Boosted Histogram Transform for Regression

Anonymous Authors[1]

Abstract

In this paper, we propose a boosting algorithm for regression problems called *boosted histogram transform for regression* (BHTR) based on histogram transforms composed of random rotations, stretchings, and translations. From the theoretical perspective, we first prove fast convergence rates for BHTR under the assumption that the target function lies in the spaces $C^{0,\alpha}$. Moreover, if the target function resides in the subspace $C^{1,\alpha}$, for the first time we manage to explain the benefits of the boosting procedure, by establishing the upper bound of the convergence rate for the boosted regressor, i.e. BHTR, and the lower bound for base regressors, i.e. histogram transform regressors (HTR). In the experiments, compared with other state-of-the-art algorithms such as gradient boosted regression tree (GBRT), Breiman's forest, and kernel-based methods, our BHTR algorithm shows promising performance on both synthetic and real datasets.

1. Introduction

Over the past two decades, boosting has become one of the most successful algorithms in the machine learning community (Bühlmann & Yu, 2003). When the idea of iterative utilizations of *weak learners* from a certain function space to generate a strong one, which is called *boosting*, first came out in Schapire (1990); Freund (1995), it gains a lot of attention, and a wealth of literature has applied it on a large number of datasets.

During this period, many boosting algorithms with impressive performance have been proposed. Perhaps the first boosting algorithm goes back to the Adaboost for classification by Schapire & Freund (1995); Freund & Schapire (1997). Another important boosting algorithm for regression called *Gradient Boosted Regression Tree* (GBRT) is proposed by Friedman (2001). GBRT takes advantages of tree-based learners to capture complex data structure. More recently, inspired by the second order method originated from Friedman (2001), Chen & Guestrin (2016) came up with the *eXtreme Gradient Boosting* (XGBoost), which achieves excellent experimental performance without overfitting with the help of certain regularization terms.

Due to the great success of these boosting algorithms, a lot of attempts have been made to establish their theoretical foundations. First of all, theoretical margin guarantees of Adaboost have been well studied by Freund & Schapire (1997); Koltchinskii & Panchenko (2002). Furthermore, based on the view of gradient descent optimization, various versions of boosting algorithms have been shown to be consistent in different settings, see Friedman et al. (2000); Mannor et al. (2002); Bühlmann & Yu (2003); Lugosi & Vayatis (2004); Zhang (2004). Moreover, Blanchard et al. (2003) conducted a deeper investigation of the convergence of regularized boosting classifiers through the restriction on weights of the composite estimator. Finally, a different method of achieving consistency and convergence rate results is through early stopping rule, which is designed to prevent overfitting (Zhang & Yu, 2005), whereas Mease & Wyner (2008) argued that in practice additional iterations beyond the necessary number actually reduce the overfitting that has already occurred.

Unfortunately, none of the above-mentioned boosting works present a satisfactory explanation from the statistical optimization view (Mease & Wyner, 2008; Wyner et al., 2017). Nevertheless, some of the boosting variants with specific base learners are more easily accessible for statistical analysis. For example, Bühlmann & Yu (2003) derived an exponential bias-variance trade-off for linear regression to illustrate the almost resistance to overfitting for L_2-Boosting in a fixed design setting. Moreover, Park et al. (2009) and Lin et al. (2019) established the theoretical analysis of boosting methods using Nadaraya-Watson kernel estimates and kernel ridge regression estimates as base learners, respectively. However, these methods are of little practical value since they fail to capture the complex data dependencies in applications. In addition, they did not show the benefits of the boosting procedure from the theoretical perspective.

Under such background, this paper aims to establish a new

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

boosting algorithm which not only has satisfactory performance but also has solid theoretical foundations. To be specific, motivated by the random rotation ensemble algorithms (López-Rubio, 2013; Blaser & Fryzlewicz, 2016), we propose *boosted histogram transform for regression* (BHTR) which takes full advantages of the high effectiveness of the boosting procedure based on the histogram transforms: First of all, we generate a random histogram transform consisting of random rotations, stretchings, and translations. Then the input space is partitioned into non-overlapping cells corresponding to the unit bin in the transformed space. On those cells, we obtain base learners where piecewise constant functions are applied. Then the iterative process to fit residuals is started with the help of a sequence of random histogram transforms by a natural adaption of gradient descent boosting algorithm. Finally, by integrating the estimators generated by the above procedure, we obtain the boosted histogram transform regressor. It is worth mentioning that BHTR enjoys two advantages which can be stated as follows: First, the algorithm can be locally adaptive by applying stretching matrices associated with the variance of samples in each dimension. Second, our obtained regression function can be globally smooth thanks to the randomness of different base learners resulting from the random histogram partitions.

