SIAM J. Optim.
Vol. 30, No. 4, pp. 2780–2808

RANDOMIZED GRADIENT BOOSTING MACHINE[*]

HAIHAO LU[†] AND RAHUL MAZUMDER[‡]

Abstract. The Gradient Boosting Machine (GBM) introduced by Friedman [J. H. Friedman, *Ann. Statist.*, 29 (2001), pp. 1189–1232] is a powerful supervised learning algorithm that is very widely used in practice—it routinely features as a leading algorithm in machine learning competitions such as Kaggle and the KDDCup. In spite of the usefulness of GBM in practice, our current theoretical understanding of this method is rather limited. In this work, we propose the Randomized Gradient Boosting Machine (RGBM), which leads to substantial computational gains compared to GBM by using a randomization scheme to reduce search in the space of weak learners. We derive novel computational guarantees for RGBM. We also provide a principled guideline towards better step-size selection in RGBM that does not require a line search. Our proposed framework is inspired by a special variant of coordinate descent that combines the benefits of randomized coordinate descent and greedy coordinate descent, and may be of independent interest as an optimization algorithm. As a special case, our results for RGBM lead to superior computational guarantees for GBM. Our computational guarantees depend upon a curious geometric quantity that we call the Minimal Cosine Angle, which relates to the density of weak learners in the prediction space. On a series of numerical experiments on real datasets, we demonstrate the effectiveness of RGBM over GBM in terms of obtaining a model with good training and/or testing data fidelity with a fraction of the computational cost.

Key words. gradient boosting, ensemble methods, convex optimization, coordinate descent, computational guarantees, first order methods

AMS subject classifications. 90C25, 68U01

DOI. 10.1137/18M1223277

1. Introduction.

The Gradient Boosting Machine (GBM) [15] is a powerful supervised learning algorithm that combines multiple weak learners into an ensemble with excellent predictive performance. It works very well in several prediction tasks arising in spam filtering, online advertising, fraud detection, anomaly detection, computational physics (e.g., the Higgs Boson discovery), etc., and has routinely featured as a top algorithm in Kaggle competitions and the KDDCup [7]. GBM can naturally handle heterogeneous datasets (highly correlated data, missing data, categorical data, etc.) and leads to interpretable models by building an additive model [14]. It is also quite easy to use with several publicly available implementations: Scikit-learn [30], Spark MLlib [25], LightGBM [20], XGBoost [7], TensorFlow Boosted Trees [31], etc.

In spite of the usefulness of GBM in practice, there is a considerable gap between its theoretical understanding and its success in practice. The traditional interpretation of GBM is to view it as a form of steepest descent in a certain functional space [15]. While this viewpoint serves as a good starting point, the framework lacks rigorous computational guarantees, especially when compared to the growing body of literature in first order convex optimization. There has been some work on deriving convergence rates of GBM—see, for example, [3, 11, 26, 39], and our discussion in

[*] Received by the editors October 29, 2018; accepted for publication (in revised form) June 9, 2020; published electronically October 7, 2020.
https://doi.org/10.1137/18M1223277

Funding: The second author's work was partially supported by grants ONR-N000141512342, ONR-N000141812298 (YIP), and NSF-IIS-1718258.

[†] Booth School of Business, University of Chicago, Chicago, IL 60637 USA (haihao@mit.edu).

[‡] MIT Sloan School of Management, Operations Research Center and MIT Center for Statistics, Massachusetts Institute of Technology, Cambridge, MA 02142 USA (rahulmaz@mit.edu).

section 1.3. Moreover, there are many heuristics employed by practical implementations of GBM that work well in practice—for example, the constant step-size rule and column subsampling mechanism implemented in XGBoost [7]. However, a formal explanation of these heuristics seems to be lacking in the current literature. This prevents us from systematically addressing important (tuning) parameter choices that may be informed by answers to questions like, how might one choose an optimal step-size, and how many weak learners should one subsample? Building a framework to help address these concerns is one goal of this paper. In this work we build a methodological framework for understanding GBM and its randomized variant introduced here, Randomized Gradient Boosting Machine (RGBM), by using tools from convex optimization. Our hope is to narrow the gap between the theory and practice of GBM and its randomized variants. Below, we revisit the classical GBM framework and then introduce RGBM.

1.1. Gradient Boosting Machine.

We consider a supervised learning problem [18], with n training examples (x_i, y_i), $i = 1, \dots, n$, such that $x_i \in \mathbb{R}^p$ is the feature vector of the ith example and $y_i \in \mathbb{R}$ is a label (in a classification problem) or a continuous response (in a regression problem). In the classical version of GBM [15], the prediction corresponding to a feature vector x is given by an additive model of the form

$$f(x) := \sum_{m=1}^{M} \beta_{j_m} b(x; \tau_{j_m}), \tag{1}$$

where each basis function $b(x;\tau) \in \mathbb{R}$ (also called a weak learner) is a simple function of the feature vector indexed by a parameter τ, and β_j is the coefficient of the jth weak learner. Here, β_{j_m} and τ_{j_m} are chosen in an adaptive fashion to improve the data fidelity (according to a certain rule), as discussed below. Examples of weak learners commonly used in practice [18] include wavelet functions, support vector machines, tree stumps (i.e., decision trees of depth one), and classification and regression trees (CART) [5]. We assume here that the set of weak learners is finite with cardinality K—in many of the examples alluded to above, K can be exponentially large, thereby posing computational challenges.

Let $\ell(y, f(x))$ be a measure of data fidelity at the observation (y, x) for the loss function ℓ, which is assumed to be differentiable in the second coordinate. A primary goal of machine learning is to obtain a function f that minimizes the expected loss $\mathbb{E}_P(\ell(y, f(x)))$, where the expectation is taken over the unknown distribution of (y, x) (denoted by P). *One* way to achieve this goal is to consider the empirical loss and *approximately* minimize it using an algorithm like GBM.[1] GBM is an algorithm that finds a good estimate of f by approximately minimizing the empirical loss:

$$\min_{f} \quad \sum_{i=1}^{n} \ell(y_i, f(x_i)), \tag{2}$$

where $\ell(y_i, f(x_i))$ measures data fidelity for the ith sample (y_i, x_i). The original version of GBM [15] (presented in Algorithm 1) can be viewed as applying a steepest

[1] Approximately minimizing the empirical loss function via GBM is empirically found to lead to models with good generalization properties—see, e.g., [43] for some formal explanation (under simple settings). The focus of this paper is on the algorithmic properties of GBM as opposed to its generalization properties.

Algorithm 1. Gradient Boosting Machine (GBM) [15].

```
Initialization. Initialize with $f^0(x) = 0$.
For $m = 0, \dots, M-1$ do:
    (1) Compute pseudoresidual $r^m = -\left[\frac{\partial \ell(y_i, f^m(x_i))}{\partial f^m(x_i)}\right]_{i=1,\dots,n}$.
    (2) Find the best weak learner: $j_m = \arg\min_{j \in [K]} \min_\sigma \sum_{i=1}^n (r_i^m - \sigma b(x_i; \tau_j))^2$.
    (3) Choose the step-size $\rho_m$ by line search: $\rho_m = \arg\min_\rho \sum_{i=1}^n \ell(y_i, f^m(x_i) + \rho b(x_i; \tau_{j_m}))$.
    (4) Update the model $f^{m+1}(x) = f^m(x) + \rho_m b(x; \tau_{j_m})$.
Output. $f^M(x)$.
```

descent algorithm to minimize the loss function (2). GBM starts from a null model $f \equiv 0$ and at iteration m computes the pseudoresidual r^m, i.e, the negative gradient of the loss function with respect to the prediction. Note that the ith coordinate of r^m is given by $r_i^m = -\partial \ell(y_i, f^m(x_i))/\partial f^m(x_i)$ for $i = 1, \dots, n$. GBM finds the best weak learner that fits r^m in the least squares sense:

$$j_m = \mathop{\arg\min}_{j \in [K]} \min_\sigma \sum_{i=1}^n (r_i^m - \sigma b(x_i; \tau_j))^2, \tag{3}$$

where $[K]$ is a shorthand for the set $\{1, \dots, K\}$. (In case of ties in the "argmin" operation in (3), we choose the one with the smallest index—this convention is used throughout the paper.) We then add the j_mth weak learner into the model by using a line search. As the iterations progress, GBM leads to a sequence of models $\{f^m\}_{m \in [M]}$ (see Algorithm 1), indexed by m (the number of GBM iterations). Each model f^m corresponds to a certain data fidelity and a (small) number of basis elements with corresponding coefficient weights [11, 15]. Together, they control the out-of-sample (or generalization) performance of the model. The usual intention of GBM is to stop early, i.e., approximately minimize problem (2)—with the hope that the corresponding model will lead to good predictive performance [11, 15, 43].

Note that since we perform a line search, rescaling the prediction vector $[b(x_i; \tau_j)]_{i \in [n]}$ does not change the output of Algorithm 1. Hence, without loss of generality, we assume that the prediction vector is normalized throughout the paper.

Assumption 1.1. The prediction vector corresponding to each weak learner is normalized—that is, for every τ, we have $\sum_{i=1}^n b(x_i; \tau)^2 = 1$.

Remark 1.1. Note that Assumption 1.1 is mainly used to simplify the notation and proofs in the paper (that follow). This assumption does not change the convergence guarantees in Theorems 4.1 and 4.2.

1.2. Randomized Gradient Boosting Machine.

The most expensive step in GBM involves finding the best weak learner (step (2) in Algorithm 1). For example, when the weak learners are decision trees of depth d, finding the best weak learner requires one to go over $O(n^{2^d-1} p^{2^d-1})$ possible tree splits—this is computationally intractable for medium-scale problems, even when $d = 1$.

It seems natural (and practical) to use a randomization scheme to reduce the cost associated with step (2) in Algorithm 1. To this end, we propose RGBM (see Algorithm 2), where the basic idea is to use a randomized approximation for step (3). To be more specific, in each iteration of RGBM, we randomly pick a small subset of weak learners J by some rule (see section 1.2.1) and then choose the best candidate from within J:

$$j_m = \underset{j \in J}{\arg\min} \; \min_{\sigma} \; \sum_{i=1}^{n} (r_i^m - \sigma b(x_i; \tau_j))^2 \; . \tag{4}$$

If we set $|J|$ (the size of J) to be much smaller than the total number of weak learners K, the cost per iteration in RGBM will be much lower than GBM. We note that the implementation of XGBoost utilizes a related heuristic (called column subsampling) [7], which has been shown to work well in practice. However, to our knowledge, we are not aware of any prior work that formally introduces and studies the RGBM algorithm—this is the main focus of our paper.

Note that the randomized selection rule we are advocating in RGBM is *different* from that employed in the well-known Stochastic Gradient Boosting framework by Friedman [16], in which Friedman introduced a procedure that randomly selects a subset of the *training* examples to fit a weak learner at each iteration. In contrast, we randomly choose a subset of weak learners in RGBM. Indeed, both feature and sample subsampling are applied in the context of random forests [21]; however, we remind the reader that random forests are quite different from GBM.

Algorithm 2. Randomized Gradient Boosting Machine (RGBM).

```
Initialization. Initialize with $f^0(x) = 0$.
For $m = 0, \ldots, M-1$ do:
    (1) Compute pseudoresidual $r^m = -\left[\frac{\partial \ell(y_i, f^m(x_i))}{\partial f^m(x_i)}\right]_{i=1,\ldots,n}$.
    (2) Pick a random subset $J$ of weak learners by some rule (i.e., one of Type 0–Type 3).
    (3) Find the best weak learner in $J$: $j_m = \arg\min_{j \in J} \min_\sigma \sum_{i=1}^n (r_i^m - \sigma b(x_i; \tau_j))^2$.
    (4) Choose the step-size $\rho_m$ by one of the following rules:
        • line search: $\rho_m = \arg\min_\rho \sum_{i=1}^n \ell(y_i, f^m(x_i) + \rho b(x_i; \tau_{j_m}))$;
        • constant step-size: $\rho_m = \rho\,(\sum_{i=1}^n r_i^m b(x_i; \tau_{j_m}))$, where $\rho$ is a constant specified a priori.
    (5) Update the model $f^{m+1}(x) = f^m(x) + \rho_m b(x; \tau_{j_m})$.
Output. $f^M(x)$.
```

1.2.1. Random selection rules for choosing subset *J*.

We present a set of selection rules to choose the random subset J in step (2) of Algorithm 2:

[Type 0]: *(Full deterministic selection.)* We choose J as the whole set of weak learners. This is a deterministic selection rule.

[Type 1]: *(Random selection.)* We choose uniformly at random t weak learners from all possible weak learners without replacement—the collection is denoted by J.

[Type 2]: *(Random single group selection.)* Given a nonoverlapping partition of the weak learners, we pick one group uniformly at random and denote the collection of weak learners in that group by J.

[Type 3]: *(Random multiple group selection.)* Given a nonoverlapping partition of the weak learners, we pick t groups uniformly at random and let the collection of weak learners across these groups be J.

Remark 1.2. RGBM with Type 0 selection rule leads to GBM.

We present an example to illustrate the different selection rules introduced above.

