Shrinkage Degree in L_2-Rescale Boosting for Regression

Lin Xu, Shaobo Lin, Yao Wang, and Zongben Xu

***Abstract*— L_2-rescale boosting (L_2-RBoosting) is a variant of L_2-Boosting, which can essentially improve the generalization performance of L_2-Boosting. The key feature of L_2-RBoosting lies in introducing a shrinkage degree to rescale the ensemble estimate in each iteration. Thus, the shrinkage degree determines the performance of L_2-RBoosting. The aim of this paper is to develop a concrete analysis concerning how to determine the shrinkage degree in L_2-RBoosting. We propose two feasible ways to select the shrinkage degree. The first one is to parameterize the shrinkage degree and the other one is to develop a data-driven approach. After rigorously analyzing the importance of the shrinkage degree in L_2-RBoosting, we compare the pros and cons of the proposed methods. We find that although these approaches can reach the same learning rates, the structure of the final estimator of the parameterized approach is better, which sometimes yields a better generalization capability when the number of sample is finite. With this, we recommend to parameterize the shrinkage degree of L_2-RBoosting. We also present an adaptive parameter-selection strategy for shrinkage degree and verify its feasibility through both theoretical analysis and numerical verification. The obtained results enhance the understanding of L_2-RBoosting and give guidance on how to use it for regression tasks.**

***Index Terms*— Boosting, generalization capability, L_2-rescale boosting (L_2-RBoosting), regression, shrinkage degree.**

I. Introduction

BOOSTING is a learning system, which combines many parsimonious models to produce a model with prominent predictive performance. The underlying intuition is that combines many rough rules of thumb can yield a good composite learner. From the statistical viewpoint, boosting can be viewed as a form of functional gradient decent [1]. It connects various boosting algorithms to optimization problems with specific loss functions. Typically, L_2-Boosting [2] can be interpreted as a stepwise additive learning scheme that concerns the problem of minimizing the L_2 risk. Boosting is resistant to overfitting [3] and, thus, has triggered enormous research activities in the past 20 years [1], [4]–[7].

Although the universal consistency of boosting has already been verified in [8], its numerical convergence rate is a bit slow [8], [9]. The main reason for is that the step size derived via linear search in boosting is usually not the most appropriate [10]. Under this circumstance, various variants of boosting, comprising the regularized boosting via shrinkage (RSBoosting) [11], regularized boosting via truncation (RTBoosting) [12], and ε-Boosting [13], have been developed via introducing additional parameters to control the step size. Both the experimental and theoretical results [1], [4], [11], [14] showed that these variants outperform boosting. However, it also needs verifying whether the learning performances of these variants can be further improved, say, to the best of our knowledge, there is not any related theoretical analysis to illustrate the optimality of these variants, at least for a certain aspect, such as the generalization capability, numerical (or population) convergence rate, and so on.

Motivated by the development of the relaxed greedy algorithm [15] and sequential greedy algorithm [16], Lin *et al.* [17] introduced a new variant of boosting named the rescale boosting (RBoosting). Different from the existing variants that concentrate on controlling the step size, RBoosting builds upon rescaling the ensemble estimator and implementing the linear search without any restrictions on the step size in each iteration. Under such setting, almost optimal (up to a logarithmic factor) numerical convergence rates of RBoosting with convex loss functions were derived in [17]. The first purpose of this paper is to prove that, when restricted to L_2-rescale boosting (L_2-RBoosting), the logarithmic term can be omitted. In short, the optimal numerical convergence rate can be derived.

As there is no free lunch, all the variants improve the learning performance of boosting at the cost of introducing an additional parameter, such as the truncated parameter in RTBoosting, regularization parameter in RSBoosting, ε in ε-Boosting, and shrinkage degree in RBoosting. To facilitate the use of these variants, one should also present the strategies to select such parameters. In particular, Elith *et al.* [18] showed that 0.1 is a feasible choice of ε in ε-Boosting; Bühlmann and Hothorn [4] recommended the selection of 0.1 for the regularization parameter in RSBoosting; Zhang and Yu [12] proved that $\mathcal{O}(k^{-2/3})$ is a good value of the truncated parameter in RTBoosting, where k is the number of iterations. Thus, it is interesting and important to provide a feasible strategy for selecting shrinkage degree in RBoosting.

The second and main purpose of this paper is to propose several feasible strategies to determine the shrinkage degree

Manuscript received January 8, 2016; accepted April 19, 2016. Date of publication May 10, 2016; date of current version July 18, 2017. This work was supported in part by the Key Project of National Science Foundation of China under Grant 11131006 and Grant 91330204 and in part by the National Basic Research Program of China (973 Program) under Grant 2013CB329404.

L. Xu, Y. Wang, and Z. Xu are with the Institute for Information and System Sciences, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: xulinshadow@gmail.com; yao.s.wang@gmail.com; zbxu@mail.xjtu.edu.cn).

S. Lin is with the College of Mathematics and Information Science, Wenzhou University, Wenzhou 325035, China (e-mail: sblin1983@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2016.2560224

in L_2-RBoosting for regression and analyze their pros and cons. For this purpose, we need to show the essential role of the shrinkage degree in L_2-RBoosting. Based on the rigorous theoretical analysis, we find that, different from other parameters, such as the truncated value, regularization parameter, and ε value, the shrinkage degree does not affect the learning rate, in the sense that, for arbitrary finite shrinkage degree, the learning rate of corresponding L_2-RBoosting can reach the existing best record of all boosting-type algorithms. It means that if the number of samples is infinite, the shrinkage degree does not affect the generalization capability of L_2-RBoosting. However, our result also shows that the essential role of the shrinkage degree in L_2-RBoosting lies in its important impact on the constant of the generalization error, which is crucial when there are only finite number of samples. In such a sense, we theoretically prove that there exists an optimal shrinkage degree to minimize the generalization error of L_2-RBoosting.

We then develop two effective methods for selecting an appropriate shrinkage degree. The first one is to consider the shrinkage degree as a parameter in the learning process of L_2-RBoosting. The other one is to learn the shrinkage degree from the samples directly and we call it as the L_2-data-driven RBoosting (L_2-DDRBoosting). We find that the two approaches can reach the same learning rate and the number of parameters in L_2-DDRBoosting is less than those in L_2-RBoosting. However, we also prove that the estimator deduced from L_2-RBoosting possesses a better structure (smaller $\mathcal{L}_1$ norm), which sometimes leads to a much better generalization capability for some special weak learners. Thus, we recommend the use of L_2-RBoosting in practice. Finally, we present an adaptive shrinkage degree selection strategy for L_2-RBoosting. Both the theoretical and experimental results verify the feasibility and outperformance of L_2-RBoosting.

The rest of this paper is organized as follows. In Section II, we introduce the L_2-Boosting, L_2-RBoosting, and L_2-DDRBoosting. In Section III, we study the related theoretical behaviors of L_2-RBoosting. In Section IV, a series of simulations and real data experiments are employed to illustrate our theoretical assertions. Finally, the conclusion is drawn in Section V.

II. L_2-Boosting, L_2-RBoosting, and L_2-DDRBoosting

Ensemble techniques, such as bagging [19], boosting [6], stacking [20], Bayesian averaging [21], and random forest [22], can significantly improve the performance in practice and benefit from favorable learning capability. In particular, boosting and its variants are based on a rich theoretical analysis, to just name a few [2], [3], [8], [12], [17], [23], [24], [25]. The aim of this section is to introduce some concrete boosting-type learning schemes for regression.

In a regression problem with a covariate X on $\mathcal{X} \subseteq \mathbf{R}^d$ and a real response variable $Y \in \mathcal{Y} \subseteq \mathbf{R}$, we observe m independent identically distributed. samples $D_m = \{(x_i, y_i)\}_{i=1}^m$ from an unknown distribution ρ. Without loss of generality, we always assume $\mathcal{Y} \subseteq [-M, M]$, where $M < \infty$ is a positive real number. The aim is to find a function to minimize the generalization error

$$\mathcal{E}(f) = \int \phi(f(x), y) d\rho$$

where $\phi : \mathbf{R} \times \mathbf{R} \to \mathbf{R}_+$ is called a loss function [12]. If $\phi(f(x), y) = (f(x) - y)^2$, then the known regression function

$$f_\rho(x) = \mathbf{E}\{Y|X = x\}$$

minimizes the generalization error. In such a setting, one is interested in finding a function f_D based on D_m, such that $\mathcal{E}(f_D) - \mathcal{E}(f_\rho)$ is small. Buhlmann and Yu [2] showed that L_2-Boosting can successfully tackle this problem.

Algorithm 1 L_2-Boosting

```
**Step** 1 (**Initialization**):
Given data$\{(x_i, y_i) : i = 1, \ldots, m\}$ and dictionary $S$.
Given initial estimator $f_0 \in \mathrm{span}(S)$.
Set the maximum number of iterations $K$, iteration $k := 0$.
**Step** 2 (**Projection of gradient**):
Find $g_k^* \in S$ such that
    $g_k^* = \arg\max_{g \in S} |\langle r_{k-1}, g\rangle_m|,$
where residual $r_{k-1} = y - f_{k-1}$ and $y$ is a function satisfying $y(x_i) = y_i$.
**Step** 3 (**Linear search**):
Generate the $k$th estimator as
    $f_k = f_{k-1} + \langle r_{k-1}, g_k^*\rangle_m g_k^*.$
**Step** 4 (**Iteration process**):
Increase $k$ by one and repeat steps 2 and 3 if $k < K$.
```

Let $S = \{g_1, \ldots, g_n\}$ be the set of weak learners (regressors) and define

$$\mathrm{span}(S) = \left\{ \sum_{j=1}^{n} a_j g_j : g_j \in S, a_j \in \mathbf{R}, n \in \mathbf{N} \right\}.$$

Let

$$\|f\|_m = \sqrt{\frac{1}{m} \sum_{i=1}^{m} f(x_i)^2} \text{ and } \langle f, g\rangle_m = \frac{1}{m} \sum_{i=1}^{m} f(x_i) g(x_i)$$

be the empirical norm and empirical inner product, respectively. Furthermore, we define the empirical risk as

$$\mathcal{E}_D(f) = \frac{1}{m} \sum_{i=1}^{m} |f(x_i) - y_i|^2.$$

Then, the gradient descent view of L_2-Boosting [1] can be interpreted as follows.

Remark 1: In step 3 of Algorithm 1, it is easy to check that

$$\langle r_{k-1}, g_k^*\rangle_m = \arg\min_{\beta_k \in \mathbf{R}} \mathcal{E}_D\left(f_{k-1} + \beta_k g_k^*\right).$$

Therefore, we call it as the linear search step.

Remark 2: In the kth iteration, searching through the set S (projection of gradient) has a complexity of $\mathcal{O}(mn)$. After selecting a new weak learner g_k^*, 1-D linear search for the step size β_k needs a complexity of $\mathcal{O}(m)$. Thus, the kth iteration of the naive boosting implementation has complexity $\mathcal{O}(mn+m)$, which shows that the projection of gradient dominates complexity. If the total iteration number is set as K, then the overall complexity of boosting is $\mathcal{O}(Kmn)$ and the memory required for the naive approach is $\mathcal{O}(mn)$.

