_{l_r}^{(r)}\mathbf{X}_{l_r}$
end for
```

2.3. *Illustration*: *Diabetes data.* Aside from the technical issue of ties, Theorem 3 and Algorithm 1 are equivalent. For convenience, we state Theorem 3 in an algorithmic form to facilitate comparison with Algorithm 1; see Algorithm 3. Computationally, Algorithm 3 improves upon Algorithm 1 by avoiding taking many small steps along a given descent. However, the difference is not substantial because the benefits only apply when ν is small, and as we will show later (Section 3), this forces the algorithm to cycle between its variables following the first descent, thus mitigating its ability to take large steps. Thus, strictly speaking, the benefit of Algorithm 3 is confined primarily to the first descent.

To investigate the differences between the two algorithms we analyzed the diabetes data used in Efron et al. (2004). The data consists of $n = 442$ patients in which the response of interest, y, is a quantitative measure of disease progression for a patient. In total there are 64 variables, that includes 10 baseline measurements for each patient, 45 interactions and 9 quadratic terms.

In order to compare results, we translated each iteration, r, used by Algorithm 3 into its corresponding number of steps, m. Thus, while we ran Algorithm 3 for $M^* = 250$ iterations, this translated into $M = 332$ steps. As expected, this difference is primarily due to the first iteration $r = 1$ which took $m = 14$ steps along the first critical direction (first panel of Figure 2; the rug indicates critical points, S_r). There are other instances where Algorithm 3 took more than one step (corresponding to the light grey tick marks on the rug), but these were generally steps of length 2. The standardized gradient-correlation is plotted along the y-axis of the figure. The standardized gradient-correlation for step m was defined as (using the notation of Algorithm 1)

$$\rho_m^* = \frac{\mathbf{X}_{k_m}^T\mathbf{g}_m}{\sqrt{\mathbf{X}_{k_m}^T\mathbf{X}_{k_m}}\sqrt{\mathbf{g}_m^T\mathbf{g}_m}} = \frac{\rho_m}{\sqrt{\mathbf{g}_m^T\mathbf{g}_m}}. \tag{2.4}$$

The middle panel displays the results using Algorithm 1 with $M = 250$ steps. Clearly, the greatest gains from Algorithm 3 occur along the $r = 1$ descent. One can see this most clearly from the last panel which superimposes the first two panels.

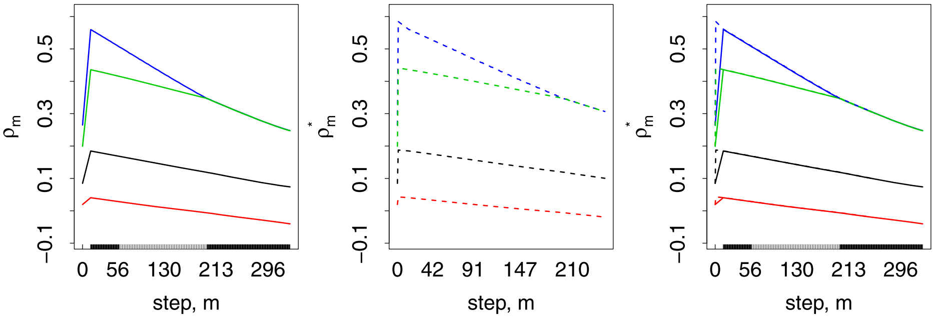

FIG. 2. *L_2Boosting applied to the diabetes data. First two panels display standardized gradient-correlation ρ^*_m against step number m for Algorithms* 3 *and* 1, *respectively. Only coordinates in the solution path are displayed* (*a total of four*). *The third panel superimposes the first two panels. All analyses used $\nu = 0.005$.*

REMARK 6. Note a potential computational optimization exists in Algorithm 3. It is possible to calculate the correlation values only once as each new variable enters the active set, then cache these values for future calculations. Thus, when l_{r+1} is a new variable in the active set, we calculate $(R_{j,l_{r+1}})_{j=1}^p$. The updated gradient-correlation is calculated efficiently by using addition and scalar multiplication using the previous gradient-correlation and the cached correlation coefficients

$$\rho_j^{(r+1)} = \rho_j^{(r)} - \nu_{L_r} \rho_{l_r}^{(r)} R_{j,l_r}.$$

This is in contrast to Algorithm 1 which requires a vector multiplication of dimension p at each step m to update the gradient-correlation: $\rho_m = \mathbf{X}_{k_m}^T \mathbf{g}_m$.

REMARK 7. Above, when we refer to the "active set," we mean the unique set of critical directions in the current solution path. This term will be used repeatedly throughout the paper.

2.4. *Visualizing the solution path.* Throughout the paper we illustrate different ways of utilizing $m_{j,k}$ of Theorem 2 to explore L_2Boosting. So far we have confined the use of Theorem 2 to determining the descent length along a fixed direction, but another interesting application is determining how far a given variable is from the active set. Note that although Theorem 2 was described in terms of an active set of only one coordinate, it applies in general, regardless of the size of the active set. Thus, $m_{j,k}$ can be calculated at any step m to determine the number of steps required for j to become more favorable than the current direction, k. This value represents the distance of j to the solution path and can be used to visualize it.

To demonstrate this, we applied Algorithm 1 to the diabetes data for $M = 10{,}000$ steps and recorded $m_{j,k}$ for each of the $p = 64$ variables. Figure 3 records

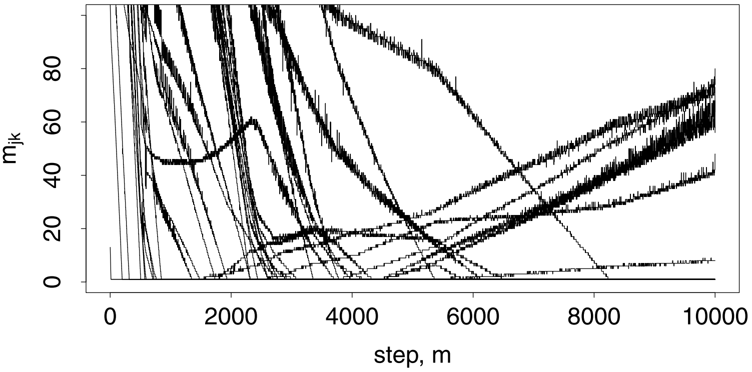

FIG. 3. *Distance $m_{j,k}$ of each variable j to favorability relative to the current descent k* (*results based on Algorithm* 1 *where* $\nu = 0.005$). *For visual clarity the $m_{j,k}$ values have been smoothed using a running median smoother.*

these values. Each "jagged path" in the figure is the trace over the 10,000 steps for a variable j. Each point on the path equals the number of steps $m_{j,k}$ to favorability relative to the current descent $k \neq j$. The patterns are quite interesting. The top variables have $m_{j,k}$ values which quickly drop within the first 1000 steps. Another group of variables have values which take much longer to drop, doing so somewhere between 2000 to 4000 steps, but then increase almost immediately. These variables enter the solution path but then quickly become unattractive regardless of the descent direction.

It has become popular to visualize the solution path of forward stagewise algorithms by plotting their gradient-correlation paths and/or their coefficient paths. Figure 3 is a similar tool. A unique feature of $m_{j,k}$ is that it depends not only on the gradient-correlation (via $d_{j,k}$), but also the correlation in the x-variables (via $R_{j,k}$) and the learning parameter ν. In this manner, Figure 3 offers a new tool for understanding and exploring such algorithms.

3. Cycling behavior.

It has been widely observed that decreasing the regularization parameter slows the convergence of stagewise descent algorithms. Efron et al. (2004) showed that the FS_ε algorithm tracks the equiangular direction of the LAR path for arbitrarily small ε. To achieve what LAR does in a single step, the FS_ε algorithm may require thousands of small steps in a direction tightly clustered around the equiangular vector, eventually ending up at nearly the same point as LAR.