Example. We consider GBM with decision stumps for a binary classification problem. Recall that a decision stump is a decision tree [18] with unit depth. The parameter τ of a decision stump contains two pieces of information: (i) which feature to split, and (ii) what value to split on. More specifically, a weak learner characterized by $\tau = (g, s)$ for $g \in [p]$ and $s \in \mathbb{R}$ is given by (up to a sign change)

$$b(x; \tau = (g, s)) = \begin{cases} 1 & \text{if } x_g \le s, \\ -1 & \text{if } x_g > s. \end{cases} \tag{5}$$

Notice that for a given feature x_g and n training samples, it suffices to consider at most n different values for s (and equality holds when the feature values are all distinct). This leads to $K = np$ many tree stumps $\{b(x; \tau)\}_\tau$ indexed by τ. For the Type 0 selection rule, we set J to be the collection of all np tree stumps in a deterministic fashion. As an example of the Type 1 selection rule, J can be a collection of t tree stumps selected randomly without replacement from all of np tree stumps. Let I_g be a group comprising all tree stumps that split on feature x_g—i.e., $I_g = \{(g, s) \mid s\}$ for a feature index $g \in [p]$. Then $\{I_g\}_{g \in [p]}$ defines a partition of all possible tree stumps. Given such a partition, an example of the Type 2 selection rule is as follows: We randomly choose $g \in [p]$ and set $J = I_g$. Instead, one can also pick t (out of p) features randomly and choose all nt tree stumps on those t features as the set J—this leads to an instance of the Type 3 selection rule. Note that a special case of Type 3 with $t = 1$ is the Type 2 selection rule.

For motivation, we illustrate the key operating characteristics of RGBM with a real-data example. Figure 1 shows the computational gains of RGBM for solving a binary classification problem with decision stumps. Here we use the Type 3 selection rule (as described above), where each group represents all tree stumps splitting on a single feature, and $G = 123$ is the total number of groups. Different lines correspond to different t values—namely, how many groups appear in the random set J in each iteration. The blue line corresponds to GBM (Algorithm 1) as it uses all the groups. The major computational cost stems from computing the best weak learner from a subset of weak learners. The implementation details (leading to Figure 1) can be found in section 5. The left column of Figure 1 presents the training and testing loss versus number of iterations. We can see that when the number of groups t gets smaller, we may get less improvement (in training loss) per iteration, but not by a large margin (for example, the case $t = 24$ has similar behavior to the case $t = 123$). The right column of Figure 1 shows the training/testing loss versus running time (in seconds). We can see that with a smaller t, the cost per iteration decreases dramatically—overall, a small value of t (though not the smallest) requires less computation (compared to a larger value of t) to achieve a similar training/testing error.

1.3. Related literature.

Convergence guarantees for GBM. The development of general convergence guarantees for GBM has seen several milestones in the past decade. After being proposed by Friedman [15], Collins, Schapire, and Singer [9] showed the convergence of GBM, without any rates. Bickel, Ritov, and Zakai [3] proved an exponential convergence rate (more precisely $O(\exp(1/\varepsilon^2)))$ when the loss function is both smooth and strongly convex. Telgarsky [39] studied the primal-dual structure of GBM. By taking advantage of the dual structure, Telgarsky presented a linear convergence result for GBM with the line-search step-size rule. However, the constants in the linear rate

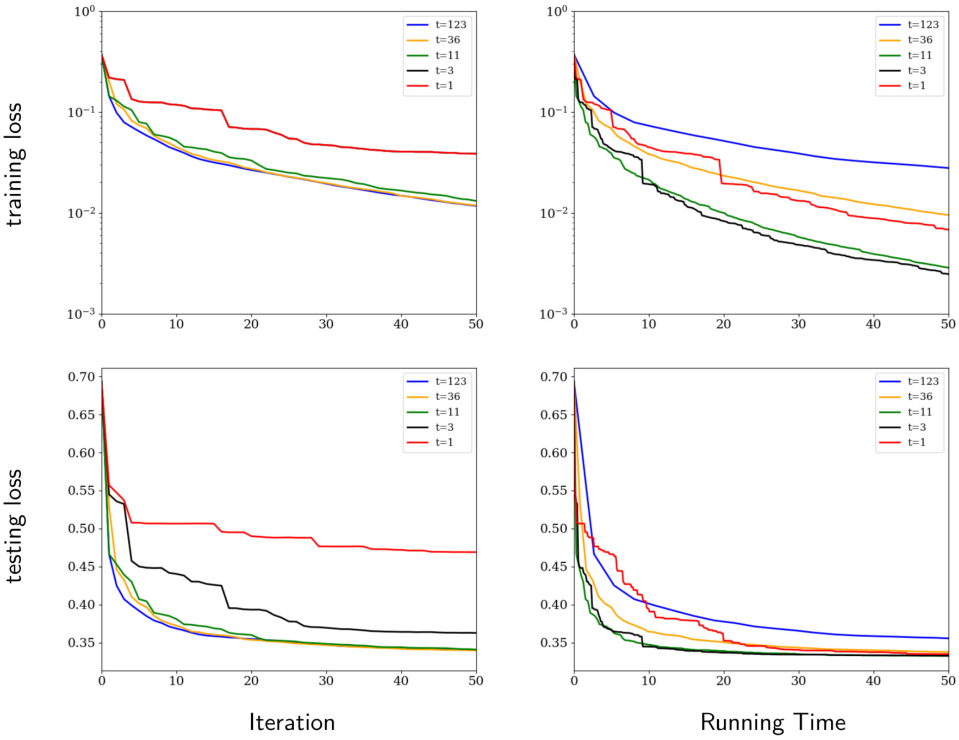


FIG. 1. *Plots showing the training optimality gap in* log *scale (top panel) and testing loss (bottom panel) versus number of RGBM iterations and the associated running time (secs) for RGBM with different t values. We consider the* a9a *dataset (for a classification task) from the LIBSVM library (see text for details). A smaller value of t corresponds to a smaller cost per iteration. As expected, we see overall computational savings for a value of t that is smaller than the maximum $t = 123$, which corresponds to GBM.*

are not as transparent as the ones we obtain in this paper, with the only exception being the exponential loss function.[2] Several works have studied the convergence rate as applied to specific loss functions. Freund and Schapire [13] showed a linear convergence rate for AdaBoost (this can be thought of as GBM with exponential loss and line-search rule) under a weak learning assumption. Mukherjee, Rudin, and Schapire [26] showed an $O(1/\varepsilon)$ rate for AdaBoost, but the constant depends on the dataset and can be exponentially large in the dimension of the problem. We refer the reader to [39] for a nice review on the early work on Boosting. For LS-Boost (gradient boosting with a least squares loss function), Freund, Grigas, and Mazumder [11] recently showed a linear rate of convergence, but the rate is not informative when the number of weak learners is large. Our analysis here provides a much sharper description of the constant—we achieve this by using a different analysis technique.

Convergence rates of iterative algorithms for classification are closely related to the notion of margins [12]. Ramdas and Pena [32, 33] established interesting geometric connections between margins, iterative algorithms for classification problems (e.g., the Perceptron and von Neumann algorithms), and condition numbers arising in the study of convex feasibility problems [8, 10].

[2] The rate for other loss functions involves a quantity that can be exponentially large in the number of features p.

Coordinate descent. Coordinate descent (CD) methods have a long history in optimization, and convergence of these methods has been extensively studied in the optimization community in the 1980s and 1990s—see [2, 23, 24] and [40] for a nice overview. There are roughly three types of CD methods, depending on how the coordinate is chosen: randomized, greedy, and cyclic CD. Randomized CD has received considerable attention since the seminal paper of Nesterov [28]. Randomized CD chooses a coordinate randomly from a certain fixed distribution. [34] provides an excellent review of theoretical results for randomized CD. Cyclic CD chooses the coordinates in a cyclic order (see [1] for its first convergence analysis). Recent work shows that cyclic CD may be inferior to randomized CD in the worst case [38]—in some examples arising in practice, however, cyclic CD can be better than randomized CD [1, 17, 19]. In greedy CD, we select the coordinate yielding the largest reduction in the objective function value. Greedy CD usually delivers better function values at each iteration (in practice), though this comes at the expense of having to compute the full gradient in order to select the coordinate with the largest magnitude of the gradient. On a related note, for the same training data fidelity (i.e., objective function value), greedy CD usually leads to a model with fewer nonzeros compared to randomized CD—in other words, greedy CD leads to models that are *more* sparse than randomized CD.[3]

As we will show later, GBM is precisely related to greedy CD. Thus, we focus here on some of the recent developments in greedy CD. [29] showed that greedy CD has faster convergence than random CD in theory, and also provided several applications in machine learning where the full gradient can be computed cheaply. Several parallel variants of greedy CD methods have been proposed in [35, 36, 41], and numerical results demonstrate their advantages in practice. [37] presents a useful scalability idea for steepest CD by maintaining an approximation of the entire gradient vector, which is used to identify the coordinate to be updated. More recently, [22] proposes an accelerated greedy coordinate descent method.

1.4. Contributions.

Our contributions in this paper can be summarized as follows:

1. We propose RGBM, a new randomized version of GBM which leads to significant computational gains compared to GBM. This is based on what we call a Random-then-Greedy procedure (i.e., we select a random subset of weak learners and then find the best candidate among them by using a greedy strategy). In particular, this provides a formal justification of heuristics used in popular GBM implementations like XGBoost, and also suggests improvements. Our framework may provide guidelines for a principled choice of step-size rules in RGBM.
2. We derive new computational guarantees for RGBM based on a CD interpretation. In particular, this leads to new guarantees for GBM that are superior to existing guarantees for certain loss functions. The constants in our computational guarantees are in terms of a curious geometric quantity that we call the Minimal Cosine Angle—this relates to the density of the weak learners in the prediction space.
3. From an optimization viewpoint, our Random-then-Greedy CD procedure leads to a novel generalization of CD-like algorithms and promises to be of independent interest as an optimization algorithm. Our proposal combines the efficiency of randomized coordinate descent and the sparsity of the solution obtained by greedy CD.

[3] We assume here that CD is initialized with a zero solution.

Notation. For an integer s, let $[s]$ denote the set $\{1, 2, \ldots, s\}$. For $a, b \in \mathbb{R}^p$, $\cos(a, b)$ denotes the cosine of the angle between a and b, that is, $\cos(a, b) = \langle a, b\rangle/(\|a\|_2\|b\|_2)$. Matrix B denotes the prediction for all samples over every possible weak learner, namely, $B_{i,j} = b(x_i; \tau_j)$ for $i \in [n], j \in [K]$. $B_{:j}$ is the jth column of B and $B_{i:}$ is the ith row of B. We say $\{I_g\}_{g\in[G]}$ is a partition of $[K]$ if $\cup_{g\in[G]} I_g = [K]$ and I_gs are disjoint. We often use the notation $[a_i]_i$ to represent a vector a.

2. Random-then-Greedy Coordinate Descent in the coefficient space.

Let $[b(x; \tau_j)]_{j\in[K]}$ be a family of all possible weak learners. Let

$$f(x) = \sum_{j=1}^{K} \beta_j b(x; \tau_j)$$

be a weighted sum of all K weak learners $b(x; \tau_j)$, where β_j is the coefficient of the jth weak learner (we expect a vast majority of the β_j's to be zero). We refer to the space of $\beta \in \mathbb{R}^K$ as the "coefficient space." We can rewrite the minimization problem (2) in the coefficient space as

$$\min_{\beta} \quad L(\beta) := \sum_{i=1}^{n} \ell\left(y_i, \sum_{j=1}^{K} \beta_j b(x_i; \tau_j)\right) . \tag{6}$$

Here, we assume K to be finite (but potentially a very large number). We expect that our results can be extended to deal with an infinite number of weak learners, but we do not pursue this direction in this paper for simplicity of exposition.

Recall that B is an $n \times K$ matrix of the predictions for all feature vectors over every possible weak learner, namely, $B = [b(x_i; \tau_j)]_{i\in[n],j\in[K]}$. Then each column of B represents the prediction of one weak learner for the n samples, and each row of B represents the prediction of all weak learners for a single sample. Thus we can rewrite (6) as

$$\min_{\beta} L(\beta) := \sum_{i=1}^{n} \ell\left(y_i, B_{i:}\beta\right) . \tag{7}$$

Algorithm 3 presents the Random-then-Greedy Coordinate Descent (RtGCD) algorithm for solving (7). We initialize the algorithm with $\beta = 0$. At the start of the mth iteration, the algorithm randomly chooses a subset J of the coordinates using one of the four types of selection rules described in section 1.2.1. The algorithm then "greedily" chooses $j_m \in J$ by finding a coordinate in $\nabla_J L(\beta^m)$ with the largest magnitude. We then perform a coordinate descent step on the j_mth coordinate with either a line-search step-size rule or a constant step-size rule.

Remark 2.1. RtGCD forms a bridge between random CD and greedy CD. RtGCD leads to greedy CD when J is the set of all coordinates and random CD when J is a coordinate chosen uniformly at random from all coordinates. To our knowledge, RtGCD is a new coordinate descent algorithm and promises to be of independent interest as an optimization algorithm.