Although L_2-Boosting was proved to be consistent [8] and overfitting resistance [2], multiple studies [9], [26], [27] showed that its numerical convergence rate is far slower than the best nonlinear approximant. The main reason is that the linear search in Algorithm 1 makes f_{k+1} to be not always the greediest one [10], [17]. Hence, an advisable method is to control the step size in the linear search step of Algorithm 1. Thus, various variants of boosting, such as the ε-Boosting [13], which specifies the step size as a fixed small positive number ε rather than using the linear search, RSBoosting [11], which multiplies a small regularized factor to the step size deduced from the linear search and RTBoosting [12], which truncates the linear search in a small interval, have been developed. It is obvious that the core difficulty of these schemes roots in how to select an appropriate step size. If the step size is too large, then these algorithms may face the same problem as that of Algorithm 1. If the step size is too small, then the numerical convergence rate is also fairly slow.

Other than the aforementioned strategies that focus on controlling the step size of g_k^*, Lin *et al.* [17] also derived a new backward type strategy, called the RBoosting, to improve the numerical convergence rate and, consequently, the generalization capability of boosting. The core idea is that if the approximation (or learning) effect of the kth iteration does not work as expected, then f_k is regarded to be too aggressive. That is, if a new iteration is employed, then the previous estimator f_k should be rescaled. Suppose L_2 loss is considered for regression tasks, the main idea of L_2-RBoosting is depicted as in Algorithm 2.

Remark 3: It is easy to see that

$$\langle r_{k-1}^s, g_k^* \rangle_m = \arg\min_{\beta_k \in \mathbf{R}} \mathcal{E}_D\big((1-\alpha_k) f_{k-1} + \beta_k g_k^*\big).$$

This is the only difference between L_2-Boosting and L_2-RBoosting. Here, we call α_k as the shrinkage degree. It can be found in Algorithm 2 that the shrinkage degree is considered as a parameter.

Remark 4: Suppose we have L candidate values of the parameter, then the overall complexity of parameterized L_2-RBoosting is about $\mathcal{O}(LKmn)$. The memory required is also $\mathcal{O}(mn)$ for the dictionary and inner products. Note that, as there is no free lunch, parameterized L_2-RBoosting improves the learning performance of boosting by imposing additional computational burden, especially when the candidate value L is large.

Algorithm 2 L_2-RBoosting

```
Step 1 (Initialization):
  Given data $\{(x_i, y_i) : i = 1, \ldots, m\}$ and dictionary $S$. Given a set of $\{\alpha_k\}_{k=1}^{k^*}$ with $\alpha_k = 2/(u+k)$ and $u \in \mathbf{N}$ being the shrinkage degree.
  Given initial estimator $f_0 \in \mathrm{span}(S)$.
  Set the maximum number of iterations $K$, iteration $k := 0$.
Step 2 (Projection of gradient):
  Find $g_k^* \in S$ such that
    $g_k^* = \arg\max_{g \in S} |\langle r_{k-1}, g \rangle_m|,$
  where the residual $r_{k-1} = y - f_{k-1}$ and $y$ is a function satisfying $y(x_i) = y_i$.
Step 3 (Re-scaled linear search):
  Generate the $k$th estimator as
    $f_k = (1-\alpha_k) f_{k-1} + \langle r_{k-1}^s, \quad g_k^* \rangle_m g_k^*,$
  where the shrinkage residual $r_{k-1}^s = y - (1-\alpha_k) f_{k-1}$.
Step 4 (Iteration process):
  Increase $k$ by one and repeat Step 2 and Step 3 if $k < K$.
```

L_2-RBoosting stems from the greedy algorithm with fixed relaxation [27] in nonlinear approximation. It is different from the L_2-Boosting algorithm proposed in [23], which adopts the idea of X-greedy algorithm with relaxation [28]. In particular, we employ r_{k-1} in step 2 to represent residual rather than the shrinkage residual r_{k-1}^s in step 3. Such a difference makes the design principles of L_2-RBoosting and the L_2-Boosting algorithm in [23] to be totally distinct. In L_2-RBoosting, the algorithm comprises two steps: the projection of gradient step to find the optimum weak learner g_k^* and the rescale linear search step to fix its step size β_k. However, the L_2-Boosting algorithm in [23] only concerns the optimization problem

$$\arg\min_{g_k^* \in S, \beta_k \in \mathbf{R}} \left\| (1-\alpha_k) f_{k-1} + \beta_k g_k^* \right\|_m^2.$$

The main drawback is, to the best of our knowledge, the closed-form solution of the optimization problem only holds for the L_2 loss. When faced with other loss, the L_2-Boosting algorithm in [23] cannot be efficiently numerical solved due to it needs to tune two parameters simultaneously in an optimization problem.

Previous works [16], [17], [23], [28], [29] have shown that introducing a parameter to rescale the previous estimator can improve the numerical convergence rate and generalization capability of boosting. However, the difficulty is how to tune such an additional parameter, the shrinkage degree α_k, just like the step-size parameter ε in ε-Boosting [13], regularization parameter in RSBoosting [11], and truncation parameter in RTBoosting [12]. Therefore, it is urgent to develop a feasible method to select the shrinkage degree. There are two ways to choose a good value of shrinkage degree in L_2-RBoosting. The first one is to parameterize the shrinkage degree as in Algorithm 2. We set the shrinkage degree $\alpha_k = 2/(k+u)$ and hope to choose an appropriate value of u via a certain parameter-selection strategy. The other one is to learn the shrinkage degree α_k from the samples directly. As we are only concerned with L_2 loss for regression in this paper, this idea can be primitively realized by Algorithm 3, which is called the L_2-DDRBoosting.

Remark 5: The projection of gradient step in Algorithm 3 also has a complexity of $\mathcal{O}(mn)$ in the kth iteration. For 2-D

Algorithm 3 L_2-DDRBoosting

```
Step 1 (Initialization):
  Given data $\{(x_i, y_i) : i = 1, \dots, m\}$ and dictionary $S$.
  Given initial estimator $f_0 \in \mathrm{span}(S)$.
  Set the maximum number of iterations $K$, iteration $k := 0$.
Step 2 (Projection of gradient):
  Find $g_k^* \in S$ such that
      $g_k^* = \arg\max_{g \in S} |\langle r_{k-1}, g \rangle_m|$,
  where residual $r_{k-1} = y - f_{k-1}$ and $y$ is a function satisfying $y(x_i) = y_i$.
Step 3 (Two dimensional linear search):
  Find $\alpha_k^*$ and $\beta_k^* \in \mathbf{R}$ such that
      $(\alpha_k^*, \beta_k^*) = \arg \min_{(\alpha_k, \beta_k) \in \mathbf{R}^2} \mathcal{E}_D((1 - \alpha_k) f_{k-1} + \beta_k g_k^*)$
  Update $f_k = (1 - \alpha_k^*) f_{k-1} + \beta_k^* g_k^*$.
Step 4 (Iteration process):
  Increase $k$ by one and repeat Step 2 and Step 3 if $k < K$.
```

linear search step, assuming the cost of inverting a complex $d \times d$ matrix is at least $\mathcal{O}(d^3)$, then d-dimensional linear search for the step size needs a complexity of $\mathcal{O}(md^3)$. Thus, 2-D ($d = 2$) linear search in L_2-DDRBoosting for the step size β_k needs a complexity of $\mathcal{O}(m)$, when $m \gg 8$. Thus, the overall complexity and the memory required of L_2-DDRBoosting are still $\mathcal{O}(Kmn)$ and $\mathcal{O}(mn)$, respectively. Note that, L_2-DDRBoosting can perfectly solve the parameter-selection problem in the rescale-type boosting algorithm with almost negligible additional computational complexity.

Remark 6: Algorithm 3 is motivated by the greedy algorithm with free relaxation [29]. As far as the L_2 loss is concerned, it is easy to deduce the close-form representation of f_k [27]. However, for other loss functions, we have not found any papers concerning the analytical solvability of the optimization problem in step 3 of Algorithm 3.

III. Theoretical Behaviors

In this section, we present some theoretical results concerning the shrinkage degree. First, we derive optimal numerical convergence rates of L_2-RBoosting. Second, we study the relationship between the shrinkage degree and the generalization capability in L_2-RBoosting. The theoretical result reveals that the shrinkage degree plays a crucial role in L_2-RBoosting for regression with finite samples. Third, we analyze the pros and cons of L_2-RBoosting and L_2-DDRBoosting. It is shown that the potential performance of L_2-RBoosting is somewhat better than that of L_2-DDRBoosting. Finally, we propose an adaptive parameter-selection strategy for the shrinkage degree and theoretically verify its feasibility.

A. Optimal Numerical Convergence Rates

Let $S = \{g_1, \dots, g_n\}$ be the set of weak learners. Denote by $\mathcal{L}_1(S) := \{f : f = \sum_{g \in S} a_g g\}$, the space of l_1-summable functions with respect to S endowed with the norm

$$\|f\|_{\mathcal{L}_1(S)} := \inf \left\{ \sum_{g \in S} |a_g| : f = \sum_{g \in S} a_g g \right\}.$$

We assume $\sup_{x \in \mathcal{X}} |g(x)| \leq 1$ for all $g \in S$.

The numerical convergence rate of a boosting-type algorithm describes the relation between the approximation accuracy and the number of boosting iterations. It determines not only the efficiency of the algorithm in implementation, but also the generalization capability [12]. Thus, the boosting-type algorithms with optimal numerical convergence rate verification are preferable. Theorem 7, which will be proved in Section V, describes the numerical convergence rates of L_2-RBoosting.

Theorem 7: Let f_k be the estimator defined in Algorithm 2. Then, for arbitrary $h \in \mathrm{span}(S)$ and $u \in \mathbf{N}$, there holds

$$\|f_k - y\|_m^2 \leq 2\|y - h\|_m^2 + 2(M + \|h\|_{\mathcal{L}_1(S)})^2 2^{\frac{3u^2+14u+20}{8u+8}} k^{-1}.$$

Since $\|y - h\|_m$ only depicts the richness of S, the numerical convergence rates derived in Theorem 7 is $\mathcal{O}(k^{-1})$. As shown in [26], the deduced convergence rate is optimal in the sense that there are an $h^* \in \mathrm{span}(S)$ with bounded $\|h^*\|_{\mathcal{L}_1}$ and a constant C independent of k such that

$$\left| \|f_k - y\|_m^2 - \|y - h^*\|_m^2 \right| \geq C k^{-1}.$$

The assertion implies that L_2-RBoosting is one of the most efficient boosting-type algorithms in the worst case analysis. The derived numerical convergence rate is much better than that of RTBoosting [12], which behaves asymptomatically as $\mathcal{O}(k^{-1/3})$. Furthermore, our estimate is also better than the numerical convergence rate derived in [17] for L_2-RBoosting, since there is an additional logarithmic factor in the estimate.

According to the greedy algorithm with free relaxation in [29], the numerical convergence rate of L_2-DDRBoosting is still $\mathcal{O}(k^{-1})$.

B. Relationship Between the Generalization Capability and the Shrinkage Degree

To describe the generalization capability of L_2-Rboosting, we should present the classes of regression functions. The space $\mathcal{L}_1^r$ is defined to be the set of all the functions f, such that, there exists a $h \in \mathrm{span}\{S\}$, such that

$$\|h\|_{\mathcal{L}_1(S)} \leq \mathcal{B}, \quad \text{and} \quad \|f - h\| \leq \mathcal{B} n^{-r}. \qquad \text{(III.1)}$$

The infimum of all such $\mathcal{B}$ defines the norm for f on $\mathcal{L}_1^r$. It follows from [28] that (III.1) defines an interpolation space, which has been widely used in nonlinear approximation [25], [27], [28].