We show that L_2Boosting exhibits this same phenomenon. We do so by describing this property as an active set cycling phenomenon. Using results from the earlier fixed descent analysis, we show in the case of an active set of two variables that L_2Boosting systematically switches (cycles) between its two variables when ν is small. For an arbitrarily small ν this forces the absolute gradient-correlations for the active set variables to be nearly equal. This point of equality represents a singularity point that triggers a near-perpetual deterministic cycle between the variables, ending only when a new variable enters the active set with nearly the same absolute gradient-correlation.

3.1. *L_2Boosting's gradient equality point.* Our insight will come from looking at Theorem 2 in more depth. As before, assume the algorithm has been initialized so that k is the first critical step. Previously the descent along k was described in terms of steps, but this can be equivalently expressed in units of the "step size" taken. Define

$$\nu_{j,k} = \nu_{m_{j,k}} = 1 - (1-\nu)^{m_{j,k}}.$$

Recall that Theorem 1 showed that a single step along k with ν replaced with $\nu_{j,k}$ yields the same limit as $m_{j,k}$ steps along k using ν. We call $\nu_{j,k}$ the step size taken along k. Because j becomes more favorable than k at $m_{j,k}+1$, the gradient following a step size of $\nu_{j,k}$ along k satisfies

$$|\mathbf{X}_j^T(\mathbf{y} - \mathbf{F}_0 - \nu_{j,k}\rho_{k,1}\mathbf{X}_k)| < |\mathbf{X}_k^T(\mathbf{y} - \mathbf{F}_0 - \nu_{j,k}\rho_{k,1}\mathbf{X}_k)|. \tag{3.1}$$

This applies to all coordinates $j \neq k$, and in particular holds for the second critical direction, l_2, which rephrased in terms of step size, is the smallest $\nu_{j,k}$ value,

$$l_2 = \arg\min_{j\neq k}\{\nu_{j,k}\}.$$

Although inequality (3.1) is strict, it becomes arbitrarily close to equality with shrinking ν. With a little bit of rearranging, (2.2) implies that

$$\hat{\nu}_j < \nu_{j,k}, \qquad \text{where } \hat{\nu}_j = 1 - \frac{|d_{j,k} - R_{j,k}|}{1 - R_{j,k}\operatorname{sgn}(d_{j,k} - R_{j,k})}. \tag{3.2}$$

We will show $\hat{\nu}_j$ is the step size making the absolute gradient-correlation between j and k equal

$$|\mathbf{X}_j^T(\mathbf{y} - \mathbf{F}_0 - \hat{\nu}_j\rho_{k,1}\mathbf{X}_k)| = |\mathbf{X}_k^T(\mathbf{y} - \mathbf{F}_0 - \hat{\nu}_j\rho_{k,1}\mathbf{X}_k)|. \tag{3.3}$$

The next theorem shows that $\nu_{l_2,k}$ converges to the smallest $\hat{\nu}_j$ satisfying (3.3); thus, (3.1) becomes an equality in the limit. For convenience, we define $\nu_{j,k}^- = \nu_{m_{j,k}-1}$.

THEOREM 4. *Let $\hat{\rho}_j = \mathbf{X}_j^T(\mathbf{y} - \mathbf{F}_0 - \hat{\nu}_j\rho_{k,1}\mathbf{X}_k)$. Then $|\hat{\rho}_j| = |\hat{\rho}_k|$. Furthermore, if $l^* = \arg\min_{j\neq k}\{\hat{\nu}_j\}$ and $\hat{\nu} = \hat{\nu}_{l^*}$, then $\nu_{l_2,k}^- \leq \hat{\nu} < \nu_{l_2,k}$ and $\nu_{l_2,k} \to \hat{\nu}$ as $\nu \to 0$.*

Therefore, for arbitrarily small ν, $\nu_{l_2,k} \asymp \hat{\nu}$ and k and l_2 will have near-equal absolute gradient-correlations. This latter property triggers two-cycling. To see why, let us assume for the moment that the active set variables have equal absolute gradient-correlations. Then by a direct application of Theorem 2, one can show that the number of steps taken along l_2 before k becomes more favorable is $m = 1$. Thus, following the descent along k, the algorithm switches to l_2, but then immediately switches back to k. If ν is small enough, this process is repeated, setting off a two-cycling pattern.

The next result is a formal statement of these arguments. Define

$$d_{j,k}^{(m)} = \frac{\rho_{j,m}}{\rho_{k,m}}, \qquad \text{where } \rho_{l,m} = \mathbf{X}_l^T(\mathbf{y} - \mathbf{F}_{m-1}),\ 1 \le l \le p.$$

For notational convenience, let $j = l_2$ and $m = m_{j,k}$. For technical reasons we shall assume $d_{j,k}^{(m)} \neq R_{j,k}$. Recall Remark 1 showed that $d_{j,k}^{(m)} = R_{j,k}$, the repressible condition, yields an infinite number of steps to favorability. Thus, for k to be even eligible for favorability we must have $d_{j,k}^{(m)} \neq R_{j,k}$.

THEOREM 5. *If the first two critical directions are* (k, j) *and* $\nu_{j,k} = \hat{\nu}_j$, *then* k *is favored over* j *for the next step after* j *if* $d_{j,k}^{(m)} \neq R_{j,k}$.

Theorem 5 assumes that $\nu_{j,k} = \hat{\nu}_j$. While this only holds in the limit, the two values should be nearly equal for arbitrarily small ν, and thus the assumption is reasonable. Notice also that Theorem 5 only shows that k is more favorable than j, and not that the algorithm switches to k. However, we can see that this must be the case. For arbitrarily small ν, k's gradient-correlation should be nearly equal to j's, and by definition, j has maximal absolute gradient-correlation along the second descent.

Indeed, the following result shows that the absolute gradient-correlations for k and j can be made arbitrarily close for small enough ν for any step $r \ge 1$ following the descent along k. The result also shows that the sign of the gradient-correlation is preserved when ν is arbitrarily small, a fact that we shall use later.

THEOREM 6. $\rho_{j,m+r}/\rho_{k,m+r} \to \operatorname{sgn}(\hat{\rho}_j)/\operatorname{sgn}(\hat{\rho}_k)$ *as* $\nu \to 0$ *for each* $r \ge 1$.

Combining Theorems 5 and 6, we see that if ν is small enough, the first three critical directions of the path must be (k, j, k) with critical points $(m, m + 1, m + 2)$. And once the descent switches back to k, it is clear from the same argument that the next critical direction, l_4, will be j, and so forth.

3.2. *Illustration of two-cycling.* We present a numerical example demonstrating two-cycling. For our example, we simulated data according to

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{I}),$$

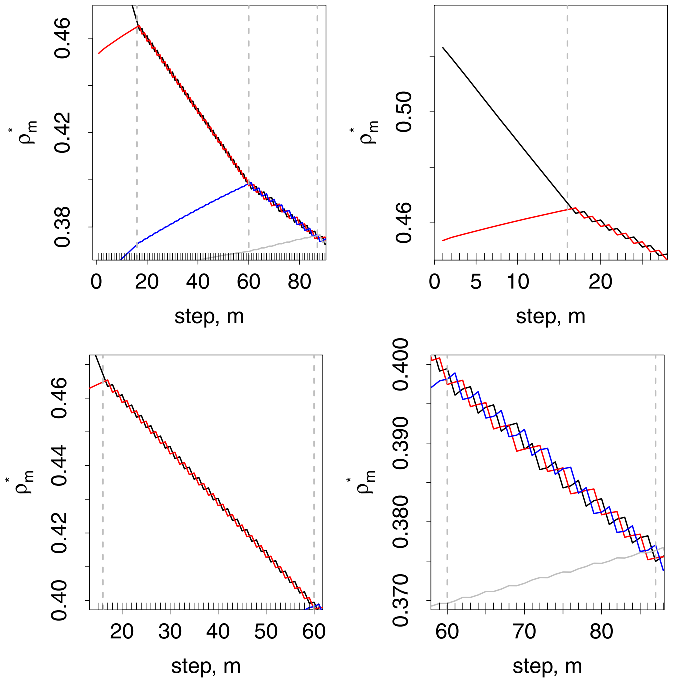

FIG. 4. *Standardized gradient-correlation path for* $\nu = 0.01$. *Top left panel details the path through the first three active variables, the remaining panels detail each active variable descent.*

where $n = 100$, and $p = 40$. The first 10 coordinates of $\boldsymbol{\beta}$ were set to 5, with the remaining coordinates set to 0. The design matrix $\mathbf{X}$ was simulated by drawing its entries independently from a standard normal distribution.