The contributions of this paper come from both theoretical and experimental perspectives: *(i)* In Section 3, we derive the theoretical results under the assumption that the target function resides in $C^{0,\alpha}$ and $C^{1,\alpha}$, respectively. By decomposing the error term into approximation error and sample error, we establish the fast convergence rates of BHTR in the space $C^{0,\alpha}$. Moreover, for the subspace $C^{1,\alpha}$ consisting of smoother functions, we are able to show that BHTR can attain the convergence rate $O(n^{-(2(1+\alpha))/(4(1+\alpha)+d)})$ whereas the lower bound of the convergence rates for HTR is merely of the order $O(n^{-2/(2+d)})$. As a result, when $d \geq 2(1+\alpha)/\alpha$, BHTR actually outperforms HTR, which confirms the benefits of the boosting procedure. *(ii)* In Section 4, several numerical experiments are designed to study the parameters including the bin width of the histogram transform, learning rate, and the iteration times of boosting, which coincide with the theoretical analysis of these parameters in the established convergence rates. Moreover, to validate the performance of BHTR, we conduct experiments of several algorithms including GBRT, Breiman's forest, and kernel-based methods on both synthetic and real datasets. Thanks to the randomness of the histogram transforms and the boosting procedure, our BHTR demonstrates both high accuracy and strong overfitting resistance.

2. Methodology

2.1. Notations

Regression is to predict the value of an unobserved output variable Y based on the observed input variable X, based on a dataset $D := \{(x_1, y_1), \ldots, (x_n, y_n)\}$ consisting of i.i.d. observations drawn from an unknown probability measure P on $\mathcal{X} \times \mathcal{Y}$. Throughout this paper, we assume that $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}$ are compact and non-empty.

For any fixed $R > 0$, we denote B_R as the centered hypercube of $\mathbb{R}^d$ with size $2R$, that is, $B_R := [-R, R]^d := \{x = (x_1, \ldots, x_d) \in \mathbb{R}^d : x_i \in [-R, R], i = 1, \ldots, d\}$, and for any $r \in (0, R/2)$, we write $B^+_{R,r} := [r, R-r]^d$. Recall that for $1 \leq p < \infty$, the L_p-norm of $x = (x_1, \ldots, x_d)$ is defined by $\|x\|_p := (|x_1|^p + \cdots + |x_d|^p)^{1/p}$, and the L_∞-norm is defined by $\|x\|_\infty := \max_{i=1,\ldots,d} |x_i|$.

Throughout this paper, we use the notation $a_n \lesssim b_n$ and $a_n \gtrsim b_n$ to denote that there exist positive constant c and c' such that $a_n \leq c b_n$ and $a_n \geq c' b_n$, for all $n \in \mathbb{N}$. Moreover, for any $x \in \mathbb{R}$, let $\lfloor x \rfloor$ denote the largest integer less than or equal to x. In the sequel, the following multi-index notations are used frequently. For any vector $x = (x_i)_{i=1}^d \in \mathbb{R}^d$, we write $\lfloor x \rfloor := (\lfloor x_i \rfloor)_{i=1}^d$, $x^{-1} := (x_i^{-1})_{i=1}^d$, $\log(x) := (\log x_i)_{i=1}^d$, $\overline{x} = \max_{i=1,\ldots,d} x_i$, and $\underline{x} = \min_{i=1,\ldots,d} x_i$.

2.2. Least Square Regression

It is legitimate to consider the least square loss $L : \mathcal{X} \times \mathcal{Y} \to [0, \infty)$ defined by $L(x, y, f(x)) := (y - f(x))^2$ for our target of regression. Then, for a measurable decision function $f : \mathcal{X} \to \mathcal{Y}$, the risk is defined by $\mathcal{R}_{L,\mathrm{P}}(f) := \int_{\mathcal{X}\times\mathcal{Y}} L(x, y, f(x))\, d\mathrm{P}(x, y)$ and the empirical risk is defined by $\mathcal{R}_{L,\mathrm{D}}(f) := \frac{1}{n}\sum_{i=1}^n L(x_i, y_i, f(x_i))$. The Bayes risk, which is the smallest possible risk with respect to P and L, is given by $\mathcal{R}^*_{L,\mathrm{P}} := \inf\{\mathcal{R}_{L,\mathrm{P}}(f) | f : \mathcal{X} \to \mathcal{Y} \text{ measurable}\}$.