Let $\pi_M t$ denote the clipped value of t at $\pm M$, that is, $\pi_M t := \min\{M, |t|\}\mathrm{sgn}(t)$. Then, it is obvious that [30] for all $t \in \mathbf{R}$ and $y \in [-M, M]$, there holds

$$\mathcal{E}(\pi_M f_k) - \mathcal{E}(f_\rho) \leq \mathcal{E}(f_k) - \mathcal{E}(f_\rho).$$

By the help of the descriptions, we are in a position to present Theorem 8, which depicts the role that the shrinkage degree plays in L_2-RBoosting.

Theorem 8: Let $0 < t < 1$, and f_k be the estimator defined in Algorithm 2. If $f_\rho \in \mathcal{L}_1^r$, then for arbitrary $k, u \in \mathbf{N}$

$$\mathcal{E}(\pi_M f_k) - \mathcal{E}(f_\rho) \leq C(M+\mathcal{B})^2 \left(2^{\frac{3u^2+14u+20}{8u+8}} k^{-1} + (m/k)^{-1} \log m \log \frac{2}{t} + n^{-2r}\right)$$

holds with probability at least $1-t$, where C is a positive constant depending only on d.

Let us first give some remarks of Theorem 8. If we set the number of iterations and the size of dictionary to satisfy $k = \mathcal{O}(m^{1/2})$, and $n \geq \mathcal{O}(m^{(1/4r)})$, then we can deduce a learning rate of $\pi_M f_k$ asymptotically as $\mathcal{O}(m^{-1/2} \log m)$. This rate is independent of the dimension and is the same as the optimal record for the greedy learning [28] and boosting-type algorithms [12]. Furthermore, under the same assumptions, this rate is faster than those of boosting [8] and RTBoosting [12]. Thus, we can draw a rough conclusion that the learning rate deduced in Theorem 8 is tight. Under this circumstance, we think that it can reveal the essential performance of L_2-RBoosting.

Then, it can be found in Theorem 8 that if u is finite and the number of samples is infinite, the shrinkage degree u does not affect the learning rate of L_2-RBoosting, which means that its generalization capability is independent of u. However, it is known that in the real-world application, there are only finite number of samples available. Thus, u plays a crucial role in the learning process of L_2-RBoosting in practice. Our results in Theorem 8 implies two simple guidance to deepen the understanding of L_2-RBoosting. The first one is that there does exist an optimal u (may be not unique) minimizing the generalization error of L_2-RBoosting. In particular, we can deduce a concrete value of optimal u via minimizing $(3u^2 + 14u + 20/8u + 8)$. As it is very difficult to prove the optimality of the constant, we think that it is more reasonable to reveal a rough trend for choosing u rather than providing a concrete value. The other one is that when $u \to \infty$, L_2-RBoosting behaves as L_2-Boosting, and the learning rate cannot achieve $\mathcal{O}(m^{-1/2} \log m)$. Thus, we indeed present a theoretical verification that L_2-RBoosting outperforms L_2-Boosting.

C. Pros and Cons of L_2-RBoosting and L_2-DDRBoosting

There is only one parameter, k^*, in L_2-DDRBoosting, as shown in Algorithm 3. This implies that L_2-DDRBoosting improves the performance of L_2-Boosting without tuning another additional parameter α_k, which is superior to the other variants of boosting. Theorem 9 shows that, as the same as L_2-RBoosting, L_2-DDRBoosting can also improve the generalization capability of L_2-Boosting.

Theorem 9: Let $0 < t < 1$, and f_k' be the estimator defined in Algorithm 3. If $f_\rho \in \mathcal{L}_1^r$, then for any arbitrary $k \in \mathbf{N}$

$$\mathcal{E}\big(\pi_M f_k'\big) - \mathcal{E}(f_\rho) \leq C(M+\mathcal{B})^2 \left(k^{-1} + (m/k)^{-1} \log m \log \frac{2}{t} + n^{-2r}\right)$$

holds with probability at least $1-t$, where C is a constant depending only on d.

By Theorem 9, it seems that L_2-DDRBoosting can perfectly solve the parameter-selection problem in the rescale-type boosting algorithm. However, we also show that compared with L_2-DDRBoosting, L_2-RBoosting possesses an important advantage, which is crucial to guaranteeing the outperformance of L_2-RBoosting. In fact, noting that L_2-DDRBoosting depends on a 2-D linear search problem (step 3 in Algorithm 3), the structure of the estimator ($\mathcal{L}_1$ norm) cannot always be good. If the estimate f_{k-1}' and g_k^* are almost linear dependent, then the values of α_k and β_k may be very large, which automatically leads a huge $\mathcal{L}_1$ norm of f_k'. We show in Proposition 10 that L_2-RBoosting can avoid this phenomenon.

Proposition 10: If the f_k is the estimate defined in Algorithm 2, then there holds

$$\|f_k\|_{\mathcal{L}_1(S)} \leq C((M + \|h\|_{\mathcal{L}_1(S)})k^{1/2} + kn^{-r}).$$

Proposition 10 implies that the estimator defined in Algorithm 2 possesses a controllable structure. This may significantly improve the learning performance of L_2-RBoosting when faced with some specified weak learners. For this purpose, we need to introduce some definitions and conditions to qualify the weak learners.

Definition 11: Let $(\mathcal{M}, d)$ be a pseudometric space and $T \subset \mathcal{M}$ a subset. For every $\varepsilon > 0$, the covering number $\mathcal{N}(T, \varepsilon, d)$ of T with respect to ε and d is defined as the minimal number of balls of radius ε whose union covers T, that is

$$\mathcal{N}(T, \varepsilon, d) := \min \left\{ l \in \mathbf{N} : T \subset \bigcup_{j=1}^{l} B(t_j, \varepsilon) \right\}$$

for some $\{t_j\}_{j=1}^l \subset \mathcal{M}$, where $B(t_j, \varepsilon) = \{t \in \mathcal{M} : d(t, t_j) \leq \varepsilon\}$.

The l_2-empirical covering number of a function set is defined by means of the normalized l_2-metric d_2 on the Euclidean space $\mathbf{R}^d$ given in [31] with $d_2(\mathbf{a}, \mathbf{b}) = \big(1/m \sum_{i=1}^m |a_i - b_i|^2\big)^{1/2}$ for $\mathbf{a} = (a_i)_{i=1}^m, \mathbf{b} = (b_i)_{i=1}^m \in \mathbf{R}^m$.

Definition 12: Let $\mathcal{F}$ be a set of functions on X, $\mathbf{x} = (x_i)_{i=1}^m \subset X^m$, and let

$$\mathcal{F}|_{\mathbf{x}} := \big\{(f(x_i))_{i=1}^m : f \in \mathcal{F}\big\} \subset R^m.$$

Set $\mathcal{N}_{2,\mathbf{x}}(\mathcal{F}, \varepsilon) = \mathcal{N}(\mathcal{F}|_{\mathbf{x}}, \varepsilon, d_2)$. The l_2-empirical covering number of $\mathcal{F}$ is defined by

$$\mathcal{N}_2(\mathcal{F}, \varepsilon) := \sup_{m \in \mathbf{N}} \sup_{\mathbf{x} \in S^m} \mathcal{N}_{2,\mathbf{x}}(\mathcal{F}, \varepsilon), \quad \varepsilon > 0.$$

Before presenting the main result in this section, we shall introduce Assumption 13.

Assumption 13: The l_2-empirical covering number of $\mathrm{span}(S)$ satisfies

$$\log \mathcal{N}_2(B_1, \varepsilon) \leq \mathcal{L}\varepsilon^{-\mu} \quad \forall \varepsilon > 0$$

where

$$B_R = \{f \in \mathrm{span}(S) : \|f\|_{\mathcal{L}^1(S)} \leq R\}.$$

Such an assumption is widely used in a statistical learning theory. For example, Shi *et al.* [31] proved that the linear spanning of some smooth kernel functions satisfies Assumption 13 with a small value of μ. By the help of Assumption 13, we can prove that the learning performance of L_2-RBoosting can be essentially improved due to the good structure of the corresponding estimator.

Theorem 14: Let $0 < t < 1$, $\mu \in (0, 1)$ and f_k be the estimator defined in Algorithm 2. If $f_\rho \in \mathcal{L}_1^r$ and Assumption 13 holds, then we have

$$\mathcal{E}(\pi_M f_k) - \mathcal{E}(f_\rho)$$
$$\leq C \log \frac{2}{t}(3M + \mathcal{B})^2 \left(n^{-2r} + k^{-1} + \left(\frac{(kn^{-r} + \sqrt{k})^\mu}{m} \right)^{\frac{2-\mu}{2+\mu}} \right).$$

It can be found in Theorem 14 that if $\mu \to 0$, then the learning rate of L_2-RBoosting can be near to m^{-1}. This depicts that, with good weak learners, L_2-RBoosting can reach a fairly fast learning rate.

D. Adaptive Parameter-Selection Strategy for L_2-RBoosting

In Section III-C, we point out that L_2-RBoosting is potentially better than L_2-DDRBoosting. Consequently, how to select the parameter, u, is of great importance in L_2-RBoosting. We present an adaptive way to determine the shrinkage degree in this section and show that, the estimator based on such a parameter-selection strategy does not degrade the generalization capability very much. To this end, we split the samples $D_m = (x_i, y_i)_{i=1}^m$ into two parts of size $[m/2]$ and $m - [m/2]$, respectively (assuming $m \geq 2$). The first half is denoted by D_m^l (the learning set), which is used to construct the L_2-RBoosting estimate $f_{D_m^l, \alpha_k, k}$. The second half, denoted by D_m^v (the validation set), is used to choose α_k by picking $\alpha_k \in I := [0, 1]$ to minimize the empirical risk

$$\frac{1}{m - [m/2]} \sum_{i=[m/2]+1}^{m} \left(y_i - f_{D_m^l, \alpha_k^*, k}\right)^2.$$

Then, we obtain the estimator

$$f^*_{D_m^l, \alpha_k, k} = f_{D_m^l, \alpha_k^*, k}.$$

Since $y \in [-M, M]$, a straightforward adaptation of [32, Th. 7.1] yields that, for any $\delta > 0$

$$\mathbf{E}\left[\| f^*_{D_m^l, \alpha_k, k} - f_\rho \|_\rho^2\right] \leq (1 + \delta) \inf_{\alpha_k \in I} \mathbf{E}\left[\| f_{D_m^l, \alpha_k^*, k} - f_\rho \|_\rho^2\right] + C \frac{\log m}{m}$$

holds some positive constant C depending only on M, d, and δ. Immediately, from Theorem 8, we can conclude the following.

Theorem 15: Let $f^*_{D_m^l, \alpha_k, k}$ be the adaptive L_2-RBoosting estimator. If $f_\rho \in \mathcal{L}_1^r$, then for arbitrary constants $k, u \in \mathbf{N}$

$$\mathbf{E}\left\{\mathcal{E}\left(\pi_M f^*_{D_m^l, \alpha_k, k}\right) - \mathcal{E}(f_\rho)\right\}$$
$$\leq C(M + \mathcal{B})^2 \left(2^{\frac{3u^2+14u+20}{8u+8}} k^{-1} + (m/k)^{-1} \log m + n^{-2r} \right)$$

where C is an absolute positive constant.