Figure 4 plots the standardized gradient-correlations (2.4) from Algorithm 3 using $\nu = 0.01$. As done earlier, we have converted iterations r into step numbers m along the x-axis. The plots show the behavior of each coordinate within an active set descent. The rug marks show each step m for clarity, and dashed vertical lines indicate the step $m_{j,k}$ where the next step adds a new critical direction to the solution path. The top left panel shows the complete descent along the first three active variables. The remaining panels detail the coordinate behavior as the active set increases from one to three coordinates.

The top right panel shows repeated selection of the l_1 direction shown in black. The last step along l_1 occurs at $m_{j,k}$ marked with the vertical dashed line, where the next step is along the l_2 direction shown in red. This point marks the beginning of the two-cycling behavior, which continues in the lower left panel. At each step, the algorithm systematically switches between the l_1 and l_2 directions, until an additional direction becomes more favorable. The cycling pattern is $\{l_1, l_2, l_1, l_2, \ldots\}$. The lower right panel demonstrates three-cycling behavior. Here it is instructive to note that the order of selection within three-cycling is nondeterministic. In this panel the order starts as $\{l_3, l_2, l_1, \ldots\}$, but changes near $m = 70$ to $\{\ldots, l_3, l_1, l_2, \ldots\}$. As discussed later, nondeterministic cycling patterns are typical behavior of higher order cycling (active sets of size greater than two).

3.3. *The limiting path.* Here we provide a formal limiting result of two-cycling. The result can be viewed as the analog of Theorem 4 when the active set involves two variables. Using a slightly modified version of L_2Boosting we show that for arbitrarily small ν, if the algorithm cycles between its two active variables, it does so until a new variable enters the active set with the same absolute gradient-correlation.

Assume the active set is $\mathcal{A} = \{k, j\}$ and that k and j are cycling according to $(k, j, k, j, \ldots)$. The m-step predictor for $m = 1, \ldots, M$ is

$$\mathbf{F}_m = \begin{cases} \mathbf{F}_{m-1} + \nu\rho_{k,m}\mathbf{X}_k, & \text{if } m \text{ is odd}, \\ \mathbf{F}_{m-1} + \nu\rho_{j,m}\mathbf{X}_j, & \text{if } m \text{ is even}, \end{cases} \tag{3.4}$$

where $\rho_{l,m} = \mathbf{X}_l^T(\mathbf{y} - \mathbf{F}_{m-1})$. The cycling pattern (3.4) is assumed to persist for a minimum length of $M \geq 3$.

It will simplify matters if the cycling is assumed to be initialized with strict equality of the gradient correlations: $|\rho_{k,1}| = |\rho_{j,1}|$. With an arbitrarily small ν, this will force near equal absolute gradient-correlations at each step and by Theorem 6 will preserve the sign of the gradient-correlation. We assume

$$\frac{\rho_{j,m}}{\rho_{k,m}} = \frac{\operatorname{sgn}(\rho_{j,1})}{\operatorname{sgn}(\rho_{k,1})} \qquad \text{for } m \geq 1.$$

It should be emphasized that the above assumptions represent a simplified version of L_2Boosting. In practice, we would have

$$\rho_{j,m} = s\rho_{k,m} + O(\nu),$$

where $s = \operatorname{sgn}(\rho_{j,1})/\operatorname{sgn}(\rho_{k,1})$. However, for convenience we will not concern ourselves with this level of detail here. Readers can consult Ehrlinger (2011) for a more refined analysis.

One way to ensure $|\rho_{k,1}| = |\rho_{j,1}|$ is to initialize the algorithm with the limiting predictor $\mathbf{F}_0 + \hat{\nu}_j\rho_{k,1}\mathbf{X}_k$ of Theorem 4 obtained by letting $\nu \to 0$ along the k-descent. With a slight abuse of notation denote this initial estimator by $\mathbf{F}_0$. However, the fact that this specific $\mathbf{F}_0$ is used does not play a direct role in the results. Under the above assumptions, the following closed form expression for the m-step predictor under two-cycling holds.

THEOREM 7. *Assume that $\rho_{j,m} = s\rho_{k,m}$ for $m \geq 1$. If $d_{j,k} \neq R_{j,k}$, then for any $0 < \nu < 1/2$ satisfying $1 + sR_{j,k} > \nu R_{j,k}^2$, we have for each $m \geq 1$,*

$$\mathbf{F}_m = \begin{cases} \mathbf{F}_0 + V_{m+1}\rho_{k,1}\left[\mathbf{X}_k + \dfrac{V_{m-1}}{V_{m+1}}(s - \nu R_{j,k})\mathbf{X}_j\right], & \textit{if } m \textit{ is odd}, \\ \mathbf{F}_0 + V_m\rho_{k,1}[\mathbf{X}_k + (s - \nu R_{j,k})\mathbf{X}_j], & \textit{if } m \textit{ is even}, \end{cases}$$

where $V_m = \nu\nu_{\mathcal{A}}^{-1}[1-(1-\nu_{\mathcal{A}})^{m/2}]$ *and* $\nu_{\mathcal{A}} = \nu(1+sR_{j,k}-\nu R_{j,k}^2)$. *Note that* $0<\nu_{\mathcal{A}}<1$ *under the asserted conditions.*

To determine the above limit requires first determining when a new direction $l \notin \mathcal{A}$ becomes more favorable. For l to be more favorable at $m+1$, we must have $|\rho_{j,m+1}| < |\rho_{l,m+1}|$ when m is odd, or $|\rho_{k,m+1}| < |\rho_{l,m+1}|$ when m is even. The following result determines the number of steps to favorability. For simplicity only the case when m is odd is considered, but this does not affect the limiting result.

THEOREM 8. *Assume the same conditions as Theorem 7. Then l becomes more favorable than j at step $m+1$ where m is the largest odd integer $m \geq 3$ such that*

$$(1-\nu_{\mathcal{A}})^{(m-1)/2} \geq \frac{|d_{l,k}-R_{j,k,l}|}{1-R_{j,k,l}\operatorname{sgn}(d_{l,k}-R_{j,k,l})} \tag{3.5}$$

where $d_{l,k} = \rho_{l,1}/\rho_{k,1}$ *and*

$$R_{j,k,l} = \frac{R_{l,k}+(s-\nu R_{j,k})R_{l,j}}{1+sR_{j,k}-\nu R_{j,k}^2}.$$

Clearly (3.5) shares common features with (2.2). This is no coincidence. The bounds are similar in nature because both are derived by seeking the point where the absolute gradient-correlation between sets of variables are equal. In the case of two-cycling, this is the singularity point where k, j and l are all equivalent in terms of absolute gradient-correlation. The following result states the limit of the predictor under two-cycling.

THEOREM 9. *Under the conditions of Theorem 7, the limit of* $\mathbf{F}_m$ *as* $\nu \to 0$ *at the next critical direction* l^* *equals*

$$\hat{\mathbf{F}} = \mathbf{F}_0 + \hat{\nu}\rho_{k,1}[\mathbf{X}_k + s\mathbf{X}_j],$$

where $l^* = \arg\min_{l\notin\mathcal{A}}\{\hat{\nu}_l\}$, $\hat{\nu} = \hat{\nu}_{l^*}$,

$$\hat{\nu}_l = \left(1 - \frac{|d_{l,k}-\hat{R}_{j,k,l}|}{1-\hat{R}_{j,k,l}\operatorname{sgn}(d_{l,k}-\hat{R}_{j,k,l})}\right)(1+sR_{j,k})^{-1}, \tag{3.6}$$

and $\hat{R}_{j,k,l} = (R_{l,k}+sR_{l,j})/(1+sR_{j,k})$. *Furthermore,* $|\hat{\rho}_{l^*}| = |\hat{\rho}_k| = |\hat{\rho}_j|$, *where for each* l, $\hat{\rho}_l = \mathbf{X}_l^T(\mathbf{y}-\hat{\mathbf{F}})$.