The choice of the group structure (or J) depends upon the application. For example, in the context of Boosting (using trees as weak learners), the groups are informed by the Boosting procedure—this is usually specified by the practitioner. In the context of parallel CD algorithms, [36] proposed a method to group coordinates into blocks for

Algorithm 3. Random-then-Greedy Coordinate Descent (RtGCD) in the coefficient space.

```
**Initialization.** Initialize with $\beta^0 = 0$.
For $m = 0, \dots, M-1$ do:
  **Perform updates:**
  (1) Pick a random subset $J$ of coordinates by *some rule* (i.e., one of Type 0–Type 3).
  (2) Use a greedy rule to find a coordinate in $J$: $j_m = \arg\max_{j \in J} |\nabla_j L(\beta^m)|$.
  (3) Choose the step-size $\rho_m$ by
      • line search: $\rho_m = \arg\min_\rho \sum_{i=1}^n \ell(y_i, B_{i:}\beta^m + \rho B_{i,j_m})$;
      • constant step-size: $\rho_m = -\rho \nabla_{j_m} L(\beta^m)$ for a given constant $\rho$.
  (4) Update coefficients: $\beta^{m+1} = \beta^m + \rho_m e^{j_m}$.
**Output.** The coefficient vector $\beta^M$.
```

algorithmic efficiency—their method updates multiple coordinates within each block. While the context of our work and that of [36] are different, it will be interesting to see how ideas in [36] can be used with Algorithm 3 for improved performance.

The following proposition shows that RGBM (Algorithm 2) is equivalent to RtGCD in the coefficient space (Algorithm 3).

PROPOSITION 2.1. *Suppose Algorithm* 2 *makes the same choice of the random set* J *as Algorithm* 3 *(in each iteration), and the step-size rules are chosen to be the same in both algorithms. Then the outputs of Algorithms* 2 *and* 3 *are the same.*

Proof. We will show by induction that $f^m(x)$ in Algorithm 2 is the same as $\sum_{j=1}^K \beta_j^m b(x;\tau_j)$ in Algorithm 3 for $m = 0, 1, \dots, M$. Then Proposition 2.1 holds as a special case for $m = M$.

For $m = 0$, we have $f^0(x) = 0 = \sum_{j=1}^K \beta_j^0 b(x;\tau_j)$. Suppose that $f^m(x) = \sum_{j=1}^K \beta_j^m b(x;\tau_j)$. Then

$$\nabla_j L(\beta^m) = -\langle B_{:j}, r^m \rangle \ , \tag{8}$$

where r^m is the pseudoresidual. In iteration m, the same random subset J is chosen by both algorithms. Next, Algorithm 2 greedily chooses the weak learner by

$$j_m = \arg\min_{j \in J} \min_\sigma \sum_{i=1}^n (r_i^m - \sigma b(x_i;\tau_j))^2 = \arg\min_{j \in J} \min_\sigma \|r^m - \sigma B_{:j}\|_2^2 \ .$$

Notice that for any j, it holds that $\arg\min_\sigma \|r^m - \sigma B_{:j}\|_2^2 = \langle B_{:j}, r^m \rangle$. Hence, we have that

$$\begin{aligned} j_m &= \arg\min_{j \in J} \ \|r^m - \langle B_{:j}, r^m \rangle B_{:j}\|_2^2 = \arg\min_{j \in J} \left(-\frac{1}{2} \langle B_{:j}, r^m \rangle^2 \right) \\ &= \arg\max_{j \in J} \ |\langle B_{:j}, r^m \rangle| = \arg\max_{j \in J} \ |\nabla_j L(\beta^m)| \ , \end{aligned}$$

where the second equality follows from $\|B_{:j}\|_2^2 = \sum_{i=1}^n b(x_i,\tau_j)^2 = 1$ due to Assumption 1.1 and the last equality utilizes (8). Therefore, coordinate j_m obtained by Algorithm 2 in the mth iteration is the same as that obtained by Algorithm 3.

Suppose that both algorithms use a step-size based on the line-search rule. Then the step-size in Algorithm 2 is given by

$$\rho_m = \arg\min_{\rho} \sum_{i=1}^{n} \ell(y_i, f^m(x_i) + \rho b(x_i; \tau_{j_m})) = \arg\min_{\rho} \sum_{i=1}^{n} \ell(y_i, B_{i:}\beta^m + \rho B_{i,j_m}) \ ,$$

where we have (by the induction hypothesis) that $f^m(x_i) = B_{i:}\beta^m$. Thus the step-size ρ_m is the same as that chosen by Algorithm 3 (with line-search rule).

Now, suppose both algorithms use a constant step-size rule with the same constant ρ. Then the step-size in Algorithm 2 is given by

$$\rho_m = \rho \left(\sum_{i=1}^{n} r_i^m b(x_i; \tau_{j_m}) \right) = \rho \langle r^m, B_{:,j_m} \rangle = -\rho \nabla_{j_m} L(\beta^m) \ ,$$

which is the same step-size as that in Algorithm 3 (with constant step-size rule).

Thus, the step-size ρ_m at the mth iteration in Algorithm 2 is the same as that of Algorithm 3 for both step-size rules. Therefore, it holds that

$$f^{m+1}(x) = f^m(x) + \rho_m b(x; \tau_{j_m}) = \sum_{j=1}^{K} \beta_j^m b(x; \tau_j) + \rho_m b(x; \tau_{j_m}) = \sum_{j=1}^{K} \beta_j^{m+1} b(x; \tau_j) \ ,$$

which completes the proof by induction. □

Remark 2.2. In the special case when J contains all weak learners (i.e., with Type 0 random selection rule), Algorithm 3 reduces to standard greedy coordinate descent and Proposition 2.1 shows that GBM (Algorithm 1) is equivalent to greedy coordinate descent in the coefficient space.

3. Machinery: Structured norms and random selection rules. In this section, we introduce four norms and establish how they relate to the four types of selection rules (for J), as described in section 1.2.1.

3.1. Infinity norm, ordered ℓ_1 norm, $\ell_{1,\infty}$ group norm, and an ordered mixed norm. We introduce the following definitions.

DEFINITION 3.1. *The "infinity norm" $\|\cdot\|_\infty$ of vector $a \in \mathbb{R}^K$ is defined as*

$$\|a\|_\infty = \max_{j \in [K]} |a_j| \ . \qquad \text{(Infinity norm)}$$

DEFINITION 3.2. *The "ordered ℓ_1 norm" $\|\cdot\|_{\mathcal{S}}$ with parameter $\gamma \in \mathbb{R}^K$ of vector $a \in \mathbb{R}^K$ is defined as*

$$\|a\|_{\mathcal{S}} = \sum_{j=1}^{K} \gamma_i |a_{(j)}| \ , \qquad \text{(Ordered } \ell_1 \text{ norm)}$$

where the parameter γ satisfies $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_K \geq 0$ with $\sum_{j=1}^{K} \gamma_j = 1$, and $|a_{(1)}| \geq |a_{(2)}| \geq \cdots \geq |a_{(K)}|$ are the decreasing absolute values of the coordinates of a.

DEFINITION 3.3. *If $\{I_g\}_{g \in [G]}$ is a partition of $[K]$, then the "$\ell_{1,\infty}$ group norm" of vector $a \in \mathbb{R}^K$ is defined as*

$$\|a\|_{\mathcal{G}} = \frac{1}{G} \sum_{g=1}^{G} \|a_{I_g}\|_\infty, \qquad \text{(Group norm)}$$

where $\|a_{I_g}\|_\infty$ is the infinity norm of a_{I_g} (i.e., the subvector of a restricted to I_g) for $g \in [G]$.

Definition 3.4. *If $\{I_g\}_{g\in[G]}$ is a partition of $[K]$, then the "ordered mixed norm"*[4] *with parameter $\gamma \in \mathbb{R}^G$ of vector $a \in \mathbb{R}^K$ is defined as*

$$\|a\|_{\mathcal{C}} = \sum_{g=1}^{G} \gamma_g \|a_{I_{(g)}}\|_\infty \ , \qquad \textit{(Ordered mixed norm)}$$

where the parameter γ satisfies $\gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_G \geq 0$ and $\sum_{g=1}^{G}\gamma_g = 1$. Note that $\|a_{I_{(1)}}\|_\infty \geq \|a_{I_{(2)}}\|_\infty \geq \cdots \geq \|a_{I_{(G)}}\|_\infty$ are the sorted values of $\|a_{I_g}\|_\infty, g \in [G]$.

Remark 3.1. Note that the group norm [27] and ordered ℓ_1 norm (arising in the context of the Slope estimator) [4] appear as common regularizers in high-dimensional linear models. In this paper, however, they arise in a very different context—see section 3.2.

It can be easily seen that the ordered ℓ_1 norm is a special instance of the ordered mixed norm where each group contains one element, and the $\ell_{1,\infty}$ group norm is another special instance of the ordered mixed norm where the parameter $\gamma_g \equiv 1/G$ for $g \in [G]$.

With some elementary calculations, we can derive the dual norms of each of the above norms.

Proposition 3.1. (1) *The dual norm of the ordered ℓ_1 norm is*

$$\|b\|_{\mathcal{S}^*} = \max_{1\leq i\leq K} \frac{\sum_{j=1}^{i} |b_{(j)}|}{\sum_{j=1}^{i} \gamma_j} \ . \tag{9}$$

(2) *The dual norm of the $\ell_{1,\infty}$ group norm is*

$$\|b\|_{\mathcal{G}^*} = G \max_{1\leq g\leq G} \|b_{I_g}\|_1 \ .$$

(3) *The dual norm of the ordered mixed norm is*

$$\|b\|_{\mathcal{C}^*} = \max_{1\leq g\leq G} \frac{\sum_{j=1}^{g} \|b_{I_{(j)}}\|_1}{\sum_{j=1}^{g} \gamma_j} \ ,$$

where $\|b_{I_{(1)}}\|_1 \geq \|b_{I_{(2)}}\|_1 \geq \cdots \geq \|b_{I_{(G)}}\|_1$ are the values of $\|b_{I_g}\|_1, g \in [G]$, sorted in decreasing order.

Remark 3.2. The proof for part (1) of Proposition 3.1 can be found in Theorem 1 in [42]. The proof of part (2) is straightforward, and the proof of part (3) follows from those of (1) and (2).

3.2. Random-then-Greedy procedure. Here we introduce a Random-then-Greedy (RtG) procedure that uses a randomized scheme to deliver an approximate maximum of the absolute entries of a vector $a \in \mathbb{R}^K$. The expected value of the (random) output available from the RtG procedure with four types of selection rules (cf. section 1.2.1) can be shown to be related to the four norms introduced in section 3.1.

[4] The name stems from the fact that it is a combination of the ordered ℓ_1 norm and the $\ell_{1,\infty}$ group norm.

Formally, the RtG procedure is summarized below.

Random-then-Greedy (RtG) procedure
Given $a \in \mathbb{R}^K$,
1. Randomly pick a subset of coordinates $J \subseteq [K]$.
2. Output $\hat{j} = \arg\max_{j \in J} |a_j|$ and $|a_{\hat{j}}|$.

We will next obtain the probability distribution of $\hat{j}$, and the expectation of $|a_{\hat{j}}|$.

Let J be chosen by the Type 1 selection rule, namely, J is given by a collection of t coordinates, chosen uniformly at random from $[K]$ without replacement. A simple observation is that the probability of a coordinate j being chosen depends upon the magnitude of a_j *relative* to the other values $|a_i|$, $i \neq j$, and not the precise values of the entries in a. Note also that if the value of $|a_j|$ is higher than others, then the probability of selecting j increases: this is because (i) all coordinate indices in $[K]$ are equally likely to appear in J, and (ii) coordinates with a larger value of $|a_j|$ are chosen with higher probability. The following proposition formalizes the above observations and presents the probability of a coordinate being chosen.

Proposition 3.2. *Consider the RtG procedure for approximately finding the maximal coordinate of $a \in \mathbb{R}^K$ (in absolute value). Recall that (j) is the index of the jth largest coordinate of a in absolute value,*[5] *namely, $|a_{(1)}| \geq |a_{(2)}| \geq \cdots \geq |a_{(K)}|$. If the subset J is chosen by the Type 1 selection rule, the probability that (j) is returned is*

$$P\left(\hat{j} = (j)\right) := \gamma_t^K(j) = \frac{\binom{K-j}{t-1}}{\binom{K}{t}} \ . \tag{10}$$

Proof. There are $\binom{K}{t}$ different choices for the subset J, and each subset is chosen with equal probability. The event $\hat{j} = (j)$ happens if and only if $(j) \in J$ and the remaining $t-1$ coordinates are chosen from the $K-j$ coordinates. There are $\binom{K-j}{t-1}$ different choices of choosing such a subset J, which completes the proof of Proposition 3.2. □

Remark 3.3. Note that $\gamma_t^K(j)$ is monotonically decreasing in j for fixed K, t (because $j \to \binom{K-j}{t-1}$ is monotonically decreasing in j). This corresponds to the intuition that the RtG procedure returns a coordinate j with a larger magnitude of a_j, with higher probability.

For most cases of interest, the dimension K of the input vector is very large. When K is asymptotically large, it is convenient to consider the distribution of the quantile $q = j/K$ (where $0 < q < 1$), instead of j. The probability distribution of this quantile evaluated at j/K is given by $K\gamma_t^K(j)$. The following proposition states that $K\gamma_t^K(j)$ asymptotically converges to $t(1-q)^{t-1}$, the probability density function of the Beta distribution with shape parameters $(1, t)$, i.e., Beta$(1, t)$.

Proposition 3.3. *We have the following limit for a fixed $q \in (0, 1)$:*

$$\lim_{j,K\to\infty,\ j/K=q} K\gamma_t^K(j) = t(1-q)^{t-1} \ .$$

Proof. By using the expression of $\gamma_t^K(j)$ and canceling out the factorials, it holds

[5] In case of ties, we choose the smallest index.