Remark 16: Theorem 15 also implies that the holdout method (or twofold cross validation) is used to tune the parameters of an adaptive estimator is feasible in L_2-RBoosting.

IV. Numerical Results

In this section, a series of simulations and real data experiments will be carried out to illustrate our theoretical assertions.

A. Simulation Experiments

In this section, we first introduce the simulation settings, including the data sets, weak learners, and experimental environment. Second, we analyze the relationship between shrinkage degree and generalization capability for the proposed L_2-RBoosting by means of ideal performance curve. Third, we draw a performance comparison of L_2-Boosting, L_2-RBoosting, and L_2-DDRBoosting. The results illustrate that L_2-RBoosting with an appropriate shrinkage degree outperforms others, especially for high-dimensional data simulations. Finally, we justify the feasibility of the adaptive parameter-selection strategy for shrinkage degree in L_2-RBoosting.

1) Simulation Settings: In the simulations, we generate the data from the following model:

$$Y = m(X) + \sigma \cdot \varepsilon \tag{IV.1}$$

where ε is the standard Gaussian noise and independent of X. The noise level σ varies among in $\{0, 0.5, 1\}$, and X is uniformly distributed on $[-2, 2]^d$ with $d \in \{1, 2, 10\}$. Nine typical regression functions are considered in this set of simulations, where these functions are the same as those in [23, Sec. IV].

1) $m_1(x) = 2 * \max(1, \min(3 + 2 * x, 3 - 8 * x))$.
2) $m_2(x) = \begin{cases} 10\sqrt{-x}\sin(8\pi x) & -0.25 \leq x < 0 \\ 0 & \text{else.} \end{cases}$
3) $m_3(x) = 3 * \sin(\pi * x/2)$.
4) $m_4(x_1, x_2) = x_1 * \sin(x_1^2) - x_2 * \sin(x_2^2)$.
5) $m_5(x_1, x_2) = 4/(1 + 4 * x_1^2 + 4 * x_2^2)$.
6) $m_6(x_1, x_2) = 6 - 2 * \min(3, 4 * x_1^2 + 4 * |x_2|)$.
7) $m_7(x_1, \ldots, x_{10}) = \sum_{j=1}^{10} (-1)^{j-1} x_j \sin(x_j{}^2)$.
8) $m_8(x_1, \ldots, x_{10}) = m_6(x_1 + \cdots + x_5, x_6 + \cdots + x_{10})$.
9) $m_9(x) = \begin{cases} 1, & x_1 + \cdots + x_{10} \leq 0 \\ 3, & \text{else.} \end{cases}$

For each regression function and each value of $\sigma \in \{0, 0.5, 1\}$, we first generate a training set of size $m = 500$ and an independent test set, including $m' = 1000$ noiseless observations. We then evaluate the generalization capability of each boosting algorithm in terms of root mean squared error (RMSE).

It is known that the boosting tree algorithm requires the specification of two parameters. One is the number of splits (or the number of nodes) that is used for fitting each regression tree. The number of leaves equals the number of splits plus one. Specifying J splits corresponds to an estimate with up to J-way interactions. Friedman and Hastie [33] suggested that $4 \leq J \leq 8$ generally works well and the estimate is typically not sensitive to the exact choice of J within that range. Thus, in the following simulations, we use the CART [34] (with the number of splits $J = 4$) to build up the week learners for regression. The other parameter is the number of iterations or the number of trees to be fitted. A suitable value of iterations can range from a few dozen to several thousand, depending on

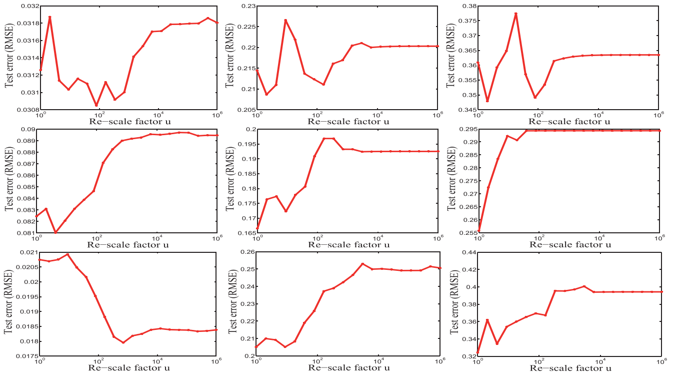


Fig. 1. Test error (RMSE) curve with respect to the rescale factor u in L_2-RBoosting. Three rows denote the 1-D regression functions m_1, m_2, and m_3 and three columns indicate that the noise level σ varies among in $\{0, 0.5, 1\}$, respectively.

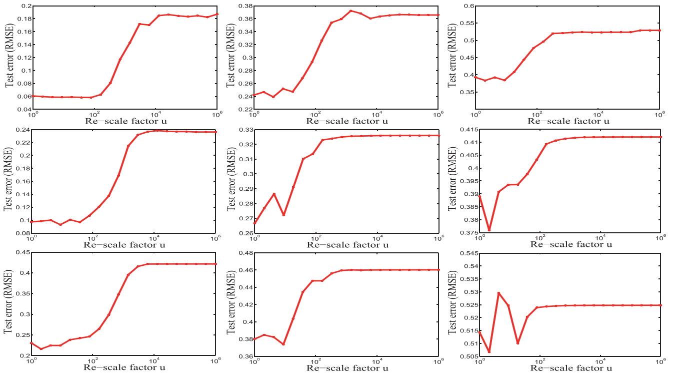


Fig. 2. Three rows denote the 2-D regression functions m_4, m_5, and m_6 and three columns indicate that the noise level σ varies among in $\{0, 0.5, 1\}$, respectively.

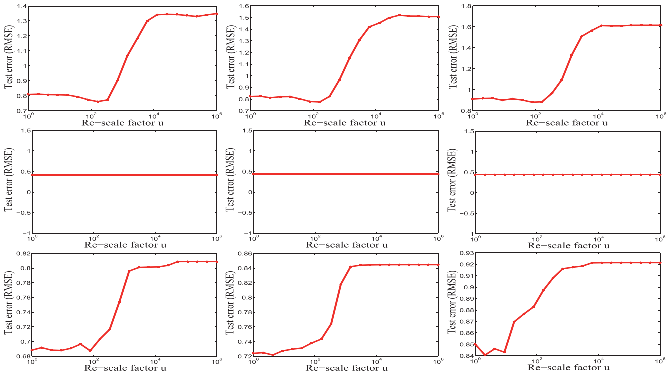


Fig. 3. Three rows denote the 10-D regression functions m_7, m_8, and m_9 and three columns indicate that the noise level σ varies among in $\{0, 0.5, 1\}$, respectively.

TABLE I

PERFORMANCE COMPARISON OF L_2-BOOSTING, L_2-RBOOSTING, AND L_2-DDRBOOSTING ON SIMULATED REGRESSION DATA SETS (1-D CASES)

	m_1	m_2	m_3
	$\sigma = 0$		
Boosting	0.0318(0.0069)	0.0895(0.0172)	0.0184(0.0025)
RBoosting	0.0308(0.0047)	0.0810(0.0183)	0.0179(0.0004)
DDRBoosting	**0.0268(0.0062)**	**0.0747(0.0232)**	**0.0178(0.0011)**
	$\sigma = 0.5$		
Boosting	0.2203(0.0161)	0.1925(0.0293)	0.2507(0.0336)
RBoosting	**0.2087(0.0181)**	**0.1665(0.0210)**	**0.2051(0.0252)**
DDRBoosting	0.2388(0.0114)	0.2142(0.0519)	0.2508(0.0179)
	$\sigma = 1$		
Boosting	0.3635(0.0467)	0.2943(0.0375)	0.3943(0.0415)
RBoosting	**0.3479(0.0304)**	**0.2558(0.0120)**	**0.3243(0.0355)**
DDRBoosting	0.3630(0.0315)	0.3787(0.0614)	0.4246(0.0360)

the shrinkage degree parameter and which data set we used. Considering the fact that we mainly focus on the impact of the shrinkage degree, the easiest way to do it is to select the theoretically optimal number of iterations via the test data set. More precisely, we select the number of iterations, k^*, as the best one according to $D_{m'}$ directly. Furthermore, for the additional shrinkage degree parameter, $\alpha_k = 2/(k+u)$, $u \in \mathbf{N}$, in L_2-RBoosting, we create 20 equally spaced values of u in the logarithmic space between 1 and 10^6.

All numerical studies are implemented using MATLAB R2014a on a Windows personal computer with Core i7-3770 3.40-GHz CPUs and RAM 4 GB, and the statistics are averaged based on 20 independent trails for each simulation.

2) Relationship Between Shrinkage Degree and Generalization Performance: For each given rescale factor $u \in [1, 10^6]$, we employ L_2-RBoosting to train the corresponding estimates on the whole training samples D_m, and then use the independent test samples $D_{m'}$ to evaluate their generalization performance. Figs. 1–3 show the performance curves of the L_2-RBoosting estimates for the aforementioned nine regression functions $m_1, \ldots, m_9$. It can be distinctly observed from Figs. 1–3 that, except for m_8, u has a great influence on the learning performance of L_2-RBoosting. Furthermore, the performance curves generally imply that there exists an optimal u, which may be not unique, to minimize the generalization error. This is consistent with our previous theoretical assertions. For m_8, the test error curve of L_2-RBoosting is flat with respect to u, that is, the generalization performance of L_2-RBoosting is irrelevant with u. The likely explanation for this observation is that the adopted weak learner is too strong (i.e., we preset the number of splits $J = 4$). Over grown tree trained on all samples are liable to autocracy and rescale operation does not bring performance benefits at all in such case. All numerical results illustrate the importance of selecting an appropriate shrinkage degree in L_2-RBoosting for regression tasks.