This shows that the predictor moves along the combined direction $\mathbf{X}_k + s\mathbf{X}_j$ taking a step size $\hat{\nu}$ that makes the absolute gradient-correlation for l^* equal to that of the active set $\mathcal{A} = \{k, j\}$. Theorem 9 is a direct analog of Theorem 4 to two-cycling.

Not surprisingly, one can easily show that this limit coincides with the LAR solution. To show this, we rewrite $\hat{\mathbf{F}}$ in a form comparable to LAR,

$$\hat{\mathbf{F}} = \mathbf{F}_0 + \hat{\nu}|\rho_{k,1}|[\operatorname{sgn}(\rho_{k,1})\mathbf{X}_k + \operatorname{sgn}(\rho_{j,1})\mathbf{X}_j].$$

Recall that LAR moves the shortest distance along the equiangular vector defined by the current active set until a new variable with equal absolute gradient-correlation is reached. The term in square brackets above is proportional to this equiangular vector. Thus, since $\hat{\mathbf{F}}$ is obtained by moving the shortest distance along the equiangular vector such that $\{j, k, l^*\}$ have equal absolute gradient-correlation, $\hat{\mathbf{F}}$ must be identical to the LAR solution.

3.4. *General cycling.* Analysis of cycling in the general case where the active set $\mathcal{A} = \{k_i\}_{i=1}^d$ is comprised of $d \geq 2$ variables is more complex. In two-cycling we observed cycling patterns of the form $(l_1, l_2, l_1, l_2, \ldots)$, but when $d > 2$, L_2Boosting's cycling patterns are often observed to be nondeterministic with no discernible pattern in the order of selected critical directions. Moreover, one often observes some coordinates being selected more frequently than others.

A study of d-cycling has been given by Ehrlinger (2011). However, the analysis assumes deterministic cycling of the form

$$(l_1, l_2, \ldots, l_d, l_{d+1}, \ldots) = (k_1, k_2, \ldots, k_d, k_1, \ldots),$$

which is the natural extension of the two-cycling just studied. To accommodate this framework, a modified L_2Boosting procedure involving coordinate-dependent step sizes was used. This models L_2Boosting's cycling tendency of selecting some coordinates more frequently by using the size of a step to dictate the relative frequency of selection. Under constraints to the coordinate step sizes, equivalent to solving a system of linear equations defining the equiangular vector used by LAR, it was shown that the modified L_2Boosting procedure yields the LAR solution in the limit. Interested readers should consult Ehrlinger (2011) for details.

4. Repressibility affects variable selection in correlated settings.

Now we turn our attention to the issue of correlation. We have shown that regardless of the size of the active set a new direction j becomes more favorable than the current direction k at step $m_{j,k} + 1$ where $m_{j,k}$ is the smallest integer value satisfying

$$1 - \frac{|d_{j,k} - R_{j,k}|}{1 - R_{j,k} \operatorname{sgn}(d_{j,k} - R_{j,k})} < 1 - (1 - \nu)^{m_{j,k}}. \tag{4.1}$$

Using our previous notation, let $\hat{\nu}_j$ and $\nu_{j,k}$ denote the left and right-hand sides of the above inequality, respectively.

Generally, large values of $m_{j,k}$ are designed to hinder noninformative variables from entering the solution path. If j requires a large number of steps to become favorable, it is noninformative relative to the current gradient and therefore unattractive as a candidate. Surprisingly, however, such an interpretation does not always apply in correlated problems. There are situations where j is informative, but $m_{j,k}$ can be artificially large due to correlation.

To see why, suppose that j is an informative variable with a relatively large value of $d_{j,k}$. Now, if j and k are correlated, so much so that $R_{j,k} \approx d_{j,k}$, then $|d_{j,k} - R_{j,k}| \approx 0$. Hence, $m_{j,k} \approx \infty$ and $\nu_{j,k} \approx 1$ due to (4.1). Thus, even though j is promising with a large gradient-correlation, it is unlikely to be selected because of its high correlation with k.

The problem is that j becomes an unlikely candidate for selection when $d_{j,k}$ is close to $R_{j,k}$. In fact, $m_{j,k} = \infty$ when $d_{j,k} = R_{j,k}$ so that j can never become more favorable than k when the two values are equal. We have already discussed the condition $d_{j,k} = R_{j,k}$ several times now, and have referred to it as the *repressible condition*. Repressibility plays an important role in correlated settings. We distinguish between two types of repressibility: weak and strong repressibility. Weak repressibility occurs in the trivial case when $|R_{j,k}| = 1$. Weak repressibility implies that $|d_{j,k}| = |R_{j,k}| = 1$. Hence the gradient-correlation for j and k are equal in absolute value and j, and k are perfectly correlated. This trivial case simply reflects a numerical issue arising from the redundancy of the j and k columns of the $\mathbf{X}$ design matrix. The stronger notion of repressibility, which we refer to as strong repressibility, is required to address the nontrivial case $|R_{j,k}| \neq 1$ in which j is repressed without being perfectly correlated with k. The following definition summarizes these ideas.

DEFINITION 3. We say j has the strong repressible condition if $d_{j,k} = R_{j,k}$ and $|R_{j,k}| < 1$. We say that j is (strongly) repressed by k when this happens. On the other hand, j has the weak repressible condition if j and k are perfectly correlated ($|R_{j,k}| = 1$) and $d_{j,k} = R_{j,k}$.

4.1. *An illustrative example.* We present a numerical example of how repressibility can hinder variables from being selected. For our illustration we use example (d) of Section 5 from Zou and Hastie (2005). The data was simulated according to

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \sigma\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \mathbf{I}),$$

where $n = 100$, $p = 40$ and $\sigma = 15$. The first 15 coordinates of $\boldsymbol{\beta}$ were set to 3; all other coordinates were 0. The design matrix $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_{40}]_{100\times 40}$ was simulated according to

$$\begin{aligned}
&\mathbf{X}_j = \mathbf{Z}_1 + \tau\boldsymbol{\varepsilon}_j, \qquad j = 1, \ldots, 5, \\
&\mathbf{X}_j = \mathbf{Z}_2 + \tau\boldsymbol{\varepsilon}_j, \qquad j = 6, \ldots, 10, \\
&\mathbf{X}_j = \mathbf{Z}_3 + \tau\boldsymbol{\varepsilon}_j, \qquad j = 11, \ldots, 15, \\
&\mathbf{X}_j = \boldsymbol{\varepsilon}_j, \qquad j > 15,
\end{aligned} \tag{4.2}$$

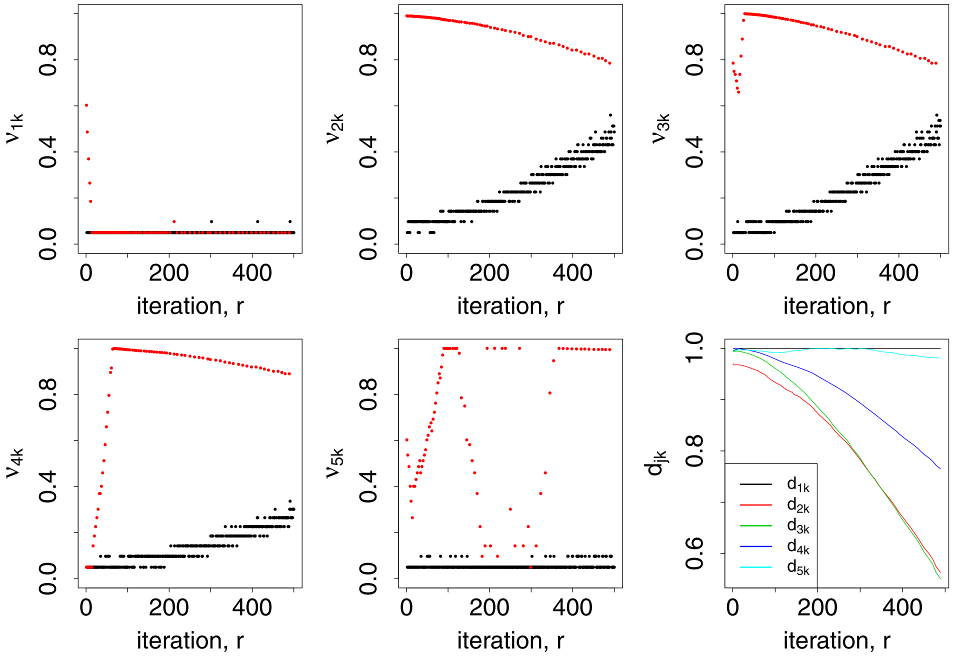

FIG. 5. *First* 5 *panels display* $(\nu_{j,k})_{j=1}^5$ *for the first* 5 *coefficients from simulation* (4.2): *red points are iterations* r *where the descent direction* $k \in \{1, \dots, 5\}$. *Variables* 2 *and* 3 *are never selected due to their excessively large* $\nu_{j,k}$ *step sizes*: *an artifact of the correlation between the* 5 *variables. The last panel* (*bottom right*) *displays* $(d_{j,k})_{j=1}^5$ *for those iterations* r *where* $k \in \{1, 4, 5\}$.