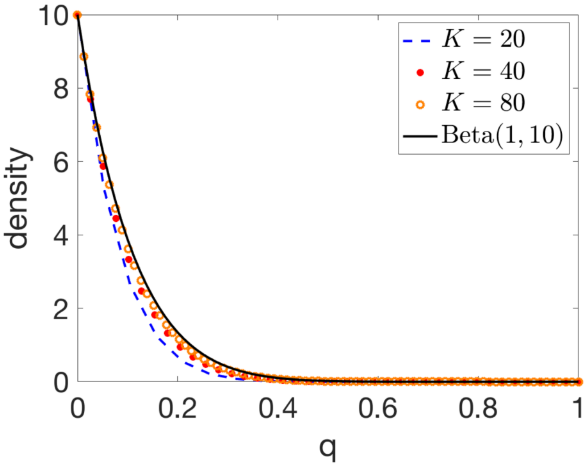


FIG. 2. *Figure shows the profiles of $K\gamma_t^K(j)$ (i.e., the probability distribution of the quantile $q = j/K$ for the RtG procedure, as described in the text) as a function of q. We consider three profiles (of $K\gamma_t^K(j)$) for three different values of K, and the Beta$(1, 10)$ density function (we fix $t = 10$). We observe that for $K \approx 40$, the profile of $K\gamma_t^K(j)$ and that of the Beta$(1, 10)$ distribution are almost indistinguishable.*

that

$$\begin{aligned}\gamma_t^K(j) = \frac{\binom{K-j}{t-1}}{\binom{K}{t}} &= \frac{t}{K}\frac{(K-t)(K-t-1)\cdots(K-j-t+2)}{(K-1)(K-2)\cdots(K-j+1)}\\ &= \frac{t}{K}\left(1-\frac{t-1}{K-1}\right)\left(1-\frac{t-1}{K-2}\right)\cdots\left(1-\frac{t-1}{K-j+1}\right).\end{aligned}$$

Denote $A_t^K(j) = \left(1-\frac{t-1}{K-1}\right)\left(1-\frac{t-1}{K-2}\right)\cdots\left(1-\frac{t-1}{K-j+1}\right)$. Then it holds that

$$\begin{aligned}\lim_{j,K\to\infty,\ j/K=q} \ln A_t^K(j) &= \lim_{j,K\to\infty,\ j/K=q} \sum_{l=1}^{j-1} \ln\left(1-\frac{t-1}{K-l}\right)\\ &= \lim_{j,K\to\infty,\ j/K=q} \sum_{l=1}^{j-1} -\frac{t-1}{K-l}\\ &= \lim_{j,K\to\infty,\ j/K=q} (t-1)\ln\left(\frac{K-j}{K}\right)\\ &= (t-1)\ln(1-q)\ ,\end{aligned}$$

where the second equality uses $\ln\left(1-\frac{t-1}{K-l}\right) \approx -\frac{t-1}{K-l}$ and the third equality is from $\sum_{l=1}^{j-1}\frac{1}{K-l} \approx \ln K - \ln(K-j) = \ln(\frac{K}{K-j})$, when both j, K are large and $j/K \approx q$. Therefore,

$$\lim_{j,K\to\infty,j/K=q} K\gamma_t^K(j) = t \lim_{j,K\to\infty,j/K=q} \exp\left(\ln A_t^{(K,j)}\right) = t(1-q)^{t-1}\ ,$$

which completes the proof. □

Figure 2 compares the probability distribution of the discrete random variable j/K and its continuous limit: as soon as $K \approx 40$, the function $K\gamma_t^K(j)$ becomes (almost) identical to the Beta density.

Given a partition $\{I_g\}_{g\in[G]}$ of $[K]$, let us denote for every $g \in [G]$

$$b_g = \max_{j\in I_g} |a_j| \quad \text{and} \quad k_g = \arg\max_{j\in I_g} |a_j| \ . \tag{11}$$

For the RtG procedure with Type 2 random selection rule, note that $\mathbb{P}(\hat{j} = k_g) = 1/G$ for all $g \in [G]$. The Type 3 selection rule is a combination of the Type 1 and Type 2 selection rules. One can view the RtG procedure with Type 3 selection rule as a two-step procedure: (i) compute b_g and k_g as in (11); and (ii) use an RtG procedure with Type 1 rule on $\{b_g\}_{g\in[G]}$. Using an argument similar to that used in Proposition 3.2, we have

$$\mathbb{P}(\hat{j} = k_{(g)}) = \gamma_t^G(g) \ , \tag{12}$$

where we recall that $|a_{k_{(1)}}| \geq |a_{k_{(2)}}| \geq \cdots \geq |a_{k_{(G)}}|$ and $b_{(g)} = |a_{k_{(g)}}|$ for all g.

The following proposition establishes a connection among the four types of selection rules and the four norms described in section 3.1.

Proposition 3.4. *Consider the RtG procedure for finding the approximate maximum of the absolute values of a. It holds that*

$$\mathbb{E}[|a_{\hat{j}}|] = \|a\|_{\mathcal{F}} \ ,$$

where $\mathcal{F}$ denotes infinity norm, the ordered ℓ_1 norm with parameter $\gamma = [\gamma_t^K(j)]_j$, the $\ell_{1,\infty}$ group norm, or the ordered mixed norm with parameter $\gamma = [\gamma_t^G(j)]_j$ when the selection rule is Type 0, *Type* 1, *Type* 2, *or Type* 3 *(cf. section* 1.2*), respectively.*

Proof.

Type 0: This corresponds to the deterministic case and $|a_{\hat{j}}| = \max_j |a_j| = \|a\|_\infty$.

Type 1: It follows from Proposition 3.2 that $\mathbb{P}(\hat{j} = (j)) = \gamma_t^K(j)$, and thus

$$\mathbb{E}[|a_{\hat{j}}|] = \sum_{j=1}^{K} \gamma_t^K(j) |a_{(j)}| = \|a\|_{\mathcal{S}} \ .$$

Type 2: For the Type 2 random selection rule, we have $\mathbb{P}(\hat{j} = k_g) = \frac{1}{G}$ for any $g \in [G]$, and thus

$$\mathbb{E}[|a_{\hat{j}}|] = \frac{1}{G}\sum_{g=1}^{G} b_g = \frac{1}{G}\sum_{g=1}^{G} \|a_{I_g}\|_\infty = \|a\|_{\mathcal{G}} \ .$$

Type 3: It follows from (12) that

$$\mathbb{E}[|a_{\hat{j}}|] = \sum_{g=1}^{G} \gamma_t^G(g) b_{(g)} = \sum_{g=1}^{G} \gamma_t^G(g) \|a_{I_{(g)}}\|_\infty = \|a\|_{\mathcal{C}} \ . \qquad \square$$

4. Computational guarantees for RGBM.

Here we derive computational guarantees for RGBM. We first introduce some standard regularity/continuity conditions on the scalar loss function $\ell(y, f)$ that we require in our analysis.

Definition 4.1. *We denote by $\partial \ell(y, f)/\partial f$ the derivative of the scalar loss function ℓ with respect to the prediction f. We say that ℓ is σ-smooth if for any y and predictions f_1 and f_2 it holds that*

$$\ell(y, f_1) \leq \ell(y, f_2) + \frac{\partial \ell(y, f_2)}{\partial f}(f_1 - f_2) + \frac{\sigma}{2}(f_1 - f_2)^2.$$

We say ℓ is μ-strongly convex (with $\mu > 0$) if for any y and predictions f_1 and f_2 it holds that

$$\ell(y, f_1) \geq \ell(y, f_2) + \frac{\partial \ell(y, f_2)}{\partial f}(f_1 - f_2) + \frac{\mu}{2}(f_1 - f_2)^2.$$

We present examples of some loss functions commonly used in GBM along with their regularity/continuity parameters:

Squared ℓ_2 or least squares loss: $\ell(y, f) = \frac{1}{2}(y - f)^2$ is 1-smooth and 1-strongly convex.

Huber loss: The Huber loss function with parameter $d > 0$, given by

$$l_d(y, f) = \begin{cases} \frac{1}{2}(y - f)^2 & \text{for } |f - y| \leq d, \\ d|y - f| - \frac{1}{2}d^2 & \text{otherwise }, \end{cases}$$

is 1-smooth but not strongly convex.

Logistic loss: We consider a regularized version of the usual logistic loss function: $\ell_d(y, f) = \log(1 + e^{-yf}) + \frac{d}{2}f^2$ with $d \geq 0$, which is $(\frac{1}{4} + d)$-smooth and d-strongly convex (when $d > 0$). A special case is the usual logistic loss when $d = 0$, which is $\frac{1}{4}$-smooth but not strongly convex.

Exponential loss: $\ell(y, f) = \exp(-yf)$ is neither strongly convex nor smooth.

Notice that the objective function $L(\beta)$ has an invariant subspace in the coefficient space, namely, for any $\omega \in \text{Ker}(B)$, it holds that $L(\beta) = L(\beta + \omega)$. Let us denote

$$Z(\hat{\beta}) := \left\{ \beta \mid B\beta = B\hat{\beta} \right\} \tag{13}$$

as the invariant subspace of $\hat{\beta}$. Recall that $\mathcal{F} \in \{\infty, \mathcal{S}, \mathcal{G}, \mathcal{C}\}$ and $\mathcal{F}^*$ is the dual norm of $\mathcal{F}$ (see section 3.1). We define a distance metric in the β-space as

$$\begin{aligned} \text{Dist}^B_{\mathcal{F}^*}(\beta_1, \beta_2) := \text{Dist}_{\mathcal{F}^*}(Z(\beta_1), Z(\beta_2)) &= \min_{b \in Z(\beta_1), \hat{b} \in Z(\beta_2)} \|b - \hat{b}\|_{\mathcal{F}^*} \\ &= \min_{\omega \in \text{Ker}(B)} \|\beta_1 - \beta_2 - \omega\|_{\mathcal{F}^*} \ , \end{aligned}$$

which is the usual notion of distance between subspaces in the $\mathcal{F}^*$ norm. In particular, if $\beta_1, \beta_2 \in Z(\hat{\beta})$, then $\text{Dist}^B_{\mathcal{F}^*}(\beta_1, \beta_2) = 0$. Note that $\text{Dist}^B_{\mathcal{F}^*}$ is a pseudonorm—Proposition 4.1 lists a few properties of $\text{Dist}^B_{\mathcal{F}^*}$.

PROPOSITION 4.1.

1. $\text{Dist}^B_{\mathcal{F}^*}(\beta_1, \beta_2)$ *is symmetric: i.e., for any β_1 and β_2, we have*

$$\text{Dist}^B_{\mathcal{F}^*}(\beta_1, \beta_2) = \text{Dist}^B_{\mathcal{F}^*}(\beta_2, \beta_1) \ .$$

2. $\text{Dist}^B_{\mathcal{F}^*}(\beta_1, \beta_2)$ *is translation invariant: i.e., for any β_1, β_2 and $\hat{\beta}$, we have*

$$\text{Dist}^B_{\mathcal{F}^*}(\beta_1 - \hat{\beta}, \beta_2 - \hat{\beta}) = \text{Dist}^B_{\mathcal{F}^*}(\beta_1, \beta_2) \ .$$

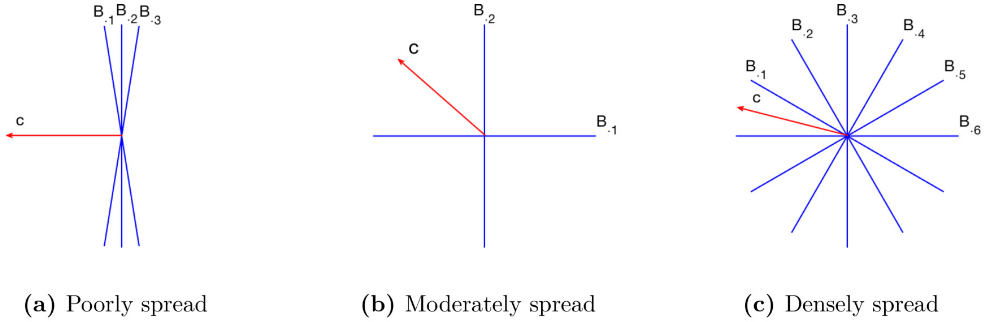
Proof.

1. The proof of this part follows from

$$\text{Dist}^B_{\mathcal{F}^*}(\beta_1, \beta_2) = \min_{b \in Z(\beta_1), \hat{b} \in Z(\beta_2)} \|b - \hat{b}\|_{\mathcal{F}^*} = \min_{b \in Z(\beta_1), \hat{b} \in Z(\beta_2)} \|\hat{b} - b\|_{\mathcal{F}^*} = \text{Dist}^B_{\mathcal{F}^*}(\beta_2, \beta_1) \ .$$

2. The proof of this part follows from

$$\begin{aligned} \text{Dist}^B_{\mathcal{F}^*}(\beta_1 - \hat{\beta}, \beta_2 - \hat{\beta}) &= \min_{\omega \in \text{Ker}(B)} \|(\beta_1 - \hat{\beta}) - (\beta_2 - \hat{\beta}) - \omega\|_{\mathcal{F}^*} \\ &= \min_{\omega \in \text{Ker}(B)} \|\beta_1 - \beta_2 - \omega\|_{\mathcal{F}^*} = \text{Dist}^B_{\mathcal{F}^*}(\beta_1, \beta_2) \ . \qquad \square \end{aligned}$$



(a) Poorly spread **(b)** Moderately spread **(c)** Densely spread

Fig. 3. *Illustration of the relationship between* Θ_∞ *and density of weak learners in a 2D example. Figures in panels* (a), (b), *and* (c) *represent poorly spread weak learners, moderately spread weak learners, and densely spread weak learners, respectively. When* $\mathcal{F}$ *is the infinity norm, the values of* Θ_∞ *are given by* (a) $\Theta_\infty^2 \approx 0$; (b) $\Theta_\infty^2 = 1/2$; *and* (c) $\Theta_\infty^2 \approx 0.933$*—the weak learners are more spread out for higher values of* Θ_∞.