3) Performance Comparison of L_2-Boosting, L_2-RBoosting, and L_2-DDRBoosting: In this section, we compare the learning performances among L_2-Boosting, L_2-RBoosting, and L_2-DDRBoosting to analyze their cons and pons. Tables I–III document the generalization errors (RMSE) of L_2-Boosting, L_2-RBoosting, and L_2-DDRBoosting for regression functions $m_1, \ldots, m_9$, respectively. The bold numbers in Tables I–III denote the optimal performance, and the standard errors are also reported as the numbers in parentheses. From Tables I–III,

TABLE II

PERFORMANCE COMPARISON OF L_2-BOOSTING, L_2-RBOOSTING, AND L_2-DDRBOOSTING ON SIMULATED REGRESSION DATA SETS (2-D CASES)

	m_4	m_5	m_6
$\sigma = 0$			
Boosting	0.2125(0.0173)	0.2391(0.0140)	0.3761(0.0235)
RBoosting	**0.0582(0.0051)**	**0.0930(0.0133)**	**0.2161(0.0763)**
DDRBoosting	0.1298(0.0167)	0.1883(0.0216)	0.3585(0.0573)
$\sigma = 0.5$			
Boosting	0.3646(0.0152)	0.3693(0.0111)	0.4658(0.0233)
RBoosting	**0.2392(0.0223)**	**0.2665(0.0163)**	**0.3738(0.0323)**
DDRBoosting	0.3439(0.0317)	0.3344(0.0117)	0.5100(0.0788)
$\sigma = 1$			
Boosting	0.5250(0.0323)	0.3967(0.0317)	0.5966(0.0424)
RBoosting	**0.3836(0.0182)**	**0.3759(0.0231)**	**0.5066(0.0701)**
DDRBoosting	0.4918(0.0209)	0.4036(0.0180)	0.5638(0.0450)

TABLE III

PERFORMANCE COMPARISON OF L_2-BOOSTING, L_2-RBOOSTING, AND L_2-DDRBOOSTING ON SIMULATED REGRESSION DATA SETS (10-D CASES)

	m_7	m_8	m_9
$\sigma = 0$			
Boosting	1.4310(0.0412)	0.4167(0.0324)	0.8274(0.0142)
RBoosting	**0.7616(0.0144)**	**0.4167(0.0403)**	**0.6875(0.0107)**
DDRBoosting	1.1322(0.0696)	0.4178(0.0409)	0.8130(0.0330)
$\sigma = 0.5$			
Boosting	1.4450(0.0435)	0.4283(0.0314)	0.8579(0.0414)
RBoosting	**0.7755(0.0475)**	**0.4283(0.0401)**	**0.7218(0.0223)**
DDRBoosting	1.2526(0.0290)	0.4381(0.0304)	0.8385(0.0222)
$\sigma = 1$			
Boosting	1.4420(0.0413)	0.4404(0.0242)	0.8579(0.0415)
RBoosting	**0.8821(0.0575)**	**0.4404(0.0321)**	**0.8406(0.0175)**
DDRBoosting	1.4423(0.0625)	0.4503(0.0393)	0.9295(0.0120)

we can get the clear results that, except for the noiseless 1-D cases, the performance of L_2-RBoosting dominates both L_2-Boosting and L_2-DDRBoosting for all regression functions by a large margin. Through this series of numerical studies, including 27 different learning tasks, we can come up with the following guidelines. First, we verify the second guidance deduced from Theorem 8 that L_2-RBoosting outperforms L_2-Boosting with finite samples available. Second, although L_2-DDRBoosting can perfectly solve the parameter-selection problem in the rescale-type boosting algorithms, the comparative results also illustrate that L_2-RBoosting endows better performance once an appropriate u is selected.

4) Adaptive Parameter-Selection Strategy for Shrinkage Degree: We employ the simulations to verify the feasibility of the proposed parameter-selection strategy. As described in Section III-C, we randomly split the train samples $D_m = (X_i, Y_i)_{i=1}^{500}$ into two disjoint equal size subsets, i.e., a learning set and a validation set. We first train on the learning set D_m^l to construct the L_2-RBoosting estimates $f_{D_m^l,\alpha_k,k}$ and then use the validation set D_m^v to choose the appropriate shrinkage degree α_k^* and iteration k^* by minimizing the validation risk. Third, we retrain the obtained α_k^* on the entire training set D_m to construct $f_{D_m,\alpha_k^*,k}$ (in general, if we have enough training samples at hand, this step is optional). Finally, an independent test set of 1000 noiseless observations are used to evaluate the performance of $f_{D_m,\alpha_k^*,k}$.

TABLE IV

PERFORMANCE OF L_2-RBOOSTING VIA PARAMETER-SELECTION STRATEGY ON SIMULATED REGRESSION DATA SETS (1-D CASE)

	m_1	u	m_2	u	m_3	u

$\sigma = 0$	0.0317(0.0069)
	0.0308(0.0062)
		158(164)	0.0791(0.0262)
			0.0747(0.0232)
				5(2)	0.0180(0.0013)
					0.0178(0.0011)
						232(95)

$\sigma = 0.5$	0.2113(0.0122)
	0.2087(0.0181)
		609(160)	0.1766(0.0135)
			0.1665(0.0210)
				3(3)	0.2118(0.0094)
					0.2051(0.0252)
						3(2)

$\sigma = 1$	0.3487(0.0132)
	0.3479(0.0304)
		987(440)	0.2800(0.0308)
			0.2558(0.0120)
				1(0)	0.3302(0.0511)
					0.3243(0.0355)
						148(400)

TABLE V

PERFORMANCE OF L_2-RBOOSTING VIA PARAMETER-SELECTION STRATEGY ON SIMULATED REGRESSION DATA SETS (2-D CASE)

	m_4	u	m_5	u	m_6	u

$\sigma = 0$	0.0593(0.0059)
	0.0582(0.0051)
		55(34)	0.0958(0.0063)
			0.0930(0.0133)
				10(8)	0.2210(0.0143)
					0.2161(0.0763)
						3(2)

$\sigma = 0.5$	0.2511(0.0130)
	0.2392(0.0223)
		4(3)	0.2848(0.0201)
			0.2665(0.0163)
				142(300)	0.3869(0.0173)
					0.3738(0.0323)
						20(30)

$\sigma = 1$	0.4001(0.0179)
	0.3836(0.0182)
		6(7)	0.4007(0.0170)
			0.3759(0.0231)
				2(1)	0.5123(0.0925)
					0.5066(0.0701)
						6(7)

TABLE VI

PERFORMANCE OF L_2-RBOOSTING VIA PARAMETER-SELECTION STRATEGY ON SIMULATED REGRESSION DATA SETS (10-D CASE)

	m_7	u	m_8	u	m_9	u

$\sigma = 0$	0.7765(0.0259)
	0.7616(0.0144)
		66(65)	0.4169(0.0313)
			0.4167(0.0403)
				$\diagdown$	0.6882(0.0113)
					0.6875(0.0107)
						42(36)

$\sigma = 0.5$	0.7757(0.0128)
	0.7755(0.0475)
		72(59)	0.4349(0.0335)
			0.4283(0.0401)
				$\diagdown$	0.7396(0.0145)
					0.7218(0.0223)
						11(9)

$\sigma = 1$	0.9093(0.0304)
	0.8821(0.0575)
		38(37)	0.4452(0.0308)
			0.4404(0.0321)
				$\diagdown$	0.8539(0.0278)
					0.8406(0.0175)
						23(30)

Tables IV–VI document the test errors (RMSE) for regression functions $m_1, \ldots, m_9$. The corresponding bold numbers denote the ideal generalization performance of the L_2-RBoosting (choose optimal iteration k^* and optimal shrinkage degree α_k^* both according to minimize the test error via the test sets). We also report the standard errors (numbers in parentheses) of selected rescale parameter u over 20 independent runs in order to check the stability of such parameter-selection strategy. From Tables IV–VI, we can easily find that the performance with such strategy approximates the ideal one. More important, comparing the mean values and standard errors of u with the performance curves in Figs. 1–3, apart

TABLE VII
PERFORMANCE COMPARISON OF DIFFERENT BOOSTING-TYPE ALGORITHMS ON REAL DATA SETS

Datasets Methods	Diabetes	Prostate	Housing	CCS	Abalone
Decision stumps					
Boosting	61.9663(2.4743)	0.3955(0.1174)	4.6692(0.2858)	6.2001(0.2143)	2.4359(0.0705)
RBoosting	58.7133(1.2158)	0.2245(0.1176)	**4.1267(0.2573)**	5.4721(0.1172)	**2.2761(0.0231)**
DDRBoosting	60.1348(1.9478)	0.3005(0.0946)	4.4375(0.3395)	6.1245(0.1815)	2.3710(0.0632)
ϵ-Boosting	58.9357(1.9287)	0.2282(0.0537)	4.3340(0.2362)	6.0992(0.6623)	2.4037(0.0486)
RSBoosting	59.8292(2.3190)	**0.2241(0.1429)**	4.3568(0.2131)	6.0231(0.3800)	2.41622(0.0921)
RTBoosting	**58.6075(2.6703)**	0.2885(0.0958)	4.1376(0.3473)	**5.3540(0.3443)**	2.2781(0.0472)
Vanilla neural networks					
Boosting	61.8423(2.5464)	0.6603(0.1479)	4.6566(0.8436)	6.7445(0.3740)	2.1419(0.0826)
RBoosting	**58.0272(2.3606)**	0.5498(0.1106)	**4.0174(0.3211)**	6.5948(0.3903)	2.1190(0.0466)
DDRBoosting	58.1272(2.9780)	0.5822(0.1162)	4.6083(0.6048)	6.6454(0.3535)	2.1391(0.0404)
ϵ-Boosting	58.0321(2.3816)	**0.5225(0.1216)**	4.4745(0.3395)	6.6219(0.2298)	**2.1005(0.0322)**
RSBoosting	58.1244(2.4471)	0.5375(0.1064)	4.5365(0.3291)	6.6231(0.2484)	2.1208(0.0237)
RTBoosting	58.0345(2.3478)	0.5751(0.0946)	4.4454(0.2935)	**6.5231(0.2581)**	2.1301(0.0212)

from m_8, we can distinctly detect that the selected u values by the proposed parameter-selection strategy are all located near the low valleys.

B. Real Data Experiments

We have verified that L_2-RBoosting outperforms L_2-Boosting and L_2-DDRBoosting on the $3 \times 9 = 27$ different distributions in the previous simulations. Now, we further compare the learning performances of these algorithms with other popular boosted-type schemes, including ϵ-Boosting [13], RSBoosting [1], and RTBoosting [12] on five real data sets.

The first data set is the Diabetes data set [10]. This data set contains 442 diabetes patients that were measured on ten independent variables, i.e., age, sex, body mass index, and so on and one response variable, i.e., a measure of disease progression. The second one is the Prostate Cancer data set derived from a study of prostate cancer in [35]. The data set consists of the medical records of 97 patients who were about to receive a radical prostatectomy. The predictors are eight clinical measures, i.e., cancer volume, prostate weight, age, and so on and one response variable, i.e., the logarithm of prostate-specific antigen. The third one is the Boston Housing data set created form a housing values survey in suburbs of Boston in [36]. This data set contains 506 instances, which include 13 attributions, i.e., per capita crime rate by town, proportion of nonretail business acres per town, average number of rooms per dwelling, and so on and one response variable, i.e., median value of owner-occupied homes. The fourth one is the concrete compressive strength (CCS) data set created from [37]. The data set contains 1030 instances, including eight quantitative independent variables, i.e., age and ingredients, and so on and one dependent variable, i.e., quantitative CCS. The fifth one is the Abalone data set, which comes from an original study in [38] for predicting the age of abalone from physical measurements. The data set contains 4177 instances, which were measured on eight independent variables, i.e., length, sex, height, and so on and one response variable, i.e., the number of rings.

Similarly, we randomly divide all the real data sets into two disjoint equal parts. The first half serves as the training set and the second half serves as the test set. We parameterize both the number of iterations and the value of shrinkage degrees within the specified interval (i.e., k in $[0, 10^4]$ and u in $[1, 10^6]$) and then adopt the method of twofold cross validation based on grid search for tuning corresponding parameters. We also utilize the Z-score standardization method [39] to normalize the data sets, in order to avoid the error caused by considerable magnitude difference among data dimensions.

For each real data experiment, weak learners are first changed to the decision stumps (specifying one split of each tree, $J = 1$) corresponding to an additive model with only main effects. Then, weak learners are changed to the vanilla neural networks to further show that the proposed approach can also boost the performance of neural networks. The neural networks are set with one sigmoid hidden layer, where the input units equal to the dimension of samples, the hidden units is set to five, and the output units (with affine transformation) is the same as the dimension of the labels, and the backpropagation algorithm is employed to train each neural network.