where $(\mathbf{Z}_j)_1^3$ and $(\boldsymbol{\varepsilon}_j)_1^{40}$ were i.i.d. $N(\mathbf{0}, \mathbf{I})$ and $\tau = 0.1$. In this simulation, only coordinates 1 to 5, 6 to 10 and 11 to 15 have nonzero coefficients. These x-variables are uncorrelated across a group, but share the same correlation within a group. Because the within group correlation is high, but less than 1, the simulation is ideal for exploring the effects of strong repressibility.

Figure 5 displays results from fitting Algorithm 3 for $M^* = 500$ iterations with $\nu = 0.05$. The first 5 panels are the values $(\nu_{j,k})_{j=1}^5$ against the iteration $r = 1, \dots, 500$, with points colored in red indicating iterations r where $k \in \{1, \dots, 5\}$ and k is used generically to denote the current descent direction. Notationally, the descent at iteration r is along k for a step size of $\nu_{l,k}$, at which point l becomes more favorable than k and the descent switches to l, the next critical direction. The value plotted, $\nu_{j,k} \le \nu_{l,k}$, is the step size for $j = 1, \dots, 5$.

Whenever the selected coordinate is from the first group of variables (we are referring to the red points) one of the coordinates $j = 1, 4, 5$ achieves a small $\nu_{j,k}$ value. However, coordinates $j = 2$ and $j = 3$ maintain very large values throughout all iterations. This is despite the fact that the two coordinates generally have large values of $d_{j,k}$, especially during the early iterations (see the bottom right panel). This suggests that 1, 4 and 5 become active variables at some point in the solution path, whereas coordinates 2 and 3 are never selected (indeed, this is exactly what happened). We can conclude that coordinates 2 and 3 are being strongly repressed by $k \in \{1, 4, 5\}$. Interestingly, coordinate 4 also appears to be repressed at later iterations of the algorithm. Observe how its $d_{j,k}$ values decrease with increasing r (blue line in bottom right panel), and that its $\nu_{j,k}$ values are only small at earlier iterations. Thus, we can also conclude that coordinates $\{1, 5\}$ eventually repress coordinate 4 as well.

We note that the number of iterations $M^* = 500$ used in the example is not very large, and if L_2Boosting were run for a longer period of time, coordinates 2 and 3 will eventually enter the solution path (panels 2 and 3 of Figure 5 show evidence of this already happening with $\nu_{j,k}$ steadily decreasing as r increases). However, doing so leads to overfitting and poor test-set performance (we provide evidence of this shortly). Using different values of ν also did not resolve the problem. Thus, similar to the lasso, we find that L_2Boosting is unable to select entire groups of correlated variables. Like the lasso this means it also will perform suboptimally in highly correlated settings. In the next section we introduce a simple way of adding L_2-regularization as a way to correct this deficiency.

5. Elastic net boosting. The tendency of the lasso to select only a handful of variables from among a group of correlated variables was noted in Zou and Hastie (2005). To address this deficiency, Zou and Hastie (2005) described an optimization problem different from the classical lasso framework. Rather than relying only on L_1-penalization, they included an additional L_2-regularization parameter designed to encourage a ridge-type grouping effect, and termed the resulting estimator "the elastic net." Specifically, for a fixed $\lambda > 0$ (the ridge parameter) and a fixed $\lambda_0 > 0$ (the lasso parameter), the elastic net was defined as

$$\hat{\boldsymbol{\beta}}_{\text{enet}} = (1+\lambda) \mathop{\arg\min}_{\boldsymbol{\beta} \in \mathbb{R}^p} \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \sum_{k=1}^{p} \beta_k^2 + \lambda_0 \sum_{k=1}^{p} |\beta_k| \right\}. \tag{5.1}$$

To calculate the elastic net, Zou and Hastie (2005) showed that (5.1) could be recast as a lasso optimization problem by replacing the original data with suitably constructed augmented values. They replaced $\mathbf{y}$ ($n \times 1$) and $\mathbf{X}$ ($n \times p$) with augmented values $\mathbf{y}^*$ and $\mathbf{X}^*$, defined as follows:

$$\mathbf{y}^* = \begin{bmatrix} \mathbf{y} \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{(n+p)\times 1}, \qquad \mathbf{X}^* = \frac{1}{\sqrt{1+\lambda}} \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix}_{(n+p)\times p} = [\mathbf{X}_1^*, \ldots, \mathbf{X}_p^*]. \tag{5.2}$$

The elastic net optimization can be written in terms of the augmented data by reparameterizing $\boldsymbol{\beta}$ as $\boldsymbol{\beta}^* = \boldsymbol{\beta}\sqrt{1+\lambda}$. By Lemma 1 of Zou and Hastie (2005), it follows that (5.1) can be expressed as

$$\hat{\boldsymbol{\beta}}_{\text{enet}} = \sqrt{1+\lambda} \mathop{\arg\min}_{\boldsymbol{\beta} \in \mathbb{R}^p} \left\{ \|\mathbf{y}^* - \mathbf{X}^*\boldsymbol{\beta}\|^2 + \frac{\lambda_0}{\sqrt{1+\lambda}} \sum_{k=1}^{p} |\beta_k| \right\},$$

which is an L_1-optimization problem that can be solved using the lasso.

One explanation for why the elastic net is so successful in correlated problems is due to its decorrelation property. Let $R^*_{j,k} = \mathbf{X}^{*T}_j \mathbf{X}^*_k$. Because the data is standardized such that $\mathbf{X}^T_j \mathbf{X}_j = \mathbf{X}^T_k \mathbf{X}_k = 1$ [recall (1.3)], we have

$$R^*_{j,k} = \begin{cases} \dfrac{\mathbf{X}^T_j \mathbf{X}_k}{1+\lambda} = \dfrac{R_{j,k}}{1+\lambda}, & \text{if } j \neq k, \\ \dfrac{\mathbf{X}^T_j \mathbf{X}_j + \lambda}{1+\lambda} = 1, & \text{if } j = k. \end{cases}$$

One can see that λ is a decorrelation parameter, with larger values reducing the correlation between coordinates. Zou and Hastie (2005) argued that this effect promotes a "grouping property" for the elastic net that overcomes the lasso's inability to select groups of correlated variables.

We believe that decorrelation is an important component of the elastic net's success. However, we will argue that in addition to its role in decorrelation, λ has a surprising connection to repressibility that further explains its role in regularizing the elastic net.