4.1. Minimal Cosine Angle. Here we introduce a novel geometric quantity, the Minimal Cosine Angle $\Theta_\mathcal{F}$, which measures the density of the collection of weak learners in the prediction space with respect to the $\mathcal{F}$ norm. We show here that Minimal Cosine Angle plays a key role in the computational guarantees for RGBM.

Definition 4.2. *The Minimal Cosine Angle (MCA) of a set of weak learners (given by the columns of the matrix* B*) with respect to the* $\mathcal{F}$ *norm is defined as*

$$\Theta_\mathcal{F} := \min_{c\in \text{Range(B)}} \left\| [\cos(B_{:j}, c)]_{j=1,\ldots,K} \right\|_\mathcal{F} . \tag{14}$$

Remark 4.1. At first look, the MCA quantity seems to be similar to the Cheung–Cucker condition number for solving a linear system. See [8, 10] for details on the Cheung–Cucker condition number, and [32, 33] for how it connects to margins and the convergence rate for iterative algorithms (e.g., the Perceptron and von Neumann algorithms) arising in binary classification tasks. However, there is an important basic difference: Our measure MCA looks at the *columns* of the basis matrix B, whereas the Cheung–Cucker condition number is based on the *rows* of B.

The quantity $\Theta_\mathcal{F}$ measures the "density" of the weak learners in the prediction space. Figure 3 provides an illustration in a simple 2D example when $\mathcal{F}$ is the infinity norm. Given weak learners $B_{:1}, \ldots, B_{:K}$, we compute the cosine of the angle between each weak learner and a direction c. The $\mathcal{F}$ norm can be viewed as an approximation to the infinity norm, which is the norm corresponding to traditional GBM. The quantity MCA refers to the minimum (over all directions indexed by c) of such reweighted angles.

We next present some equivalent definitions of $\Theta_\mathcal{F}$.

Proposition 4.2.

$$\Theta_\mathcal{F} = \min_{c\in \text{Range}(B)} \frac{\|B^Tc\|_\mathcal{F}}{\|c\|_2} = \min_a \frac{\|Ba\|_2}{\text{Dist}_{\mathcal{F}^*}^B(0,a)} > 0 \ . \tag{15}$$

Proof. The first equality follows directly by rewriting (14). Notice that for any norm $\mathcal{F}$ in $\mathbb{R}^K$ (a finite-dimensional space), there exists a scalar $\gamma > 0$ such that $\|B^Tc\|_\mathcal{F} \geq \gamma \|B^Tc\|_\infty$. Thus

$$\Theta_\mathcal{F} = \min_{c\in \text{Range}(B), \|c\|_2=1} \|B^Tc\|_\mathcal{F} \geq \gamma \|B^Tc\|_\infty > 0 \ ,$$

where the second inequality follows from the observation that $c \in \text{Range}(B)$. We now proceed to show the second equality of (15).

By the definition of $\text{Dist}_{\mathcal{F}^*}^B$ and the definition of the dual norm, we have

$$\begin{aligned}\text{Dist}_{\mathcal{F}^*}^B(0,a) &= \min_{\omega\in\text{Ker}(B)} \|a-\omega\|_{\mathcal{F}^*} = \min_{\omega\in\text{Ker}(B)} \max_{\|b\|_{\mathcal{F}}\le 1} \langle a-\omega, b\rangle = \max_{\|b\|_{\mathcal{F}}\le 1} \min_{\omega\in\text{Ker}(B)} \langle a-\omega, b\rangle \\ &= \max_{\|b\|_{\mathcal{F}}\le 1, b\in\text{Range}(B^T)} \langle a,b\rangle = \max_{\|b\|_{\mathcal{F}}\le 1, b\in\text{Range}(B^T)} |\langle a,b\rangle| = \max_{b\in\text{Range}(B^T)} \frac{|\langle a,b\rangle|}{\|b\|_{\mathcal{F}}} \ ,\end{aligned}$$

where the third equality uses von Neumann's Minimax Theorem, and the fourth equality is based on the observation

$$\min_{\omega\in\text{Ker}(B)} \langle a-\omega, b\rangle = \begin{cases} -\infty & \text{for } b \notin \text{Range}(B^T)\,, \\ \langle a,b\rangle & \text{for } b \in \text{Range}(B^T)\,. \end{cases}$$

Therefore,

$$\min_a \frac{\|Ba\|_2}{\text{Dist}_{\mathcal{F}^*}^B(0,a)} = \min_{b\in\text{Range}(B^T),a} \frac{\|Ba\|_2\|b\|_{\mathcal{F}}}{|\langle a,b\rangle\,|} \ .$$

Denote $P_B = B^T(BB^T)^\dagger B$ as the projection matrix onto $\text{Range}(B^T)$. Then we have $P_B b = b$ for any $b \in \text{Range}(B^T)$. Thus

$$\min_a \frac{\|Ba\|_2}{\text{Dist}_{\mathcal{F}^*}^B(0,a)} = \min_{b\in\text{Range}(B^T),a} \frac{\|Ba\|_2\|b\|_{\mathcal{F}}}{|\,\langle a, P_B b\rangle\,|} = \min_{b\in\text{Range}(B^T),a} \frac{\|Ba\|_2\|b\|_{\mathcal{F}}}{|\,\langle Ba, (BB^T)^\dagger Bb\rangle\,|} \ . \tag{16}$$

Now denote $c = (BB^T)^\dagger Bb$. Then $c \in \text{Range}(B)$ and $B^Tc = P_Bb = b$. Note that for any a, we have $\|Ba\|_2\|c\|_2 \ge |\langle Ba, c\rangle|$, which implies

$$\min_a \frac{\|Ba\|_2}{|\,\langle Ba, c\rangle\,|} \ge \frac{1}{\|c\|_2} \ .$$

Since $c \in \text{Range}(B)$, there exists a vector a satisfying $Ba = c$, which leads to

$$\frac{\|Ba\|_2}{|\,\langle Ba, c\rangle\,|} = \frac{\|c\|_2}{\|c\|_2^2} = \frac{1}{\|c\|_2} \ ,$$

from which it follows that

$$\min_a \frac{\|Ba\|_2}{|\,\langle Ba, c\rangle\,|} = \frac{1}{\|c\|_2} \ . \tag{17}$$

Substituting $c = (BB^T)^\dagger Bb$ and combining (16) and (17) yields

$$\min_a \frac{\|Ba\|_2}{\text{Dist}_{\mathcal{F}^*}^B(0,a)} = \min_{c\in\text{Range}(B)} \frac{\|B^Tc\|_{\mathcal{F}}}{\|c\|_2} \ ,$$

which completes the proof. □

To gain additional intuition about MCA, we consider some examples.

Example 1 (orthogonal basis with infinity norm). Suppose $\mathcal{F}$ is the infinity norm and the set of weak learners in $\mathbb{R}^p$ forms an orthogonal basis (e.g., the discrete Fourier basis in $\mathbb{R}^p$). Then $\Theta_\infty = 1/\sqrt{p}$.

Example 2 (orthogonal basis with ordered ℓ_1 norm). Suppose $\mathcal{F}$ is the ordered ℓ_1 norm with a parameter sequence $\gamma \in \mathbb{R}^p$ and the set of weak learners in $\mathbb{R}^p$ forms an orthogonal basis. Then

$$\Theta_{\mathcal{S}} = \min\left\{\gamma_1, \frac{1}{\sqrt{2}}(\gamma_1+\gamma_2), \ldots, \frac{1}{\sqrt{p}}(\gamma_1+\cdots+\gamma_p)\right\} . \tag{18}$$

We present a proof for (18)—note that the result for Example 1 follows as a special case. Without loss of generality, we assume B to be an identity matrix. It then follows from the second equality of (15) that

$$\Theta_{\mathcal{S}} = \min_{\|a\|_{\mathcal{S}^*}=1} \|a\|_2 . \tag{19}$$

By flipping the constraint and the objective function of (19) we can instead consider the equivalent problem

$$\Phi = \max_{\|a\|_2=1} \|a\|_{\mathcal{S}^*} = \max_{\|a\|_2\le 1} \|a\|_{\mathcal{S}^*} ,$$

and we have $\Theta_{\mathcal{S}} = 1/\Phi$. Using the definition of the dual of the ordered ℓ_1 norm (see (9)), notice that for any $i \in [p]$, it follows from the ℓ_1-ℓ_2 norm inequality that

$$\sum_{j=1}^{i} |a_{(j)}| \le \sqrt{i\left(\sum_{j=1}^{i} a_{(j)}^2\right)} \le \sqrt{i}\|a\|_2 \le \sqrt{i} ,$$

and therefore

$$\Phi = \max_{\|a\|_2\le 1}\max_{i\in[p]}\left\{\frac{\sum_{j=1}^{i}|a_{(j)}|}{\sum_{j=1}^{i}\gamma_j}\right\} \le \max_{i\in[p]}\left\{\frac{\sqrt{i}}{\sum_{j=1}^{i}\gamma_j}\right\} .$$

For any $i \in [p]$ define $\tilde{a}_1 = \cdots = \tilde{a}_i = 1/\sqrt{i}$ and $\tilde{a}_{i+1} = \cdots = \tilde{a}_p = 0$. Then we have $\Phi \ge \|\tilde{a}\|_{\mathcal{S}^*} = \sqrt{i}/(\sum_{j=1}^{i}\gamma_j)$. Therefore, we have

$$\Phi = \max_{i\in[p]}\left\{\frac{\sqrt{i}}{\sum_{j=1}^{i}\gamma_j}\right\} = \frac{1}{\Theta_{\mathcal{S}}} ,$$

which completes the proof of (18).

Remark 4.2. Consider using a Type 1 random selection rule in RGBM. Then the corresponding norm $\mathcal{F}$ is the ordered ℓ_1 norm with parameter $\gamma_t^p = [\gamma_t^p(j)]_j$ as defined in (10). Figure 4 shows the value of $\Theta_{\mathcal{S}}$ (computed by formula (18)) versus the dimension p—we consider different values of t and use an orthogonal basis. The figure suggests that $\Theta_{\mathcal{S}}$ depends upon p, t as follows:

$$\Theta_{\mathcal{S}} \sim \begin{cases} \frac{1}{\sqrt{p}} & \text{if } p \le t^2 , \\ \frac{t}{p} & \text{otherwise} . \end{cases}$$

Example 3 (binary basis with infinity norm). Suppose $\mathcal{F}$ is the infinity norm, and the basis matrix B has entries $B_{i,j} \in \{-1, 0, 1\}$—leading to 3^p different weak learners. In this case,

$$\Theta_\infty = \frac{1}{\sqrt{1^2 + (\sqrt{2}-1)^2 + \cdots + (\sqrt{p}-\sqrt{p-1})^2}} \propto \frac{1}{\sqrt{\ln p}} . \tag{20}$$

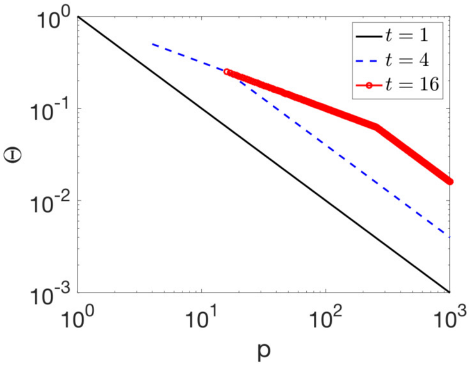


FIG. 4. *Plot shows how $\Theta_{\mathcal{S}}$ varies with p (*log-log *plot) when the weak learners are orthogonal and $\mathcal{F}$ corresponds to the ordered ℓ_1 norm with parameter $\gamma = [\gamma_t^p(j)]_j$ (see* (10)*). We show three profiles for three different values of t. Note that $\Theta_{\mathcal{S}}$ is defined only for $p \geq t$. The setup is described in Remark* 4.2.