Table VII documents the performance (test RMSE) comparison results of different boosting-type algorithms on five real data sets, respectively (the bold numbers denote the optimal performance). We can observe from Table VII that the performance of L_2-RBoosting with u selected via our recommended strategy outperforms both L_2-Boosting and L_2-DDRBoosting on all real data sets, especially for some data sets, i.e., diabetes, prostate, and CCS, making a large improvement. It is consistent with the previous toy simulations and, therefore, experimentally verifies our theoretical assertions. Furthermore, it can be found that all the boosting variants outperform the original boosting algorithm to some extent and L_2-RBoosting generally performs as the second best algorithm among all the variants. It indicates that the idea of rescale in boosting is feasible and comparable with the idea of regularization in other variants, which provides a new direction to improve the performance of boosting.

V. Conclusion and Discussion

In this paper, we draw a concrete analysis concerning L_2-RBoosting and how to determine the shrinkage degree in L_2-RBoosting. The contributions can be concluded in six aspects. First, we derived the optimal numerical convergence rate of L_2-RBoosting. Second, we deduced the generalization error bound of L_2-RBoosting and demonstrated the importance of the shrinkage degree. It was shown that, under certain conditions, the learning rate of L_2-RBoosting can reach $O(m^{-1/2}\log m)$, which is the same as the optimal record for the greedy-type learning and boosting-type algorithms. Furthermore, our results showed that although the shrinkage degree did not affect the learning rate, it determined the constant of the generalization error bound and, therefore, played a crucial role in L_2-RBoosting learning with finite samples. Third, we proposed two schemes to determine the shrinkage degree. The first one is the parameterized L_2-RBoosting, and the other one is to learn the shrinkage degree from the samples directly (L_2-DDRBoosting). We further provided the theoretical optimality of these approaches. Fourth, we compared these two approaches and proved that, although L_2-DDRBoosting avoided introducing additional parameters, the estimator deduced from L_2-RBoosting possessed a better structure ($\mathcal{L}_1$ norm). Therefore, for some special weak learners, L_2-RBoosting can achieve better performance than L_2-DDRBoosting. Fifth, we developed an adaptive parameter-selection strategy for the shrinkage degree. Our theoretical results demonstrated that L_2-RBoosting with such a shrinkage degree selection strategy did not degrade the generalization capability very much. Finally, a series of numerical simulations and real data experiments have been carried out to verify our theoretical assertions. The obtained results enhanced the understanding of RBoosting and could provide guidance on how to utilize L_2-RBoosting for regression tasks.

Eventually, we present the following three remarks at the end of this paper to demonstrate the generality and potential research or application directions of RBoosting.

Remark 17: RBoosting is a variant of boosting, which can essentially improve the generalization performance of boosting learning. According to [17, Remark 1], it is also a general idea for promoting other regularized boosting-type algorithms, such as RSBoosting [11], RTBoosting [12], and ε-Boosting [13]. Some preliminary results about such synthesized new boosting-type algorithms are shown in Fig. 4, where the simulation settings are the same as described previously in Section IV. The ϵ value is chosen from a 20 points set whose elements are uniformly localized in [0.01, 1] for ϵ-Boosting, and the truncated parameter value in RTboosting is set as that in [12]. The results illustrate that, if the two additional parameters appropriately selected, then the performance of such a synthesized new boosting-type algorithm can be further improved. However, the current primary difficulties lies in how to adjust two additional parameters simultaneously and how to balance the performance and additional cost (may be tremendous) in practice. We will keep working on this issue and report our progress in a future publication.

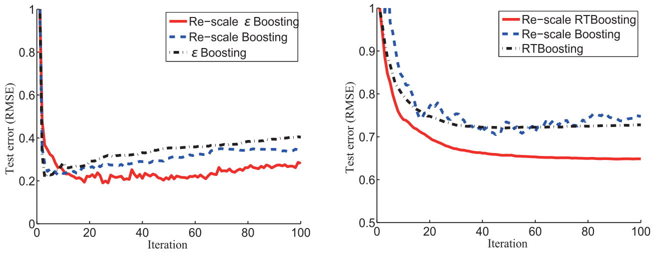


Fig. 4. Performance of synthesized new boosting-type algorithm. Left: rescale ε-Boosting for fitting regression function m_1. Right: rescale RTBoosting for fitting regression function m_9. The noise level $\sigma = 0.5$.

Remark 18: According to [17], the general idea of RBoosting is feasible for arbitrary convex loss and can be used to tackle both regression and classification tasks. Thus, we also present some empirical results to reveal the relationship between the shrinkage degree and the generalization performance in RBoosting for classification. In the case of categorical response, the response variable y typically takes on binary values $y \in \{0, 1\}$, and thus, two popular choice of categorical loss functions are utilized in boosting for classification [1]. The one is commonly referred to as the Bernoulli loss (or logistic loss) [1] $\Phi(y, f)_{\text{Logit}} = \log(1+\exp(-2yf))$, as it is employed in the Logitboost algorithm [40]. Another common choice is the exponential loss function $\Phi(y, f)_{\text{Ada}} = \exp(-yf)$, as used in the Adaboost algorithm [41].

Both two categorical loss functions are utilized to illustrating the relationship between the shrinkage degree and the classification error. Two real data sets are considered for classification. The first data set is the Banknote Authentication data set [42], which contains 1372 instances that were measured on 5 independent variables. Data were extracted from images that were taken for the evaluation of an authentication procedure for banknotes. Another data set is Breast Cancer Wisconsin (Original) data set [43], which contains 699 instances with 10 features. It is used to identify whether a patient suffered from breast cancer. In addition, the other settings are the same as described previously in Section IV-B.

It can be observed in Fig. 5 that the shrinkage degree still has a great influence on the classification error (or generalization performance) in Logitboost and Adaboost, which is consistent with our analysis on L_2-Boosting. It implies that the rescale operator can be used to improve the performance of boosting in both the regression and classification tasks.

The aim of this paper is to develop a concrete analysis only concerning regression tasks, which is vital for the following reasons. First, our theoretical results (Theorems 7–9) for RBoosting and DDRBoosting are specific to the L_2 loss. For other loss functions, the theoretical behaviors need to be further pursued. Second, as far as the L_2 loss is concerned, it is easy to deduce the close-form representation of the estimator of DDRBoosting [27]. For other loss functions, we have not found any works concerning the analytical solvability of the optimization problem defined in step 3 of the DDRBoosting algorithm. Finally, compared with logistic loss, exponential

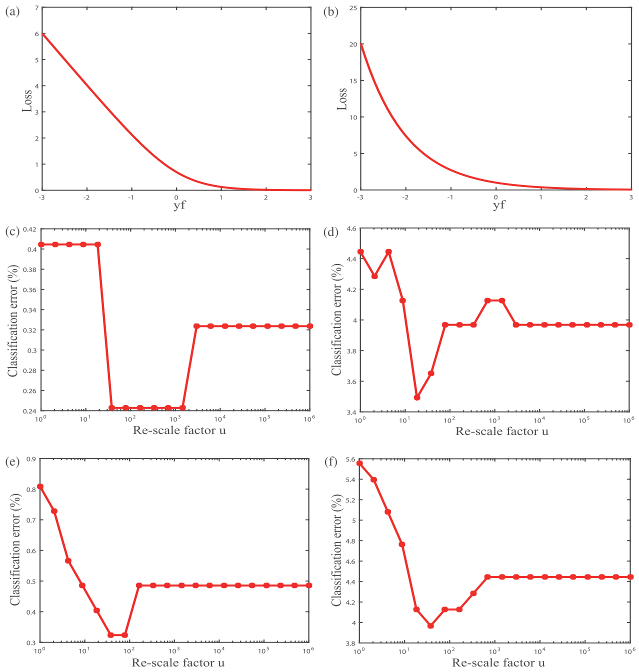


Fig. 5. Relationship between shrinkage degree and classification error in Logitboost and Adaboost. (a) Bernoulli loss function. (b) Exponential loss function. (c) and (d) Rescale Logitboost performs on Banknote Authentication and Breast Cancer Wisconsin (Original) data sets. (e) and (f) Rescale Adaboost performs on Banknote Authentication and Breast Cancer Wisconsin (Original) data sets.

loss or hinge loss and L_2 loss (or squared loss) are not commonly employed for the classification and maybe badly behaved [1], [18]. That is why, we only focus on L_2-RBoosting for the regression tasks in this paper.

Remark 19: To facilitate the use of RBoosting, a feasible parameter-selection strategy is crucial. For this purpose, we developed two approaches for selecting shrinkage degree for L_2-RBoosting in this paper. L_2-DDRBoosting can improve the performance of the boosting without tuning additional parameters, which is superior to the other variants of boosting. However, the fact that parameterized L_2-RBoosting can work better than L_2-DDRBoosting is an empirical observation, despite the fact that their theoretical behavior is equal. We showed that the performance of L_2-DDRBoosting cannot be guaranteed in some special cases, such as data with high-level noise or the previous estimate f'_{k-1}, which is highly linear correlate with the current selected atom g_k^*. This motivated us to further study some model selection strategies for parameterized L_2-RBoosting and further developed an adaptive parameter-selection strategy for it. Assuredly, it should pay more computational complexity for training and may be unsuitable for some real-time request applications. Nonetheless, we supposed that RBoosting may be applicable for some high-performance requirement tasks, such as anomaly detection, information retrieval, and so on. In addition, in the recent work, Yang *et al.* [44] have already tried to apply RBoosting for attack detection in collaborative filtering recommender systems, and the results demonstrated that it is competent for such tasks. We plan to do more application studies for RBoosting in our future work.

Appendix A
Proof of Theorem 7

To prove Theorem 7, we need the following two lemmas. The first one can be found in [17], which is a direct generalization of [15, Lemma 2.3].

Lemma 20: Let $j_0 > 2$ be a natural number. Suppose that three positive numbers $c_1 < c_2 \leq j_0$, $\mathcal{C}_0$ be given. Assume that a sequence $\{a_n\}_{n=1}^{\infty}$ has the following two properties.

1) For all $1 \leq n \leq j_0$

$$a_n \leq \mathcal{C}_0 n^{-c_1}$$

and, for all $n \geq j_0$

$$a_n \leq a_{n-1} + \mathcal{C}_0(n-1)^{-c_1}.$$

2) If for some $\upsilon \geq j_0$, we have

$$a_\upsilon \geq \mathcal{C}_0 \upsilon^{-c_1}$$

then

$$a_{\upsilon+1} \leq a_\upsilon(1 - c_2/\upsilon).$$

Then, for all $n = 1, 2, \ldots,$ we have

$$a_n \leq 2^{1+\frac{c_1^2+c_1}{c_2-c_1}} \mathcal{C}_0 n^{-c_1}.$$

The second one can be easily deduced from [15, Lemma 2.2].

Lemma 21: Let $h \in \operatorname{span}(S)$, $f_{,k}$ be the estimate defined in Algorithm 2 and $y(\cdot)$ is an arbitrary function satisfying $y(x_i) = y_i$. Then, for arbitrary $k = 1, 2, \ldots,$ we have

$$\|f_k - y\|_m \leq \|f_{k-1} - y\|_m \left(1 - \alpha_k\left(1 - \frac{\|y-h\|_m}{\|f_{k-1} - y\|_m}\right) + 2\left(\frac{\alpha_k(\|y\|_m + \|h\|_{\mathcal{L}_1(S)})}{(1-\alpha_k)\|f_{k-1} - y\|_m}\right)^2\right).$$

Now, we are in a position to present the proof of Theorem 7.