The argument for the elastic net follows as a special case (the limit) of a generalized L_2Boosting procedure we refer to as elasticBoost. The elasticBoost algorithm is a modification of L_2Boosting applied to the augmented problem. To implement elasticBoost one runs L_2Boosting on the augmented data (5.2), adding a post-processing step to rescale the coefficient solution path: see Algorithm 4 for a precise description. For arbitrarily small ν, the solution path for elasticBoost approximates the elastic net, but for general $0 < \nu \leq 1$, elasticBoost represents a novel extension of L_2Boosting. We study the general elasticBoost algorithm, for arbitrary $0 < \nu \leq 1$, and present a detailed explanation of how λ imposes L_2-regularization.

5.1. *How λ regularizes the solution path.* To study the effect λ has on elasticBoost's solution path we consider in detail how λ effects $m^*_{j,k}$, the number of steps to favorability [defined as in (2.3) but with $\mathbf{y}$ and $\mathbf{X}$ replaced by their augmented

Algorithm 4 elasticBoost

```
1: Augment the data (5.2). Set $F^*_{0,i} = 0$ for $i = 1, \dots, n + p$.
2: Run Algorithm 3 for $M$ iterations using the augmented data.
3: Let $F^*_{M,i}$ denote the $M$-step predictor (discard $F^*_{M,i}$ for $i > n$). Let $\beta^*_{M,k}$ denote the $M$-step coefficient estimate.
4: Rescale the regression estimates: $\beta_{M,k} = \sqrt{1+\lambda}\beta^*_{M,k}$.
```

values $\mathbf{y}^*$ and $\mathbf{X}^*$]. At initialization, the gradient-correlation for $j \neq k$ is

$$\begin{aligned}\rho^*_{j,1} &= \mathbf{X}^{*T}_j(\mathbf{y}^* - \hat{\mathbf{F}}^*_0)\\ &= \frac{1}{\sqrt{1+\lambda}}\mathbf{X}^T_j\mathbf{y} - \frac{1}{\sqrt{1+\lambda}}\left(\sum_{i=1}^n x_{i,j}F^*_{0,i} + \sqrt{\lambda}F^*_{0,n+j}\right).\end{aligned}$$

In the special case when $F^*_{0,i} = 0$, corresponding to the first descent of the algorithm,

$$\rho^*_{j,1} = \frac{1}{\sqrt{1+\lambda}}\mathbf{X}^T_j\mathbf{y} = \frac{1}{\sqrt{1+\lambda}}\rho_{j,1}.$$

Therefore, $d^*_{j,k} = \rho_{j,1}/\rho_{k,1} = d_{j,k}$, and hence

$$m^*_{j,k} = \mathrm{floor}\left[1 + \frac{\log|d_{j,k} - R^*_{j,k}| - \log(1 - R^*_{j,k}\,\mathrm{sgn}(d_{j,k} - R^*_{j,k}))}{\log(1-\nu)}\right].$$

This equals the number of steps in the original (nonaugmented) problem but where $\mathbf{X}$ is replaced with variables decorrelated by a factor of $\sqrt{1+\lambda}$. For large values of λ this addresses the problem seen in Figure 5. Recall we argued that $m_{j,k}$ can became inflated due to the near equality of $d_{j,k}$ with $R_{j,k}$. However, $R^*_{j,k} = R_{j,k}/\sqrt{1+\lambda}$ shrinks to zero with increasing λ, which keeps $m^*_{j,k}$ from becoming inflated.

This provides one explanation for λ's role in regularization, at least for the case when λ is large. But we now suggest another theory that applies for both small and large λ. We argue that regularization is imposed not just by decorrelation, but through a combination of decorrelation and reversal of repressibility. Thus λ's role is more subtle than our previous argument suggests.

To show this, let us suppose that near-repressibility holds. We assume therefore that $R_{j,k} = d_{j,k}(1+\delta)$ for some small $|\delta| < 1$. Then,

$$\begin{aligned}&\log|d_{j,k} - R^*_{j,k}| - \log\bigl(1 - R^*_{j,k}\,\mathrm{sgn}(d_{j,k} - R^*_{j,k})\bigr)\\ &\quad= \underbrace{\left[\log|d_{j,k}| + \log\left|1 - \frac{1+\delta}{\sqrt{1+\lambda}}\right|\right]}_{\text{Repressibility effect}} \qquad (5.3)\\ &\qquad \underbrace{-\log\left(1 - \frac{R_{j,k}}{\sqrt{1+\lambda}}\,\mathrm{sgn}\left(R_{j,k}\left[\frac{1}{1+\delta} - \frac{1}{\sqrt{1+\lambda}}\right]\right)\right)}_{\text{Decorrelation effect}}.\end{aligned}$$

The first term on the right captures the effect of repressibility. When δ is small, λ plays a crucial role in controlling its size. If $\lambda = 0$, the expression reduces to $\log|d_{j,k}| + \log|\delta|$ which converges to $-\infty$ as $|\delta| \to 0$; thus precluding j from being selected [keep in mind that (5.3) is divided by $\log(1-\nu)$, which is negative;

thus $m^*_{j,k} \to \infty$]. On the other hand, any $\lambda > 0$, even a relatively small value, ensures that the expression remains small even for arbitrarily small δ, thus reversing the effect of repressibility.

The second term on the right of (5.3) is related to decorrelation. If $1 + \lambda > (1 + \delta)^2$ (which holds if λ is large enough when $\delta > 0$, or for all $\lambda > 0$ if $\delta < 0$), the term reduces to

$$-\log\left(1 - \frac{R_{j,k}}{\sqrt{1+\lambda}} \operatorname{sgn}(R_{j,k})\right),$$

which remains bounded when $\lambda > 0$ if $R_{j,k} \to 1$. On the other hand, if $1 + \lambda < (1 + \delta)^2$, the term reduces to

$$-\log\left(1 + \frac{R_{j,k}}{\sqrt{1+\lambda}} \operatorname{sgn}(R_{j,k})\right),$$

which remains bounded if $R_{j,k} \to 1$ and shrinks in absolute size as λ increases.

Taken together, these arguments show λ imposes L_2-regularization through a combination of decorrelation and the reversal of repressibility which applies even when λ is relatively small.

These arguments apply to the first descent. The general case when $F^*_{0,i} \neq 0$ requires a detailed analysis of $d^*_{j,k}$. In general,

$$d^*_{j,k} = \frac{\mathbf{X}_j^T\mathbf{y} - \sum_{i=1}^n x_{i,j}F^*_{0,i} - \sqrt{\lambda}F^*_{0,n+j}}{\mathbf{X}_k^T\mathbf{y} - \sum_{i=1}^n x_{i,k}F^*_{0,i} - \sqrt{\lambda}F^*_{0,n+k}}.$$

We break up the analysis into two cases depending on the size of λ. Suppose first that λ is small. Then

$$d^*_{j,k} \asymp \frac{\mathbf{X}_j^T\mathbf{y} - \sum_{i=1}^n x_{i,j}F^*_{0,i}}{\mathbf{X}_k^T\mathbf{y} - \sum_{i=1}^n x_{i,k}F^*_{0,i}},$$

which is the ratio of gradient correlations based on the original $\mathbf{X}$ without pseudodata. If j is a promising variable, then $d^*_{j,k}$ will be relatively large, and our argument from above applies. On the other hand if λ is large, then the third term in the numerator and the denominator of $d^*_{j,k}$ become the dominating terms and

$$d^*_{j,k} \asymp \frac{F^*_{0,n+j}}{F^*_{0,n+k}}.$$

The growth rate of $F^*_{0,i}$ for the pseudo data is $O(\nu)$ for a group of variables that are actively being explored by the algorithm. Thus $|d^*_{j,k}| \asymp 1$ and our previous argument applies.