We present a proof for (20). Since $B_{i,j} \in \{-1, 0, 1\}$, we have

$$\begin{aligned}\Theta_\infty &= \min_c \max_j |\cos(B_{:j}, c)| = \min_c \max_{i\in[p]} \max_{\|B_{:j}\|_1 = i} |\cos(B_{:j}, c)| \\ &= \min_c \max_{i\in[p]} \frac{\sum_{k=1}^i |c_{(k)}|}{\sqrt{i}\|c\|_2} \,. \end{aligned} \tag{21}$$

Let us recall the form of $\mathcal{S}^*$, i.e., the dual ordered ℓ_1 norm appearing in Proposition 3.1. Observe that $\max_{i\in[p]}(\sum_{k=1}^i |c_{(k)}|/\sqrt{i}) = \|c\|_{\mathcal{S}^*}$, where $\gamma = [\sqrt{i} - \sqrt{i-1}]_{i\in[p]}$ is the parameter of the ordered ℓ_1 norm $\mathcal{S}$. Thus

$$\Theta_\infty = \min_c \frac{\|c\|_{\mathcal{S}^*}}{\|c\|_2} = \min_a \frac{\|a\|_2}{\|a\|_{\mathcal{S}}} = \min_{\|a\|_{\mathcal{S}}=1} \|a\|_2 \,,$$

where the second equality uses (15) with $\mathcal{F} = \mathcal{S}^*$ and B as the identity matrix. By flipping the constraint and the objective function, we can instead consider the equivalent problem

$$\Phi = \max_{\|a\|_2=1} \|a\|_{\mathcal{S}} = \max_{\|a\|_2\leq 1} \|a\|_{\mathcal{S}} \,,$$

with $\Theta_\infty = 1/\Phi$. By the Cauchy–Schwarz inequality, it holds that

$$\|a\|_S^2 = \left(\sum_{i=1}^p \gamma_i |a_{(i)}|\right)^2 \leq \left(\sum_{i=1}^p \gamma_i^2\right) \|a\|_2^2 = \left(\sum_{i=1}^p \left(\sqrt{i} - \sqrt{i-1}\right)^2\right) \|a\|_2^2 \,,$$

with equality being achieved when $a \propto [\sqrt{i} - \sqrt{i-1}]_i$. Thus we have that $\Phi = \sqrt{\sum_{i=1}^p \left(\sqrt{i} - \sqrt{i-1}\right)^2}$ and $\Theta_\infty = 1/\Phi$. Notice that

$$\frac{1}{4}\sum_{i=1}^p \frac{1}{i} \leq \sum_{i=1}^p \left(\sqrt{i} - \sqrt{i-1}\right)^2 = \sum_{i=1}^p \frac{1}{\left(\sqrt{i} + \sqrt{i-1}\right)^2} \leq 1 + \frac{1}{4}\sum_{i=2}^p \frac{1}{i-1} \,,$$

where the left- and right-hand sides of the above are both $O(\ln p)$. This implies that $\sum_{i=1}^p \left(\sqrt{i} - \sqrt{i-1}\right)^2 \propto \ln p$, thereby completing the proof. ∎

Remark 4.3. The binary basis described in Example 3 (with $\Theta_\infty = O(1/\sqrt{\ln p})$) is more densely distributed in the prediction space when compared to Example 1 (with $\Theta_\infty = O(1/\sqrt{p})$)—see Figures 3(b) and 3(c) for a schematic illustration.

4.2. Computational guarantees: Strongly convex loss function. We establish computational guarantees for RGBM when the scalar loss function ℓ is both smooth and strongly convex. Let $\mathbb{E}_m$ denote the expectation over the random selection scheme at iteration m, conditional on the selections up to iteration $m-1$. Let $\mathbb{E}_{\xi_m}$ denote the expectation over the random selection scheme up to (and including) iteration m. The following theorem presents the linear convergence rate for RGBM.

THEOREM 4.1. *Let ℓ be μ-strongly convex and σ-smooth. Consider RGBM (Algorithm 2) or RtGCD (Algorithm 3) with either a line-search step-size rule or constant step-size rule with $\rho = 1/\sigma$. If $\Theta_{\mathcal{F}}$ denotes the value of the corresponding MCA, then for all $M \geq 0$ we have*

$$\mathbb{E}_{\xi_M}[L(\beta^M) - L(\beta^*)] \leq \left(1 - \tfrac{\mu}{\sigma}\Theta_{\mathcal{F}}^2\right)^M \left(L(\beta^0) - L(\beta^*)\right). \tag{22}$$

Notice that in the special case when J is chosen deterministically as the set of all weak learners, Theorem 4.1 leads to a linear convergence rate for GBM [15]. Some prior works have also presented a linear convergence rate for GBM, but our results are different. For example, [39] shows a linear convergence rate, but the constant is exponential in the number of features p, except for the exponential loss.[6] [11] presents a linear convergence rate for LS-Boost (GBM with a least squares loss function) of the form $O(\tau^M)$, where the parameter $\tau = 1 - \lambda_{\text{pmin}}(B^TB)/4K$ depends upon $\lambda_{\text{pmin}}(A)$, the minimal nonzero eigenvalue of a matrix A. In GBM, K is usually exponentially large, and thus τ can be close to one. The linear convergence constant derived herein (i.e., $1 - \frac{\mu}{\sigma}\Theta_{\mathcal{F}}^2$) has a superior dependence on the number of weak learners, and it stays away from 1 as K becomes large. We obtain an improved rate since we employ a different analysis technique based on MCA.

Remark 4.4. We study the convergence rate of RGBM as a function of t using the same setup considered in Remark 4.2. Using an "epoch" (i.e., the cost to evaluate all weak learners across all samples) as the unit of computational cost, the cost per iteration of RGBM is t/p epochs. Then the (multiplicative) improvement per epoch is

$$\left(1 - \frac{\mu}{\sigma}\Theta_{\mathcal{S}}^2\right)^{p/t} \sim \begin{cases} \left(1 - \frac{\mu}{p\sigma}\right)^{p/t} & \text{if } t \geq \sqrt{p}, \\ \left(1 - \frac{t^2\mu}{p^2\sigma}\right)^{p/t} & \text{otherwise.} \end{cases}$$

This suggests that we should choose $t \sim \sqrt{p}$ when the weak learners are almost orthogonal. Recall that from a coordinate descent perspective, RtGCD with $t = 1$ leads to random CD, and RtGCD with $t = p$ leads to greedy CD. Choosing t to be larger than $O(\sqrt{p})$ will not lead to any improvement in the theoretical convergence rate, though it will lead to an increase in computational cost.

Remark 4.5. Since traditional GBM is equivalent to greedy CD in the coefficient space, theoretical guarantees of greedy CD can be used to analyze GBM. In this case, however, the resulting computational guarantees may contain the total number of weak learners K—the bounds we present here are of a different flavor (they depend upon MCA).

Recently, interesting techniques have been proposed to improve the efficiency of greedy CD. For example, [37] proposed a scheme to approximate the entire gradient vector and use it to update the coordinates (in the spirit of approximate steepest

[6] The result of [39] for the exponential loss function is superior to that presented here, as their analysis is targeted towards this loss function.

CD). It will be interesting to adapt the ideas from [37] to the case of Boosting-like algorithms presented herein.

Propositions 4.3–4.5 presented below will be needed for the proof of Theorem 4.1. Proposition 4.3 establishes a relationship among the four selection rules for choosing subset J in RGBM (Algorithm 3) and the norms introduced in section 3.1.

PROPOSITION 4.3. *Consider Algorithm* 3 *with the four types of selection rules for choosing the set J as described in section* 1.2.1*. For any iteration index m, we have*

$$\mathbb{E}_m\left[(\nabla_{j_m}L(\beta^m))^2\right] = \left\|\left[\nabla_j L(\beta^m)^2\right]_j\right\|_{\mathcal{F}} \geq \|\nabla L(\beta^m)\|_{\mathcal{F}}^2 \ ,$$

where $\mathcal{F}$ is the infinity norm, the ordered ℓ_1 norm with parameter $\gamma = [\gamma_t^K(j)]_j$, the $\ell_{1,\infty}$ group norm, or the ordered mixed norm with parameter $\gamma = [\gamma_t^G(j)]_j$ when the selection rule is Type 0*, Type* 1*, Type* 2*, or Type* 3*, respectively.*

Proof. The equality is a direct result of Proposition 3.4 with $a_j = (\nabla_j L(\beta^m))^2$. Notice that the $\mathcal{F}$ norm of a is a weighted sum of its coordinates—for notational convenience, we denote these weights by a vector $\lambda \in \mathbb{R}^K$ that satisfies $\left\|\left[\nabla L_j(\beta^m)^2\right]_j\right\|_{\mathcal{F}} = \sum_j \lambda_j \left(\nabla_j L(\beta^m)\right)^2$ and $\lambda_j \geq 0, j \in [K]$, $\sum_j \lambda_j = 1$. Thus we have

$$\begin{aligned}\left\|\left[\nabla L_j(\beta^m)^2\right]_j\right\|_{\mathcal{F}} &= \left(\sum_j \lambda_j\right)\left(\sum_j \lambda_j \left(\nabla_j L(\beta^m)\right)^2\right) \geq \left(\sum_j \lambda_j \left|\nabla_j L(\beta^m)\right|\right)^2 \\ &= \|\nabla L(\beta^m)\|_{\mathcal{F}}^2 \ ,\end{aligned}$$

where the inequality above follows from the Cauchy–Schwarz inequality. ∎

The following proposition can be viewed as a generalization of the mean-value inequality.

PROPOSITION 4.4. *For $a \in \text{Range}(B^T)$ and $t > 0$, it holds that*

$$\min_\beta \left\{\langle a, \beta - \beta^*\rangle + \frac{t}{2}\text{Dist}_{\mathcal{F}^*}^B(\beta, \beta^*)^2\right\} = -\frac{1}{2t}\|a\|_{\mathcal{F}}^2 \ .$$

Proof. Let $b = \beta - \beta^*$, $\text{Ker}(B) = \{\omega \mid B\omega = 0\}$, and $c = b + \omega$. By the definition of $\text{Dist}_{\mathcal{F}^*}^B$, we have

$$\min_\beta \left\{\langle a, \beta - \beta^*\rangle + \frac{t}{2}\text{Dist}_{\mathcal{F}^*}^B(\beta, \beta^*)^2\right\} = \min_b \min_{\omega \in \text{Ker}(B)} \left\{\langle a, b\rangle + \frac{t}{2}\|b + \omega\|_{\mathcal{F}^*}^2\right\}$$

$$= \min_{\omega \in \text{Ker}(B)} \left\{-\langle a, \omega\rangle + \min_{b+\omega} \langle a, b + \omega\rangle + \frac{t}{2}\|b+\omega\|_{\mathcal{F}^*}^2\right\} = \min_{\omega \in \text{Ker}(B)} \min_c \left\{\langle a, c\rangle + \frac{t}{2}\|c\|_{\mathcal{F}^*}^2\right\}$$

$$= \min_c \left\{\langle a, c\rangle + \frac{t}{2}\|c\|_{\mathcal{F}^*}^2\right\},$$

where the third equality considers $a \in \text{Range}(B^T)$ and makes use of the observation that $\langle a, \omega\rangle = 0$ for $\omega \in \text{Ker}(B)$. Notice that

$$\frac{t}{2}\|c\|_{\mathcal{F}^*}^2 + \frac{1}{2t}\|a\|_{\mathcal{F}}^2 \geq \|c\|_{\mathcal{F}^*}\|a\|_{\mathcal{F}} \geq |\langle a, c\rangle|$$

and hence $\min_c \left\{\langle a,c\rangle + \frac{t}{2}\|c\|^2_{\mathcal{F}^*}\right\} \le -\frac{1}{2t}\|a\|^2_{\mathcal{F}}$. Now, if $\hat{c} = \frac{1}{t}\|a\|_{\mathcal{F}} \arg\min_{\|c\|_{\mathcal{F}^*}\le 1} \langle a,c\rangle$, we have

$$\|\hat{c}\|_{\mathcal{F}^*} = \tfrac{1}{t}\|a\|_{\mathcal{F}} \qquad \text{and} \qquad \langle a,\hat{c}\rangle = -\tfrac{1}{t}\|a\|_{\mathcal{F}} \max_{\|c\|_{\mathcal{F}^*}\le 1} \langle a,c\rangle = -\tfrac{1}{t}\|a\|^2_{\mathcal{F}} \ ,$$

whereby $\langle a,\hat{c}\rangle + \frac{t}{2}\|\hat{c}\|^2_{\mathcal{F}^*} = -\frac{1}{2t}\|a\|^2_{\mathcal{F}}$. Therefore it holds that

$$\min_c \left\{\langle a,c\rangle + \tfrac{t}{2}\|c\|^2_{\mathcal{F}^*}\right\} = -\tfrac{1}{2t}\|a\|^2_{\mathcal{F}} \ ,$$

which completes the proof. □

PROPOSITION 4.5. *If ℓ is μ-strongly convex, it holds for any β and $\hat{\beta}$ that*

$$L(\hat{\beta}) \ge L(\beta) + \left\langle \nabla L(\beta), \hat{\beta}-\beta \right\rangle + \frac{1}{2}\mu\Theta^2_{\mathcal{F}} \mathrm{Dist}^B_{\mathcal{F}^*}(\hat{\beta},\beta) \ .$$

Proof. Since ℓ is μ-strongly convex, we have

$$\begin{aligned} L(\hat{\beta}) &= \sum_{i=1}^n \ell(y_i, B_{i:}\hat{\beta}) \\ &\ge \sum_{i=1}^n \left\{ \ell(y_i, B_{i:}\beta) + \frac{\partial \ell(y_i, B_{i:}\hat{\beta})}{\partial f}\langle B_{i:}, \hat{\beta}_i - \beta_i\rangle + \tfrac{\mu}{2}\|B_{i:}\|_2^2(\hat{\beta}_i-\beta_i)^2 \right\} \\ &= L(\beta) + \langle \nabla L(\beta), \hat{\beta}-\beta\rangle + \tfrac{\mu}{2}\|B(\hat{\beta}-\beta)\|_2^2 \\ &\ge L(\beta) + \langle \nabla L(\beta), \hat{\beta}-\beta\rangle + \tfrac{\mu\Theta^2_{\mathcal{F}}}{2}\mathrm{Dist}^B_{\mathcal{F}^*}(0,\hat{\beta}-\beta)^2 \\ &= L(\beta) + \langle \nabla L(\beta), \hat{\beta}-\beta\rangle + \tfrac{\mu\Theta^2_{\mathcal{F}}}{2}\mathrm{Dist}^B_{\mathcal{F}^*}(\hat{\beta},\beta)^2 \ , \end{aligned} \tag{23}$$

where the second inequality follows from Proposition 4.2, and the last equality utilizes the symmetry and translation invariance of $\mathrm{Dist}^B_{\mathcal{F}^*}$ (Proposition 4.1). □