Proof of Theorem 7: By Lemma 21, for $k \geq 1$, we obtain

$$\begin{aligned}\|f_k - y\|_m - \|y - h\|_m &\leq (1-\alpha_k)(\|f_{k-1} - y\|_m - \|y-h\|_m)\\ &\quad + C\|f_{k-1} - y\|_m \left(\frac{\alpha_k(\|y\|_m + \|h\|_{\mathcal{L}_1(S)})}{\|f_{k-1} - y\|_m}\right)^2.\end{aligned}$$

Let

$$a_{k+1} = \|f_k - y\|_m - \|y - h\|_m.$$

Then, by noting $\|y\|_m \leq M$, we have

$$a_{k+1} \leq (1-\alpha_k)a_k + C\frac{\alpha_k^2(M + \|h\|_{\mathcal{L}_1(S)})^2}{a_k}.$$

We plan to apply Lemma 20 to the sequence $\{a_n\}$. Let $\mathcal{C}_0 = \max\{1, \sqrt{C}\}2(M + \|h\|_{l^1(\mathcal{D}_n)})$. According to the definitions of $\{a_k\}_{k=1}^{\infty}$ and f_k, we obtain

$$a_1 = \|y\|_m - \|y - h\|_m \leq 2M + \|h\|_{\mathcal{L}_1(S)} \leq \mathcal{C}_0$$

and

$$a_{k+1} \leq a_k + \alpha_k\|y\|_m \leq a_k + \mathcal{C}_0 k^{-1/2}.$$

Let $a_k \geq \mathcal{C}_0 k^{-1/2}$, since $\alpha_k = 2/k + u$, we then obtain

$$a_k \leq \frac{k+u-2}{k+u} a_{k-1} + C a_{k-1} k \frac{4}{\mathcal{C}_0^2 (k+u)^2} (M + \|h\|_{\mathcal{L}_1(S)})^2$$

that is

$$\begin{aligned} a_k &\leq a_{k-1} \left(1 - \frac{2}{k+u} + C \frac{4k}{\mathcal{C}_0^2 (k+u)^2} (M + \|h\|_{\mathcal{L}_1(S)})^2 \right) \\ &\leq \left(1 - \left(\frac{1}{2} + \frac{2u+2}{(2+u)^2} \right) \frac{1}{k-1} \right). \end{aligned}$$

Now, it follows from Lemma 20 with $c_1 = (1/2)$ and $c_2 = (1/2) + (2u + 2/(2+u)^2)$ that:

$$a_n \leq \max\{1, \sqrt{C}\} 2(M + \|h\|_{\mathcal{L}_1(S)}) 2^{1+\frac{3(u+2)^2}{8u+8}} n^{-1/2}.$$

Therefore, we obtain

$$\|f_k - y\|_m \leq \|y - h\|_m + (M + \|h\|_{\mathcal{L}_1(S)}) 2^{\frac{3u^2+14u+20}{8u+8}} k^{-1/2}.$$

This finishes the proof of Theorem 7. ■

Appendix B
Proof of Theorem 8

To prove Theorem 8, we shall give an error decomposition strategy for $\mathcal{E}(\pi_M f_k) - \mathcal{E}(f_\rho)$. The method is somewhat standard and similar to the proof [25], [45]. Let f_k be defined as in Algorithm 2 and arbitrary $f^* \in \text{span}S$. Direct computation yields

$$\begin{aligned} &\mathcal{E}(\pi_M f_k) - \mathcal{E}(f_\rho) \\ &\quad = \mathcal{E}(f^*) - \mathcal{E}(f_\rho) + \mathcal{E}_{\mathbf{z}}(\pi_M f_k) - \mathcal{E}_D(f^*) \\ &\qquad + \mathcal{E}_D(f^*) - \mathcal{E}(f^*) + \mathcal{E}(\pi_M f_k) - \mathcal{E}_D(\pi_M f_k). \end{aligned}$$

Upon making the shorthand notations

$$\begin{aligned} \mathcal{D}(k) &:= \mathcal{E}(f^*) - \mathcal{E}(f_\rho) \\ \mathcal{S}(D,k) &:= \mathcal{E}_D(f^*) - \mathcal{E}(f^*) + \mathcal{E}(\pi_M f_k) - \mathcal{E}_D(\pi_M f_k) \end{aligned}$$

and

$$\mathcal{P}(D,k) := \mathcal{E}_D(\pi_M f_k) - \mathcal{E}_D(f^*)$$

respectively, for the approximation error, sample error, and hypothesis error, we have

$$\mathcal{E}(\pi_M f_k) - \mathcal{E}(f_\rho) = \mathcal{D}(k) + \mathcal{S}(D,k) + \mathcal{P}(D,k). \quad \text{(B.1)}$$

In order to give a bound for $\mathcal{D}(k)$, we need to use Lemma 22, which can be easily deduced from [15] and [25, Proposition 1, Lemma 1].

Lemma 22: If $f_\rho \in \mathcal{L}_1^r$, then there exists an $f^* \in \text{span}S$, such that $\|f^*\|_{\mathcal{L}_1(S)} \leq \mathcal{B}$ and

$$\mathcal{D}(k) \leq \mathcal{B}^2 (k^{-1/2} + n^{-r})^2. \quad \text{(B.2)}$$

Now, we proceed the proof of Theorem 8.

Proof of Theorem 8: Based on Theorem 7 and the fact $\|f^*\|_{\mathcal{L}_1(S)} \leq \mathcal{B}$, we obtain

$$\begin{aligned} \mathcal{P}(D,k) &\leq 2\mathcal{E}_D(\pi_M f_k) - \mathcal{E}_D\big(f_k^*\big) \\ &\leq 2(M + \mathcal{B})^2 2^{\frac{3u^2+14u+20}{8u+8}} k^{-1}. \quad \text{(B.3)} \end{aligned}$$

Therefore, both the approximation error and hypothesis error are deduced. The only thing remainder is to bound the sample error $\mathcal{S}(D,k)$. Upon using the shorthand notations

$$S_1(D,k) := \{\mathcal{E}_D(f_k^*) - \mathcal{E}_D(f_\rho)\} - \{\mathcal{E}(f_k^*) - \mathcal{E}(f_\rho)\}$$

and

$$S_2(D,k) := \{\mathcal{E}(\pi_M f_k) - \mathcal{E}(f_\rho)\} - \{\mathcal{E}_D(\pi_M f_k) - \mathcal{E}_D(f_\rho)\}$$

we write

$$\mathcal{S}(D,k) = \mathcal{S}_1(D,k) + \mathcal{S}_2(D,k). \quad \text{(B.4)}$$

It can be found in [25, Proposition 2] that for any $0 < t < 1$, with confidence $1 - (t/2)$

$$\mathcal{S}_1(D,k) \leq \frac{7\left(3M + \mathcal{B}\log\frac{2}{t}\right)}{3m} + \frac{1}{2}\mathcal{D}(k). \quad \text{(B.5)}$$

It also follows from [46, eq. (A.10)] that:

$$\mathcal{S}_2(D,k) \leq \frac{1}{2}\mathcal{E}(\pi_M f_k) - \mathcal{E}(f_\rho) + \log\frac{2}{t}\frac{Ck\log m}{m} \quad \text{(B.6)}$$

holds with confidence at least $1 - t/2$. Therefore, (B.1)–(B.6) yield that

$$\begin{aligned} &\mathcal{E}(\pi_M f_k) - \mathcal{E}(f_\rho) \\ &\leq C(M+\mathcal{B})^2 \left(2^{\frac{3u^2+14u+20}{8u+8}} k^{-1} + (m/k)^{-1} \log m \log\frac{2}{t} + n^{-2r} \right) \end{aligned}$$

holds with confidence at least $1 - t$. This finishes the proof of Theorem 8. ■

Appendix C
Proof of Theorem 9

Proof of Theorem 9: It can be deduced from [15, Th. 1.2] and the same method as in the proof of Theorem 8. For the sake of brevity, we omit the details. ■

Appendix D
Proof of Proposition 10

Proof of Proposition 10: It is easy to check that

$$f_k = (1 - \alpha_k) f_{k-1} + \langle y - (1 - \alpha_k) f_{k-1}, g_k \rangle_2 g_k.$$

As $\|g_k\| \leq 1$, we obtain from the Hölder inequality that

$$\begin{aligned} \langle y - (1-\alpha_k) f_{k-1}, g_k \rangle_2 &\leq \|y - (1-\alpha_k) f_{k-1}\|_2 \\ &\leq (1-\alpha_k)\|y - f_{k-1}\|_2 + \alpha_k M. \end{aligned}$$

As

$$\|y - f_{k-1}\|_2 \leq C(M + \|h\|_{\mathcal{L}_1(S)}) k^{-1/2} + n^{-r}$$

we can obtain

$$\|f_k\|_1 \leq C((M + \|h\|_{\mathcal{L}_1(S)}) k^{1/2} + k n^{-r}).$$

This finishes the proof of Proposition 10. ■

Appendix E
Proof of Theorem 14

Now, we turn to prove Theorem 14. The following concentration inequality [47] plays a crucial role in the proof.

Lemma 23: Let $\mathcal{F}$ be a class of measurable functions on Z. Assume that there are constants $B, c > 0$ and $\alpha \in [0, 1]$, such that $\|f\|_\infty \leq B$ and $\mathbf{E}f^2 \leq c(\mathbf{E}f)^\alpha$ for every $f \in \mathcal{F}$. If for some $a > 0$ and $\mu \in (0, 2)$

$$\log \mathcal{N}_2(\mathcal{F}, \varepsilon) \leq a\varepsilon^{-\mu} \quad \forall \varepsilon > 0 \tag{E.1}$$

then there exists a constant c'_p depending only on p, such that for any $t > 0$, with probability at least $1 - e^{-t}$, there holds

$$\mathbf{E}f - \frac{1}{m}\sum_{i=1}^{m} f(z_i) \leq \frac{1}{2}\eta^{1-\alpha}(\mathbf{E}f)^\alpha + c'_\mu \eta + 2\left(\frac{ct}{m}\right)^{\frac{1}{2-\alpha}} + \frac{18Bt}{m} \quad \forall f \in \mathcal{F} \tag{E.2}$$

where

$$\eta := \max\left\{c^{\frac{2-\mu}{4-2\alpha+\mu\alpha}}\left(\frac{a}{m}\right)^{\frac{2}{4-2\alpha+\mu\alpha}}, B^{\frac{2-\mu}{2+\mu}}\left(\frac{a}{m}\right)^{\frac{2}{2+\mu}}\right\}.$$

We continue the proof of Theorem 14.