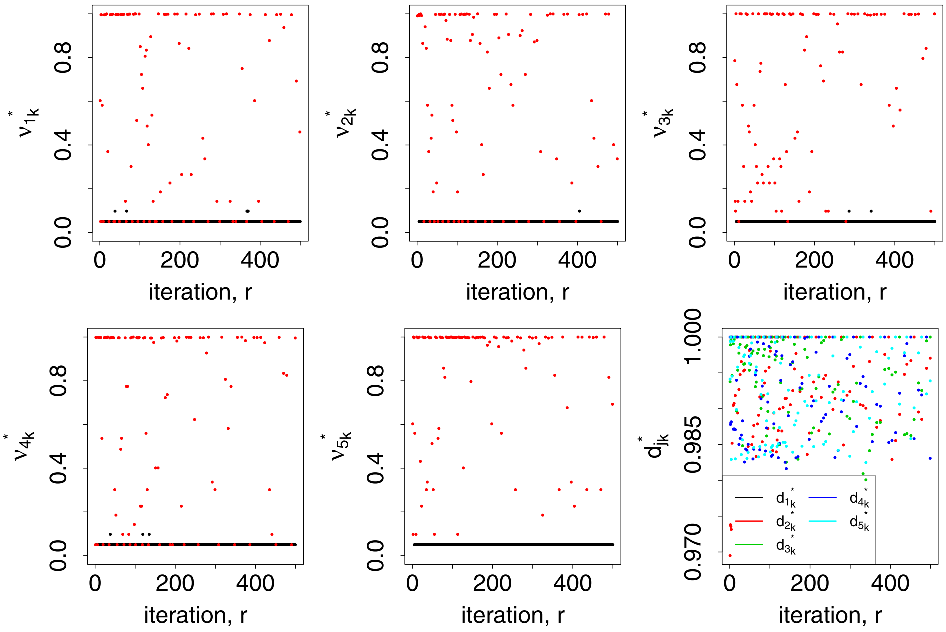

FIG. 6. *elasticBoost applied to simulation* (4.2) (*plots are constructed as in Figure* 5). *Now each of the first* 5 *coordinates are selected and each has* $d^*_{j,k}$ *values near one.*

5.2. *Illustration.* As evidence of this, and to demonstrate the effectiveness of elasticBoost, we re-analyzed (4.2) using Algorithm 4. We used the same parameters as in Figure 5 ($M^* = 500$ and $\nu = 0.05$). We set $\lambda = 0.5$. The results are displayed in Figure 6. In contrast to Figure 5, notice that all 5 of the first group of correlated variables achieve small $\nu^*_{j,k}$ values (and we confirmed that all 5 variables enter the solution path). It is interesting to note that $d^*_{j,k}$ is nearly 1 for each of these variables.

To compare L_2Boosting and elasticBoost more evenly, we used 10-fold cross-validation to determine the optimal number of iterations (for elasticBoost, we used doubly-optimized cross-validation to determine both the optimal number of iterations and the optimal λ value; the latter was found to equal $\lambda = 0.1$). Figure 7 displays the results. The top row displays L_2Boosting, while the bottom row is elasticBoost (fit under the optimized λ). The minimum mean-squared-error (MSE) is slightly smaller for elasticBoost (217.9) than L_2Boosting (231.7) (first panels in top and bottom rows). Curiously, the MSE is minimized using about same number of iterations for both methods (190 for L_2Boosting and 169 for elasticBoost). The middle panels display the coefficient paths. The vertical blue line indicates the MSE optimized number of iterations. In the case of L_2Boosting only 4 nonzero coefficients are identified within the optimal number of steps, whereas elasticBoost finds all 15 nonzero coefficients. This can be seen more clearly in the right panels

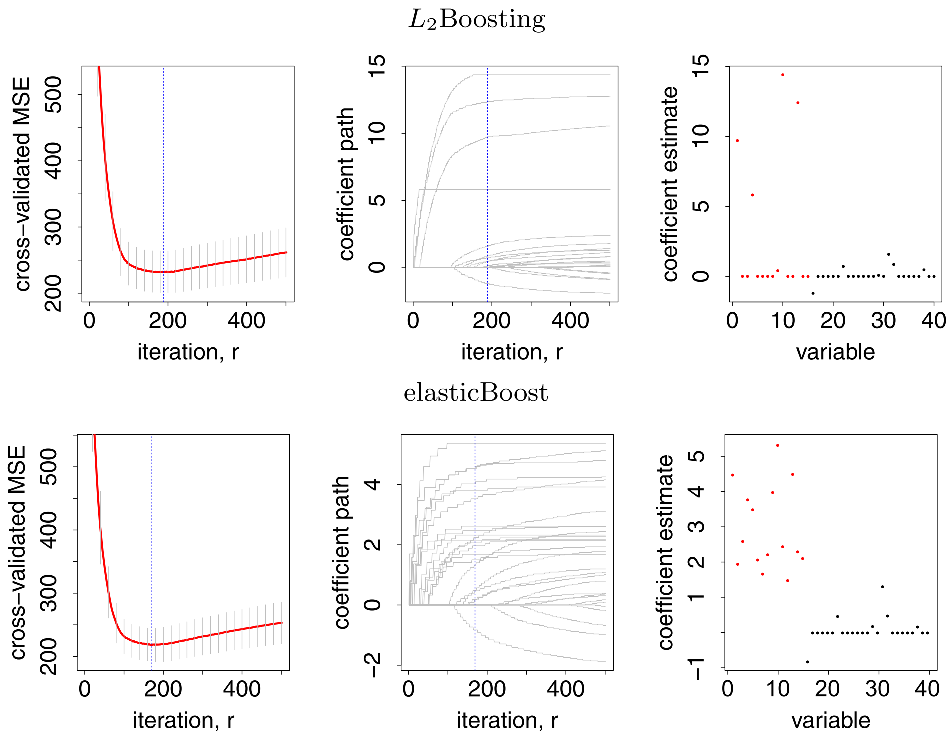

FIG. 7. *L_2Boosting (top row) versus elasticBoost (bottom row) from simulation* (4.2).

which show coefficient estimates at the optimized stopping time. Not only are all 15 nonzero coefficients identified by elasticBoost, but their estimated coefficient values are all roughly near the true value of 3. In contrast, L_2Boosting finds only 4 coefficients due to strong repressibility. Its coefficient estimates are also wildly inaccurate. While this does not overly degrade prediction error performance (as evidenced by the first panel), variable selection performance is seriously impacted.

The entire experiment was then repeated 250 times using 250 independent learning sets. Figure 8 displays the coefficient estimates from these 250 experiments for elasticBoost (left side) and L_2Boosting (right side) as boxplots. The top panel are based on the original sample size of $n = 100$ and the bottom panel use a larger sample size $n = 1000$. The results confirm our previous finding: elasticBoost is consistently able to group variables and outperform L_2Boosting in terms of variable selection.