Proof of Theorem 4.1. For either the line-search step-size rule or the constant step-size rule, it holds that

$$\begin{aligned} L(\beta^{m+1}) &\le L(\beta^m - \tfrac{1}{\sigma}\nabla_{j_m}L(\beta^m)e_{j_m}) \\ &\le L(\beta^m) - \tfrac{1}{\sigma}\nabla_{j_m}L(\beta^m)\langle \nabla L(\beta^m), e_{j_m}\rangle + \tfrac{1}{2\sigma}\|\nabla_{j_m}L(\beta^m)e_{j_m}\|^2 \\ &= L(\beta^m) - \tfrac{1}{\sigma}\left(\nabla_{j_m}L(\beta^m)\right)^2 + \tfrac{1}{2\sigma}\left(\nabla_{j_m}L(\beta^m)\right)^2 \\ &= L(\beta^m) - \tfrac{1}{2\sigma}\left(\nabla_{j_m}L(\beta^m)\right)^2 \ , \end{aligned} \tag{24}$$

where the second inequality uses the fact that the loss function ℓ is σ-smooth. Thus $L(\beta^{m+1}) \le L(\beta^m)$ with probability one. As a result of Proposition 4.3, taking expectation over both sides of (24) with respect to $\mathbb{E}_{m+1}$ yields

$$\mathbb{E}_{m+1}[L(\beta^{m+1})] \le L(\beta^m) - \tfrac{1}{2\sigma}\|\nabla L(\beta^m)\|^2_{\mathcal{F}} \ . \tag{25}$$

Meanwhile, it follows from Proposition 4.5 that

$$\begin{aligned} L(\beta^*) &= \min_\beta L(\beta) \\ &\geq \min_\beta \left[L(\beta^m) + \langle \nabla L(\beta^m), \beta - \beta^m \rangle + \tfrac{\mu\Theta_{\mathcal{F}}^2}{2} \mathrm{Dist}_{\mathcal{F}^*}^B(\beta, \beta^m) \right] \qquad (26) \\ &= L(\beta^m) - \tfrac{1}{2\mu\Theta_{\mathcal{F}}^2} \|\nabla L(\beta^m)\|_{\mathcal{F}}^2 \ , \end{aligned}$$

where the last equality utilizes Proposition 4.4. Note that (26) together with (25) leads to

$$\mathbb{E}_{m+1}[L(\beta^{m+1})] - L(\beta^*) \leq L(\beta^m) - L(\beta^*) - \frac{1}{2\sigma}\|\nabla L(\beta^m)\|_{\mathcal{F}}^2 \leq (1 - \tfrac{\mu}{\sigma}\Theta_{\mathcal{F}}^2)(L(\beta^m) - L(\beta^*)) \ ,$$

and finally (22) follows by a telescoping argument. ∎

4.3. Computational guarantees: Non-strongly convex loss function. Define the initial level set of the loss function in the β-space (i.e., coefficient space) as

$$\mathcal{LS}_0 = \left\{ \beta \mid L(\beta) \leq L(\beta^0) \right\}$$

and its maximal distance to the optimal solution set in $\mathrm{Dist}_{\mathcal{F}^*}^B$ as

$$\mathrm{Dist}_0 = \max_{\beta \in \mathcal{LS}_0} \mathrm{Dist}_{\mathcal{F}^*}^B(\beta, \beta^*) \ .$$

Note that $\mathcal{LS}_0$ is unbounded if $Z(\beta^0)$ (cf. (13)) is unbounded. But interestingly, $\mathcal{LS}_0$ is bounded in $\mathrm{Dist}_{\mathcal{F}^*}$, i.e., $\mathrm{Dist}_0 < \infty$, when the scalar loss function ℓ has a bounded level set.

PROPOSITION 4.6. *Suppose ℓ has a bounded level set. Then* Dist_0 *is finite.*

Proof. Since the convex function ℓ has a bounded level set, the set $\{B(\beta - \beta^*) \mid \beta \in \mathcal{LS}_0\}$ is bounded. Thus there is a finite constant C such that $\max_{\beta \in \mathcal{LS}_0} \|B(\beta - \beta^*)\|_2 \leq C$. Therefore,

$$\begin{aligned} \mathrm{Dist}_0 &= \max_{\beta \in \mathcal{LS}_0} \mathrm{Dist}_{\mathcal{F}^*}^B(0, \beta - \beta^*) \\ &\leq \max_{\|B(\beta - \beta^*)\|_2 \leq C} \mathrm{Dist}_{\mathcal{F}^*}^B(0, \beta - \beta^*) \\ &= \max_{\|Ba\|_2 \leq C} \mathrm{Dist}_{\mathcal{F}^*}^B(0, a) \\ &\leq \max_{\|Ba\|_2 \leq C} \frac{\|Ba\|_2}{\Theta_{\mathcal{F}}} \\ &= \frac{C}{\Theta_{\mathcal{F}}} \ , \end{aligned}$$

where the second inequality follows from Proposition 4.2. ∎

Theorem 4.2 presents convergence guarantees (that hold in expectation over the random selection rule) for Algorithms 2 and 3 for a non-strongly convex loss function ℓ.

THEOREM 4.2. *Consider RGBM (Algorithm 2) or, equivalently, RtGCD (Algorithm 3) with either line-search step-size rule or constant step-size rule with $\rho = 1/\sigma$. If ℓ is a σ-smooth function and has a bounded level set, it holds for all $M \geq 0$ that*

$$\mathbb{E}_{\xi_M}[L(\beta^M) - L(\beta^*)] \leq \frac{1}{\frac{1}{L(\beta^0) - L(\beta^*)} + \frac{M}{2\sigma \mathrm{Dist}_0^2}} \leq \frac{2\sigma \mathrm{Dist}_0^2}{M} \ .$$

Before presenting the proof of Theorem 4.2, we present the following proposition, which is a generalization of the Cauchy–Schwarz inequality.

PROPOSITION 4.7. *For $a \in Range(B^T)$, it holds that*

$$\|a\|_{\mathcal{F}} \mathrm{Dist}^B_{\mathcal{F}^*}(\beta, \hat{\beta}) \geq \left\langle a, \beta - \hat{\beta} \right\rangle \ .$$

Proof. Assume $a = B^T s$ and let $t = \arg\min_{t \in Z(\hat{\beta})} \|\beta - t\|_{\mathcal{F}^*}$. Then it holds that

$$\|a\|_{\mathcal{F}} \mathrm{Dist}^B_{\mathcal{F}^*}(\beta, \hat{\beta}) = \|B^T s\|_{\mathcal{F}} \|\beta - t\|_{\mathcal{F}^*} \geq \left\langle B^T s, \beta - t \right\rangle = \langle s, B\beta - Bt \rangle$$

$$= \left\langle s, B\beta - B\hat{\beta} \right\rangle = \left\langle B^T s, \beta - \hat{\beta} \right\rangle = \left\langle a, \beta - \hat{\beta} \right\rangle \ . \qquad \square$$

Proof of Theorem 4.2. Recall from (25) that for both step-size rules it holds that

$$\mathbb{E}_{m+1}[L(\beta^{m+1})] \leq L(\beta^m) - \frac{1}{2\sigma} \|\nabla L(\beta^m)\|^2_{\mathcal{F}} \ . \tag{27}$$

Moreover, it follows from (24) that $L(\beta^{m+1}) \leq L(\beta^m)$ (with probability one), and thus for any iteration m, with probability one, we have $\beta^m \in \mathcal{LS}_0$. Noting that $\nabla L(\beta^m) \in \mathrm{Range}(B^T)$ and by using Proposition 4.7 we have

$$\mathbb{E}_{m+1}[L(\beta^{m+1})] \leq L(\beta^m) - \frac{\langle \nabla L(\beta^m), \beta^m - \beta^* \rangle^2}{2\sigma \mathrm{Dist}^B_{\mathcal{F}^*}(\beta^m, \beta^*)^2} \leq L(\beta^m) - \frac{\langle \nabla L(\beta^m), \beta^m - \beta^* \rangle^2}{2\sigma \mathrm{Dist}_0^2}$$

$$\leq L(\beta^m) - \frac{(L(\beta^m) - L(\beta^*))^2}{2\sigma \mathrm{Dist}_0^2} \ ,$$

where the second inequality is due to $\beta^m \in \mathcal{LS}_0$ (almost surely), and the third inequality follows from the convexity of L. Taking expectation with respect to ξ_m, we arrive at

$$\mathbb{E}_{\xi_{m+1}}[L(\beta^{m+1})] \leq \mathbb{E}_{\xi_m}[L(\beta^m)] - \frac{\mathbb{E}_{\xi_m}[(L(\beta^m) - L(\beta^*))^2]}{2\sigma \mathrm{Dist}_0^2}$$

$$\leq \mathbb{E}_{\xi_m}[L(\beta^m)] - \frac{(\mathbb{E}_{\xi_m}[L(\beta^m) - L(\beta^*)])^2}{2\sigma \mathrm{Dist}_0^2} \ .$$

Now define $\delta_m := \mathbb{E}_{\xi_m}[L(\beta^m) - L(\beta^*)]$. Then we have $\delta_m \geq 0$ and

$$\delta_{m+1} \leq \delta_m - \frac{\delta_m^2}{2\sigma \mathrm{Dist}_0^2} \ .$$

Noticing that $\delta_{m+1} = \mathbb{E}_{\xi_m}[\mathbb{E}_{m+1}[L(\beta^{m+1}) \mid \xi_m] \leq \mathbb{E}_{\xi_m}[L(\beta^m)] = \delta_m$, we have

$$\delta_{m+1} \leq \delta_m - \frac{\delta_m \delta_{m+1}}{2\sigma \mathrm{Dist}_0^2} \ .$$

Dividing both sides by $\delta_m \delta_{m+1}$, we arrive at

$$\frac{1}{\delta_{m+1}} \geq \frac{1}{\delta_m} + \frac{1}{2\sigma \mathrm{Dist}_0^2}.$$

Hence, we have that

$$\frac{1}{\delta_M} \geq \frac{1}{\delta_0} + \frac{M}{2\sigma \mathrm{Dist}_0^2} \ ,$$

which completes the proof of the theorem. $\square$

5. Numerical experiments. In this section, we present computational experiments discussing the performance of RGBM for solving classification and regression problems with tree stumps as weak learners. Our code is publicly available at https://github.com/haihaolu/RGBM.

Datasets. The datasets we use in the numerical experiments were gathered from the LIBSVM library [6]. Table 1 presents basic summary statistics of these datasets. For each dataset, we randomly choose 80% as the training dataset and the remainder as the testing dataset. In our experiments, we use the squared ℓ_2 loss for the regression problem. To be consistent with our theory (i.e., to have a strongly convex loss function), we use a regularized logistic loss with a small parameter $d = 0.0001$ for the classification problems (see section 4).

Table 1

Basic statistics of the (real) datasets used in numerical experiments. The training/testing datasets are obtained by a 80%/20% (random) split on these sample sizes.

Dataset	Task	# Samples	# Features
a9a	classification	32,561	123
colon-cancer	classification	62	2,000
rcv1	classification	20,242	47,236
YearPrediction	regression	463,715	90

RGBM with tree stumps. All algorithms consider tree stumps (see (5)) as the weak learners, as described in section 1.2. In our experiments (involving datasets with $n > 10{,}000$), to reduce the computational cost, we decrease the number of candidate splits for each feature by considering 100 quantiles instead of all n quantiles (corresponding to n samples). (We note that this simply reduces the number of weak learners considered, and our methodological framework applies.) This strategy is commonly used in implementations of GBM, e.g., XGBoost [7]. For each feature, we consider the candidate splitting points according to the percentiles of its empirical distribution; thus there are in total $100p$ weak learners. All the tree stumps that perform a split on one feature are considered as a group—thereby, resulting in p groups. In RGBM, we randomly choose t out of p features and consider the $100t$ features as the set J, among which we pick the best weak learner to perform an update. The values of t are chosen on a geometrically spaced grid from 1 to p with five values for each dataset. In particular, the case $t = p$ corresponds to traditional GBM.

Performance measures. Figure 5 shows the performance of RGBM with different t values. The x-axis is the running time (in seconds). All computations were carried out on MIT Sloan's Engaging Cluster on an Intel Xeon 2.30GHz machine (one CPU) with 10GB of RAM memory. The y-axis denotes the quality of solution (or the data fidelity) obtained, i.e., the objective value, for both the training and testing datasets.