In what follows, it is sufficient to consider predictors with values in $[-M, M]$. To this end, we introduce the concept of *clipping* for the decision function, see also Definition 2.22 in Steinwart & Christmann (2008). Let $\widehat{t}$ be the *clipped* value of $t \in \mathbb{R}$ at $\pm M$ defined by $-M$ if $t < -M$, t if $t \in [-M, M]$, and M if $t > M$. Then, a loss is called *clippable* at $M > 0$ if, for all $(y, t) \in \mathcal{Y} \times \mathbb{R}$, there holds $L(x, y, \widehat{t}) \leq L(x, y, t)$. According to Example 2.26 in Steinwart & Christmann (2008), the least square loss L is *clippable* at M with the risk reduced after clipping, i.e. $\mathcal{R}_{L,\mathrm{P}}(\widehat{f}) \leq \mathcal{R}_{L,\mathrm{P}}(f)$. Therefore, in the following, we only consider the clipped version $\widehat{f}_\mathrm{D}$ of the decision function as well as the risk $\mathcal{R}_{L,\mathrm{P}}(\widehat{f}_\mathrm{D})$.

2.3. Histogram transform for Regression

To give a clear description of one possible construction procedure of histogram transforms, we introduce a random vector (R, S, b) where each element represents the rotation matrix, stretching matrix and translation vector, respectively. To be specific,

- R denotes the rotation matrix which is a real-valued $d \times d$ orthogonal square matrix with unit determinant, that is
$$R^\top = R^{-1} \quad \text{and} \quad \det(R) = 1. \tag{1}$$
- S stands for the stretching matrix which is a positive real-valued $d \times d$ diagonal scaling matrix with diagonal elements $(s_i)_{i=1}^d$ that are certain random variables. Obviously, there holds
$$\det(S) = \prod_{i=1}^{d} s_i. \tag{2}$$
Moreover, we denote $s = (s_i)_{i=1}^d$, and the bin width vector defined on the input space is given by
$$h = s^{-1}. \tag{3}$$
- $b \in [0, 1]^d$ is a d dimensional vector named translation vector.

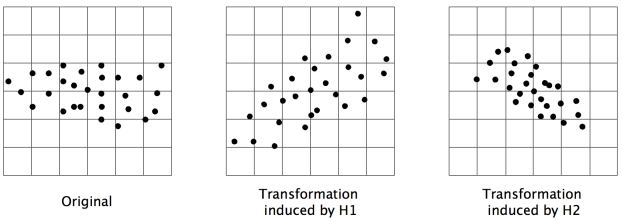


Figure 1. Two-dimensional examples of histogram transforms. The left subfigure is the original data and the other two subfigures are possible histogram transforms of the original sample space, with different rotating orientations and scales of stretching.

Based on the above notation, we define the histogram transform $H : \mathcal{X} \to \mathcal{X}$ by
$$H(x) := R \cdot S \cdot x + b. \tag{4}$$
It is important to note that there is no point to consider the bin width $h_0 \neq 1$ in the transformed space since the same effect can be achieved by scaling the transformation matrix H'. Therefore, let $\lfloor H(x) \rfloor$ be the transformed bin indices, then the transformed bin is given by
$$A'_H(x) := \{H(x') \mid \lfloor H(x') \rfloor = \lfloor H(x) \rfloor\}. \tag{5}$$
The corresponding histogram bin containing $x \in \mathcal{X}$ in the input space is
$$A_H(x) := \{x' \mid H(x') \in A'_H(x)\} \tag{6}$$
and we further denote all the bins induced by H as $\{A'_j\} = \{A_H(x) : x \in \mathcal{X}\}$ with the repetitive bin counted only once, and $\mathcal{I}_H$ as the index set for H such that for $j \in \mathcal{I}_H$, we have $A'_j \cap B_R \neq \emptyset$. As a result, the set
$$\pi_H := \{A_j\}_{j \in \mathcal{I}_H} := \{A'_j \cap B_R\}_{j \in \mathcal{I}_H}$$
forms a partition of partition of B_R. For the sake of convenience, we substitute A_0 for B_R^c and then
$$\pi'_H := \{A_j\}_{j \in \pi_H \cup \{0\}}$$
forms a partition of $\mathbb{R}^d$.

Here we describe a practical method for the construction of histogram transforms we are confined to in this study. Starting with a $d \times d$ square matrix M, consisting of d^2 independent univariate standard normal random variates, a Householder QR decomposition is applied to obtain a factorization of the form $M = R \cdot W$, with orthogonal matrix R and upper triangular matrix W with positive diagonal elements. The resulting matrix R is orthogonal by construction and can be shown to be uniformly distributed. Unfortunately, if R does not feature a positive determinant then it is not a proper rotation matrix according to definition (1). In this case, we can change the sign of the first column of R to construct a new rotation matrix R^+ that satisfies the condition (1).