Finally, the left panel of Figure 9 displays the difference in test set MSE for L_2Boosting and elasticBoost as a function of λ over the 250 experiments ($n = 100$). Negative values indicate a lower MSE for elasticBoost, which is generally the case for larger λ. The right panel displays the MSE optimized number of iterations for L_2Boosting compared to elasticBoost. Generally, elasticBoost re-

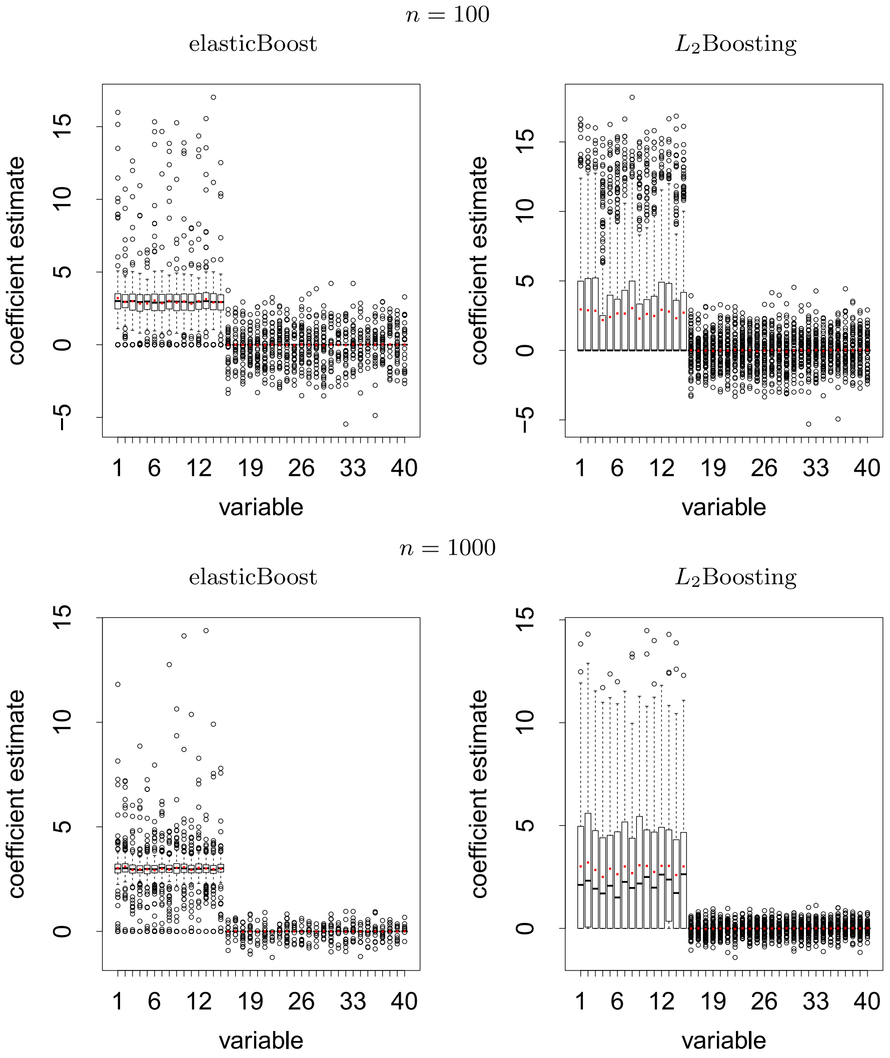

FIG. 8. *elasticBoost* (*left*) *versus L_2Boosting* (*right*) *from simulation* (4.2) *for* $n = 100$ (*top*) *and* $n = 1000$ (*bottom*) *based on* 250 *independent learning samples. The distribution of coefficient estimates are displayed as boxplots*; *mean values are given in red.*

quires fewer steps as λ increases. This is interesting, because as pointed out, this generally coincides with better MSE performance.

6. Discussion.

A key observation is that L_2Boosting's behavior along a fixed descent direction is fully specified with the exception of the descent length, L_r. In Theorem 2, we described a closed form solution for $m_{j,k}$, the number of steps until favorability, where $k = l_r$ is the currently selected coordinate direction and $j = l_{r+1}$ is the next most favorable direction. Theorem 2 quantifies L_2Boosting's

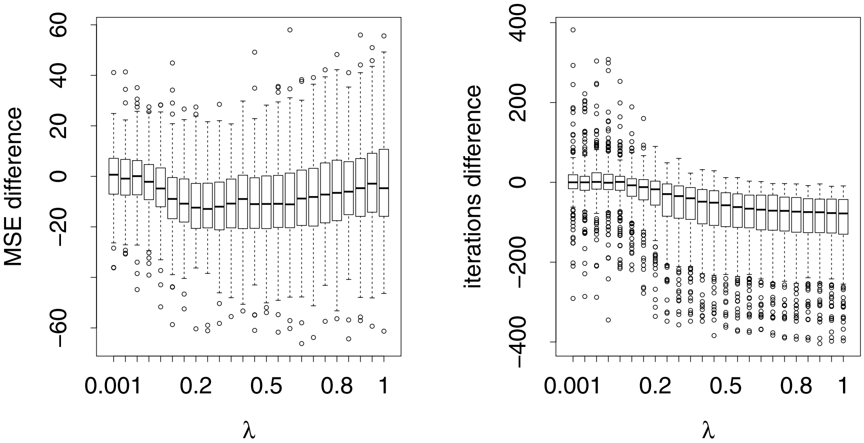

FIG. 9. *Left*: *difference in test set performance of L_2Boosting compared to elasticBoost. Right*: *difference in MSE optimized number of iterations for L_2Boosting compared to elasticBoost.*

descent length, thus allowing us to characterize its solution path as a series of fixed descents where the next coordinate direction, chosen from all candidates $j \neq k$, is determined as that with the minimal descent length $m_{j,k}$ (assuming no ties). Since we choose from among all directions $j \neq k$, $m_{j,k}$, and equivalently the step length $\nu_{j,k}$, can be characterized as measures to favorability, a property of each coordinate at any iteration r. These measures are a function of ν and the ratio of gradient-correlations $d_{j,k}$ and the correlation coefficient $R_{j,k}$ relative to the currently selected direction k.

Characterizing the L_2Boosting solution path by $m_{j,k}$ provides considerable insight when examining the limiting conditions. When $m_{j,k} \to 1$, L_2Boosting exhibits active set cycling, a property explored in detail in Section 3. We note that this condition is fundamentally a result of the optimization method which drives $|d_{j,k}| \to 1$ when ν is arbitrarily small. This virtually guarantees the notorious slow convergence seen with infinitesimal forward stagewise algorithms.

The repressibility condition occurs in the alternative limiting condition $m_{j,k} \to \infty$. Repressibility arises when the gradient correlation ratio $d_{j,k}$ equals the correlation $R_{j,k}$. When $|R_{j,k}| < 1$, j is said to be strongly repressed by k, and while descending along k, the absolute gradient-correlation for j can never be equal to or surpass the absolute gradient-correlation for k. Strong repressibility plays a crucial role in correlated settings, hindering variables from being actively selected. Adding L_2 regularization reverses repressibility and substantially improves variable selection for elasticBoost, an L_2Boosting implementation involving the data augmentation framework used by the elastic net.

SUPPLEMENTARY MATERIAL

Proofs of results from "Characterizing L_2Boosting" (DOI: 10.1214/12-AOS997SUPP; .pdf). An online supplementary file contains the detailed proofs for Theorems 1 through 9. These proofs make use of various notation described in the paper.

REFERENCES

BÜHLMANN, P. (2006). Boosting for high-dimensional linear models. *Ann. Statist.* **34** 559–583. MR2281878

BÜHLMANN, P. and YU, B. (2003). Boosting with the L_2 loss: Regression and classification. *J. Amer. Statist. Assoc.* **98** 324–339. MR1995709

EFRON, B., HASTIE, T., JOHNSTONE, I. and TIBSHIRANI, R. (2004). Least angle regression (with discussion, and a rejoinder by the authors). *Ann. Statist.* **32** 407–499. MR2060166

EHRLINGER, J. (2011). Regularization: Stagewise regression and bagging. Ph.D. thesis, Case Western Reserve Univ., Cleveland, OH. MR2873516

EHRLINGER, J. and ISHWARAN, H. (2012). Supplement to "Characterizing L_2Boosting." DOI:10.1214/12-AOS997SUPP.

FRIEDMAN, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.* **29** 1189–1232. MR1873328

HASTIE, T. (2007). Comment on "Boosting algorithms: Regularization, prediction and model fitting." *Statist. Sci.* **22** 513–515. MR2420456

HASTIE, T., TAYLOR, J., TIBSHIRANI, R. and WALTHER, G. (2007). Forward stagewise regression and the monotone lasso. *Electron. J. Stat.* **1** 1–29. MR2312144

MALLAT, S. and ZHANG, Z. (1993). Matching pursuits with time–frequency dictionaries. *IEEE Trans. Signal Proc.* **41** 3397–3415.

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* **58** 267–288. MR1379242

ZOU, H. and HASTIE, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **67** 301–320. MR2137327

DEPARTMENT OF QUANTITATIVE HEALTH SCIENCES
CLEVELAND CLINIC
CLEVELAND, OHIO 44195
USA
E-MAIL: john.ehrlinger@gmail.com

DIVISION OF BIOSTATISTICS
DEPARTMENT OF EPIDEMIOLOGY
AND PUBLIC HEALTH
UNIVERSITY OF MIAMI
MIAMI, FLORIDA 33136
USA
E-MAIL: hemant.ishwaran@gmail.com
URL: http://web.ccs.miami.edu/~hishwaran