Proof of Theorem 14: For arbitrary $h \in \mathrm{span}(S)$

$$\mathcal{E}(f_k) - \mathcal{E}(h) = \mathcal{E}(f_k) - \mathcal{E}(h) - (\mathcal{E}_D(f_k) - \mathcal{E}_D(h)) + \mathcal{E}_D(f_k) - \mathcal{E}_D(h).$$

Set

$$\mathcal{G}_R := \{g(z) = (\pi_M f(x) - y)^2 - (h(x) - y)^2 : f \in B_R\}. \tag{E.3}$$

Using the obvious inequalities $\|\pi_M f\|_\infty \leq M$, $|y| \leq M$ a.e., we get the inequalities

$$|g(z)| \leq (3M + \|h\|_{\mathcal{L}_1(S)})^2$$

and

$$\mathbf{E}g^2 \leq (3M + \|h\|_{\mathcal{L}_1(S)})^2\mathbf{E}g.$$

For $g_1, g_2 \in \mathcal{G}_R$, it follows that:

$$|g_1(z) - g_2(z)| \leq (3M + \|h\|_{\mathcal{L}_1(S)})|f_1(x) - f_2(x)|.$$

Then

$$\mathcal{N}_2(\mathcal{G}_R, \varepsilon) \leq \mathcal{N}_{2,\mathbf{x}}\left(B_R, \frac{\varepsilon}{3M + \|h\|_{\mathcal{L}_1(S)}}\right) \leq \mathcal{N}_{2,\mathbf{x}}\left(B_1, \frac{\varepsilon}{R(3M + \|h\|_{\mathcal{L}_1(S)})}\right).$$

Using the inequality and Assumption 13, we have

$$\log \mathcal{N}_2(\mathcal{F}_R, \varepsilon) \leq \mathcal{L}(R(3M + \|h\|_{\mathcal{L}_1(S)}))^\mu \varepsilon^{-\mu}.$$

By Lemma 23 with $B = c = (3M + \|h\|_{\mathcal{L}_1(S)})^2$, $\alpha = 1$, and $a = \mathcal{L}(R(3M + \|h\|_1))^\mu$, we know that for any $t \in (0, 1)$, with confidence $1 - (t/2)$, there exists a constant C depending only on d, such that for all $g \in \mathcal{G}_R$

$$\mathbf{E}g - \frac{1}{m}\sum_{i=1}^{m} g(z_i) \leq \frac{1}{2}\mathbf{E}g + c'_\mu \eta + 20(3M + \|h\|_{\mathcal{L}_1(S)})^2 \frac{\log 4/t}{m}.$$

Here

$$\eta = ((3M + \|h\|_{\mathcal{L}_1(S)})^2)^{\frac{2-\mu}{2+\mu}}\left(\frac{\mathcal{L}(R(3M + \|h\|_{\mathcal{L}_1(S)}))^\mu}{m}\right)^{\frac{2-\mu}{2+\mu}}.$$

It then follows from Proposition 10 that:

$$\mathcal{E}(\pi_M f_k) - \mathcal{E}(f_\rho) \leq C \log\frac{2}{t}(3M + \mathcal{B})^2\left(n^{-2r} + k^{-1} + \left(\frac{(kn^{-r} + \sqrt{k})^\mu}{m}\right)^{\frac{2-\mu}{2+\mu}}\right).$$

This finishes the proof of Theorem 14. ■

Acknowledgment

Three anonymous reviewers have carefully read this paper and have provided to us numerous constructive suggestions. As a result, the overall quality of this paper has been noticeably enhanced, to which we feel much indebted and are grateful.

References

[1] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.

[2] P. Bühlmann and B. Yu, “Boosting with the L_2 loss: Regression and classification,” *J. Amer. Statist. Assoc.*, vol. 98, no. 462, pp. 324–339, 2003.

[3] 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.*, vol. 28, no. 2, pp. 337–407, 2000.

[4] P. Bühlmann and T. Hothorn, “Boosting algorithms: Regularization, prediction and model fitting,” *Statist. Sci.*, vol. 22, no. 4, pp. 477–505, Nov. 2007.

[5] N. Duffy and D. Helmbold, “Boosting methods for regression,” *Mach. Learn.*, vol. 47, no. 2, pp. 153–200, May 2002.

[6] Y. Freund, “Boosting a weak learning algorithm by majority,” *Inf. Comput.*, vol. 121, no. 2, pp. 256–285, Sep. 1995.

[7] R. E. Schapire, “The strength of weak learnability,” *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, Jun. 1990.

[8] P. L. Bartlett and M. Traskin, “AdaBoost is consistent,” *J. Mach. Learn. Res.*, vol. 8, no. 1, pp. 2347–2368, 2007.

[9] E. D. Livshits, “Lower bounds for the rate of convergence of greedy algorithms,” *Izvestiya, Math.*, vol. 73, no. 6, pp. 1197–1215, 2009.

[10] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression,” *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[11] J. Ehrlinger and H. Ishwaran, “Characterizing L_2 boosting,” *Ann. Statist.*, vol. 40, no. 2, pp. 1074–1101, 2012.

[12] T. Zhang and B. Yu, “Boosting with early stopping: Convergence and consistency,” *Ann. Statist.*, vol. 33, no. 4, pp. 1538–1579, Aug. 2005.

[13] T. Hastie, J. Taylor, R. Tibshirani, and G. Walther, “Forward stagewise regression and the monotone lasso,” *Electron. J. Statist.*, vol. 1, pp. 1–29, Apr. 2007.

[14] P. Zhao and B. Yu, “Stagewise lasso,” *J. Mach. Learn. Res.*, vol. 8, pp. 2701–2726, Dec. 2007.

[15] V. N. Temlyakov, “Relaxation in greedy approximation,” *Constructive Approx.*, vol. 28, no. 1, pp. 1–25, Jun. 2008.

[16] T. Zhang, “Sequential greedy approximation for certain convex optimization problems,” *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 682–691, Mar. 2003.

[17] S. Lin, Y. Wang, and L. Xu. (2015). “Re-scale boosting for regression and classification.” [Online]. Available: https://arxiv.org/abs/1505.01371

[18] J. Elith, J. R. Leathwick, and T. Hastie, “A working guide to boosted regression trees,” *J. Animal Ecol.*, vol. 77, no. 4, pp. 802–813, Jul. 2008.

[19] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[20] P. Smyth and D. Wolpert, "Linearly combining density estimators via stacking," *Mach. Learn.*, vol. 36, no. 1, pp. 59–83, Jul. 1999.

[21] D. J. C. MacKay, "Bayesian methods for adaptive models," Ph.D. dissertation, Dept. Comput. Neural Syst., California Inst. Technol., Pasadena, CA, USA, 1991.

[22] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[23] A. M. Bagirov, C. Clausen, and M. Kohler, "An L_2-boosting algorithm for estimation of a regression function," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1417–1429, Mar. 2010.

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

[25] S. Lin, Y. Rong, X. Sun, and Z. Xu, "Learning capability of relaxed greedy algorithms," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1598–1608, Oct. 2013.

[26] R. A. DeVore and V. N. Temlyakov, "Some remarks on greedy algorithms," *Adv. Comput. Math.*, vol. 5, no. 1, pp. 173–187, Dec. 1996.

[27] V. N. Temlyakov, "Greedy approximation," *Acta Numer.*, vol. 17, pp. 235–409, May 2008.

[28] A. R. Barron, A. Cohen, W. Dahmen, and R. A. DeVore, "Approximation and learning by greedy algorithms," *Ann. Statist.*, vol. 36, no. 1, pp. 64–94, Feb. 2008.

[29] V. N. Temlyakov, "Greedy approximation in convex optimization," *Constructive Approx.*, vol. 41, no. 2, pp. 269–296, Jan. 2015.

[30] D.-X. Zhou and K. Jetter, "Approximation with polynomial kernels and SVM classifiers," *Adv. Comput. Math.*, vol. 25, no. 1, pp. 323–344, Jul. 2006.

[31] L. Shi, Y.-L. Feng, and D.-X. Zhou, "Concentration estimates for learning with ℓ^1-regularizer and data dependent hypothesis spaces," *Appl. Comput. Harmon. Anal.*, vol. 31, no. 2, pp. 286–302, Sep. 2011.

[32] L. Györfi, M. Kohler, A. Krzyzak, and H. Walk, *A Distribution-Free Theory of Nonparametric Regression*. NY, USA: Springer Science & Business Media, 2006.

[33] J. Friedman and T. Hastie, *The Elements of Statistical Learning*. Berlin, Germany: Springer, 2001.

[34] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Boca Raton, FL, USA: CRC Press, 1984.

[35] T. A. Stamey *et al.*, "Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients," *J. Urol.*, vol. 141, no. 5, pp. 1076–1083, May. 1989.

[36] D. Harrison, Jr., and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *J. Environ. Econ.*, vol. 5, no. 1, pp. 81–102, Mar. 1978.

[37] I.-C. Yeh, "Modeling of strength of high-performance concrete using artificial neural networks," *Cement Concrete Res.*, vol. 28, no. 12, pp. 1797–1808, Dec. 1998.

[38] W. J. Nash *et al.*, "The population biology of abalone (haliotis species) in tasmania. I. Blacklip abalone (H. rubra) from the north coast and islands of bass strait," Dept. Primary Ind. Fisheries, Sea Fisheries Division, Marine Res. Lab.-Taroona, Tasmania, Tech. Rep. 48, 1994.

[39] E. Kreyszig, *Applied Mathematics*. NY, USA: Wiley, 1979.

[40] S. B. Kotsiantis, "Logitboost of simple Bayesian classifier," *Informatica*, vol. 29, no. 1, pp. 53–59, Nov. 2004.

[41] R. E. Schapire, "The boosting approach to machine learning: An overview," in *Nonlinear Estimation and Classification*. NY, USA: Springer, 2003, pp. 149–171.

[42] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[43] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, "Breast cancer diagnosis and prognosis via linear programming," *Oper. Res.*, vol. 43, no. 4, pp. 570–577, Aug. 1995.

[44] Z. Yang, L. Xu, Z. Cai, and Z. Xu, "Re-scale AdaBoost for attack detection in collaborative filtering recommender systems," *Knowl.-Based Syst.*, vol. 100, pp. 74–88, May 2016.

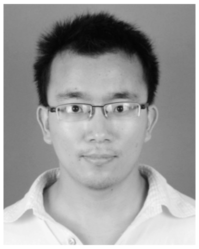
[45] L. Xu, S. Lin, J. Zeng, X. Liu, and Z. Xu. (2016). "Greedy criterion in orthogonal greedy learning." [Online]. Available: http://arxiv.org/abs/1604.05993

[46] C. Xu, S. Lin, J. Fang, and R. Li, "Prediction-based termination rule for greedy learning with massive data," *Statist. Sinica*, vol. 26, pp. 841–860, Jan. 2016.

[47] Q. Wu, Y. Ying, and D.-X. Zhou, "Multi-kernel regularized classifiers," *J. Complex.*, vol. 23, no. 1, pp. 108–134, Feb. 2007.

Lin Xu is currently pursuing the Ph.D. degree with the Institute for Information and System Sciences, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China.

His current research interests include neural networks, learning algorithms, and applications in computer vision.



Shaobo Lin received the Ph.D. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2014.

He is currently with Wenzhou University, Wenzhou, China. His current research interests include neural networks and learning theory.

Yao Wang received the Ph.D. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2014.

He is currently an Assistant Professor with the Department of Statistics, Xi'an Jiaotong University. His current research interests include statistical signal processing, high dimensional statistical inference, and computational biology.

Zongben Xu received the Ph.D. degree in mathematics from Xi'an Jiaotong University, Xi'an, China, in 1987.

He is currently the Academician with the Chinese Academy of Sciences, Beijing, China, and the Director of the Institute for Information and System Sciences with Xi'an Jiaotong University. His current research interests include applied mathematics, intelligent information processing, and data science and technology.

Dr. Xu is a member of the Chinese Academy of Sciences, Beijing. He was a recipient of the National Natural Science Award of China in 2007 and the CSIAM Su Buchin Applied Mathematics Prize in 2008. He delivered a 45-minute sectional talk at the International Congress of Mathematicians in 2010.