Comparisons. For all datasets, RGBM with a medium t value leads to a model with the smallest training error with the same running time. This demonstrates the (often significant) computational gains possible by using RGBM. For datasets with $n \gg p$, the profile of the testing loss is similar to that of the training loss. The colon-cancer dataset is a high-dimensional problem with $p \gg n$, and its training/testing profile is somewhat different from the other datasets—here, the model with the best test error corresponds to a moderately large training error, and models with the

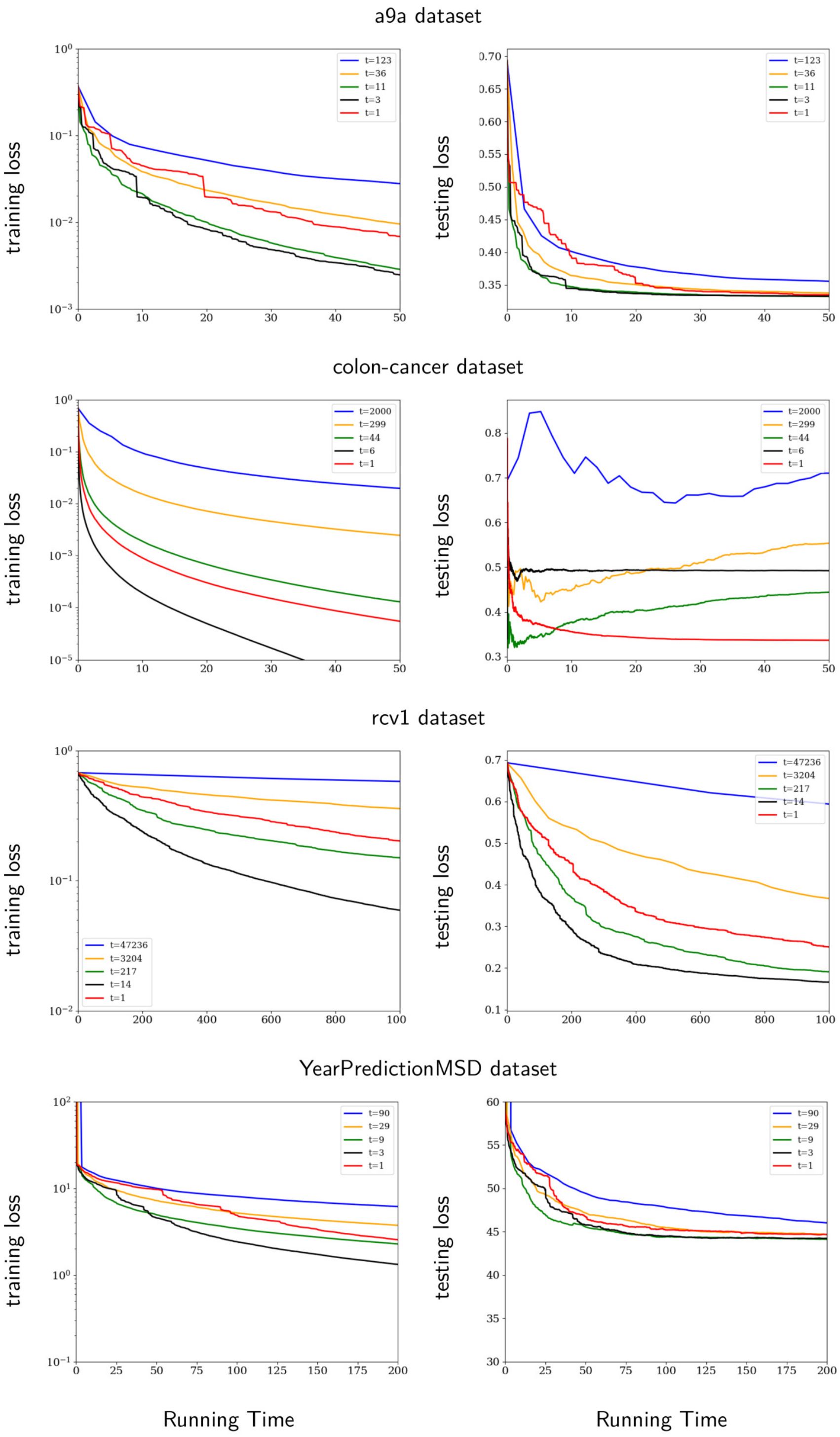


Fig. 5. *Plots showing the training optimality gap (in* log *scale) and testing loss versus running time for four different datasets. We consider RGBM for different t values (with the largest corresponding to GBM). The general observations are similar to that in Figure* 1*—we get significant computational savings by using a smaller value of t, without any loss in training/testing error.*

smallest training error lead to poor generalization. In many examples, we observe that a choice of t in the interior of its range of possible values, leads to a model with best test performance. For many examples (e.g., the third and fourth rows), we see that the testing error profiles saturate near the minimum even if the training error continues to decrease. The last two observations empirically suggest that the randomization scheme within RGBM potentially imparts additional regularization resulting in good generalization performance.

6. Discussion.

In this paper we present a greedy coordinate descent perspective of the popular GBM algorithm, where the coordinates correspond to weak learners and the collection of weak learners/coordinates can be exponentially large. We introduce and formally analyze RGBM, a randomized variant of popular GBM. RGBM can be thought as a Random-then-Greedy Coordinate Descent procedure where we randomly select a subset of weak learners and then choose the best weak learner from these candidates by a greedy mechanism. This presents a formal algorithmic justification of common heuristics used within the popular GBM implementations (e.g., XGBoost). From an optimization perspective, RGBM can be interpreted as a natural bridge between greedy coordinate descent on one end and randomized coordinate descent on the other. Our Random-then-Greedy Coordinate Descent procedure can be used as a stand-alone algorithm and can be potentially employed in machine learning contexts where coordinate descent is a popular choice [40]. On a related note, recent developments in large scale coordinate descent—such as the works of [36] and [37]—may be used to improve upon our proposed coordinate descent procedure (and in particular RGBM). We derive new computational guarantees for RGBM based on a coordinate descent interpretation. The guarantees depend upon a quantity that we call MCA (Minimum Cosine Angle) relating to the density of the weak learners or basis elements in the prediction space. The MCA quantity seems to bear some similarities with the Cheung–Cucker condition number [8, 33] used to analyze computational guarantees of solving feasibility problems in a linear system. A precise connection between these quantities does not seem to be straightforward, and a detailed investigation of their links is an interesting direction for future research.

The focus of our paper is on the algorithmic properties of RGBM—in terms of minimizing the empirical loss function as opposed to the population risk. Boosting is empirically known to lead to excellent out-of-sample properties by virtue of its implicit regularization properties [18, 39, 43] that are imparted by the algorithm. As our numerical experiments suggest, RGBM appears to have superior generalization properties by virtue of its random-then-greedy selection rule as opposed to a pure greedy method (as in GBM). A formal explanation of the generalization ability of RGBM is not addressed in this work and is an important topic of future research.

Acknowledgments. The authors would like to thank Robert Freund for helpful discussions and feedback. The authors would also like to thank the Associate Editor and reviewers for their helpful feedback that helped us improve the paper.

REFERENCES

[1] A. Beck and L. Tetruashvili, *On the convergence of block coordinate descent type methods*, SIAM J. Optim., 23 (2013), pp. 2037–2060.

[2] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[3] P. J. Bickel, Y. Ritov, and A. Zakai, *Some theory for generalized boosting algorithms*, J. Mach. Learn. Res., 7 (2006), pp. 705–732.

[4] M. Bogdan, E. Van Den Berg, C. Sabatti, W. Su, and E. J. Candès, *Slope—adaptive variable selection via convex optimization*, Ann. Appl. Stat., 9 (2015), pp. 1103–1140.

[5] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth, New York, 1984.

[6] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM Trans. Intell. Syst. Technol., 2 (2011), 27.

[7] T. Chen and C. Guestrin, *XGBoost: A scalable tree boosting system*, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, 2016, pp. 785–794.

[8] D. Cheung and F. Cucker, *A new condition number for linear programming*, Math. Program., 91 (2001), pp. 163–174.

[9] M. Collins, R. E. Schapire, and Y. Singer, *Logistic regression, AdaBoost and Bregman distances*, Mach. Learn., 48 (2002), pp. 253–285.

[10] M. Epelman and R. M. Freund, *Condition number complexity of an elementary algorithm for computing a reliable solution of a conic linear system*, Math. Program., 88 (2000), pp. 451–485.

[11] R. M. Freund, P. Grigas, and R. Mazumder, *A new perspective on boosting in linear regression via subgradient optimization and relatives*, Ann. Statist., 45 (2017), pp. 2328–2364.

[12] Y. Freund and R. E. Schapire, *Game theory, on-line prediction and boosting*, in Proceedings of the Ninth Annual Conference on Computational Learning Theory, 1996, pp. 325–332.

[13] Y. Freund and R. E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting*, J. Comput. System Sci., 55 (1997), pp. 119–139.

[14] J. Friedman, T. Hastie, and R. Tibshirani, *Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors)*, Ann. Statist., 28 (2000), pp. 337–407.

[15] J. H. Friedman, *Greedy function approximation: A gradient boosting machine*, Ann. Statist., 29 (2001), pp. 1189–1232.

[16] J. H. Friedman, *Stochastic gradient boosting*, Comput. Statist. Data Anal., 38 (2002), pp. 367–378.

[17] M. Gurbuzbalaban, A. Ozdaglar, P. A. Parrilo, and N. Vanli, *When cyclic coordinate descent outperforms randomized coordinate descent*, in Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 7002–7010.

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

[19] H. Hazimeh and R. Mazumder, *Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms*, Oper. Res., 68 (2020), pp. 1517–1537.

[20] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, *LightGBM: A highly efficient gradient boosting decision tree*, in Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 3146–3154.

[21] A. Liaw and M. Wiener, *Classification and regression by randomForest*, R News, 2 (2002), pp. 18–22.

[22] H. Lu, R. M. Freund, and V. Mirrokni, *Accelerating greedy coordinate descent methods*, In International Conference on Machine Learning, 2018, pp. 3257–3266.

[23] Z.-Q. Luo and P. Tseng, *On the convergence of the coordinate descent method for convex differentiable minimization*, J. Optim. Theory Appl., 72 (1992), pp. 7–35.

[24] Z.-Q. Luo and P. Tseng, *Error bounds and convergence analysis of feasible descent methods: A general approach*, Ann. Oper. Res., 46 (1993), pp. 157–178.

[25] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. B. Tsai, M. Amde, and S. Owen, *MLlib: Machine learning in Apache spark*, J. Mach. Learn. Res., 17 (2016), pp. 1235–1241.

[26] I. Mukherjee, C. Rudin, and R. E. Schapire, *The rate of convergence of AdaBoost*, J. Mach. Learn. Res., 14 (2013), pp. 2315–2347.

[27] S. Negahban and M. J. Wainwright, *Joint support recovery under high-dimensional scaling: Benefits and perils of $\ell_{1,\infty}$-regularization*, in Proceedings of the 21st International Conference on Neural Information Processing Systems, 2008, pp. 1161–1168.

[28] Y. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM J. Optim., 22 (2012), pp. 341–362.

[29] J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke, *Coordinate descent converges faster with the Gauss-Southwell rule than random selection*, in Proceedings of the 32nd International Conference on Machine Learning, 2015, pp. 1632–1641.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, *Scikit-learn: Machine learning in*

Python, J. Mach. Learn. Res., 12 (2011), pp. 2825–2830.

[31] N. Ponomareva, S. Radpour, G. Hendry, S. Haykal, T. Colthurst, P. Mitrichev, and A. Grushetsky, *TF boosted trees: A scalable TensorFlow based framework for gradient boosting*, in Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2017, pp. 423–427.

[32] A. Ramdas and J. Pena, *Margins, kernels and non-linear smoothed perceptrons*, in Proceedings of the 31st International Conference on Machine Learning, JMLR.org, 2014, pp. 244–252.

[33] A. Ramdas and J. Pena, *Towards a deeper geometric, analytic and algorithmic understanding of margins*, Optim. Methods Softw., 31 (2016), pp. 377–391.

[34] P. Richtarik and M. Takac, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Math. Program., 144 (2014), pp. 1–38.

[35] C. Scherrer, M. Halappanavar, A. Tewari, and D. Haglin, *Scaling up coordinate descent algorithms for large ℓ_1 regularization problems*, in Proceedings of the 29nd International Conference on International Conference on Machine Learning, 2012.

[36] C. Scherrer, A. Tewari, M. Halappanavar, and D. Haglin, *Feature clustering for accelerating parallel coordinate descent*, in Proceedings of the 25th International Conference on Neural Information Processing Systems, 2012, pp. 28–36.

[37] S. U. Stich, A. Raj, and M. Jaggi, *Approximate steepest coordinate descent*, in Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 3251–3259.

[38] R. Sun and Y. Ye, *Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version*, Math. Program., to appear, https://doi.org/10.1007/s10107-019-01437-5.

[39] M. Telgarsky, *A primal-dual convergence analysis of boosting*, J. Mach. Learn. Res., 13 (2012), pp. 561–606.

[40] S. J. Wright, *Coordinate descent algorithms*, Math. Program., 151 (2015), pp. 3–34.

[41] Y. You, X. Lian, J. Liu, H.-F. Yu, I. S. Dhillon, J. Demmel, and C.-J. Hsieh, *Asynchronous parallel greedy coordinate descent*, in Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 4682–4690.

[42] X. Zeng and M. A. T. Figueiredo, *Decreasing weighted sorted ℓ_1 regularization*, IEEE Signal Process. Lett., 21 (2014), pp. 1240–1244.

[43] T. Zhang and B. Yu, *Boosting with early stopping: Convergence and consistency*, Ann. Statist., 33 (2005), pp. 1538–1579.