We build a diagonal scaling matrix with the signs of the diagonal of S where the elements s_k are drawn from the well known Jeffreys prior, that is, $\log(s_i)$ follows the uniform distribution over certain interval of real numbers $[\log(\underline{s}_0), \log(\overline{s}_0)]$ for fixed constants $\underline{s}_0$ and $\overline{s}_0$ with $0 < \underline{s}_0 < \overline{s}_0 < \infty$. For simplicity and uniformity of notations, in the sequel, we denote $\overline{h}_0 = \underline{s}_0^{-1}$ and $\underline{h}_0 = \overline{s}_0^{-1}$, and then we say $h_i \in [\underline{h}_0, \overline{h}_0] = [\overline{s}_0^{-1}, \underline{s}_0^{-1}]$, $i = 1, \ldots, d$. Moreover, the translation vector b is drawn from the uniform distribution over the hypercube $[0, 1]^d$.

Given a histogram transform H, the set $\pi_H = \{A_j\}_{j \in \mathcal{I}_H}$ forms a partition of B_R. We consider the following function set $\mathcal{F}_H$ defined by
$$\mathcal{F}_H := \left\{ \sum_{j \in \mathcal{I}_H} c_j \mathbf{1}_{A_j} : c_j \in [-M, M] \right\}. \tag{7}$$
In order to constrain the complexity of $\mathcal{F}_H$, we penalize on the bin width $h := (h_i)_{i=1}^d$ of the partition π_H. Then the histogram transform regressor (HTR) can be produced by the regularized empirical risk minimization (RERM) over $\mathcal{F}_H$, i.e.
$$(f_{\mathrm{D}}, h^*) = \underset{f \in \mathcal{F}_H,\, h \in \mathbb{R}^d}{\arg\min} \; \Omega(h) + \mathcal{R}_{L,\mathrm{D}}(f), \tag{8}$$

where $\Omega(h) := \lambda \underline{h}_0^{-2d}$. It is worth pointing out that we adopt the isotropic penalty for each dimension rather than each elements $h_1, \ldots, h_d$ for simplicity of computation.

2.4. Boosted histogram transform with L_2 penalty

Boosting is the task of converting inaccurate weak learners into a single accurate predictor. To be specific, we define a restricted family of functions $\mathcal{F}$ be a set of base learners and a general boosting algorithm is combining a sequence of functions $\{f_t\}_{t=1}^T$ from $\mathcal{F}$ to minimize a certain empirical loss. Then the final predictor can be represented as

$$F = \sum_{t=1}^{T} w_t f_t,$$

where $w_t \geq 0$, $t = 1, \ldots, T$, are weights and $f_t \in \mathcal{F}$, $t = 1, \ldots, T$. From a functional gradient descent viewpoint in statistics (Friedman, 2001), boosting is reformulated as a stage-wise optimization problem with different loss functions. In this scenario, gradient boosting requires computing the negative *functional gradient* as the response

$$U_i = -\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\bigg|_{f(x_i)=\hat{f}(x_i)}$$

and select a particular model from the allowable class of functions at each boosting iteration to update the predictor.

In this work, we mainly focus on the boosting algorithm equipped with histogram transform regressors as base learners since they are weak predictors and enjoy computational efficiency. Before we proceed, we need to introduce the function space that we are most interested in to establish our learning theory. Assume that $\{H_t\}_{t=1}^T$ is an i.i.d. sequence of histogram transforms drawn from some probability measure P_H and $\mathcal{F}_t := \mathcal{F}_{H_t}$, $t = 1, \ldots, T$, are defined as in (7). Then we define the function space E by

$$E := \left\{ f : B_R \to \mathbb{R} \,\middle|\, f = \sum_{t=1}^{T} w_t f_t,\ f_t \in \mathcal{F}_t \right\}. \tag{9}$$

Moreover, for $f \in E$, we define

$$\|f\|_E := \inf\left\{ \sum_{t=1}^{T} |w_t|^2 \text{ with } f = \sum_{i=1}^{T} w_t f_t \right\}.$$

Then for any $f \in E$, by the Cauchy-Schwarz inequality, we immediately get

$$\|f\|_\infty \leq M \sum_{t=1}^{T} |w_t| \leq M(T\|f\|_E)^{1/2}.$$

In fact, $(E, \|\cdot\|_E)$ is a function space that consists of measurable and bounded functions.

As is mentioned above, boosting methods may be viewed as iterative methods for optimizing a convex empirical cost function. To simplify the theoretical analysis, following the approach of Blanchard et al. (2003), we ignore the dynamics of the optimization procedure and simply consider minimizers of an empirical cost function to establish the oracle inequalities, which leads to the following definition.

Definition 1 *Let E be the function space (9) and L be the least square loss. Given $\lambda_1 > 0$, $\lambda_2 > 0$, we call a learning method that assigns to every $D \in (\mathcal{X} \times \mathcal{Y})^n$ a function $f_{\mathrm{D},B} : \mathcal{X} \to \mathbb{R}$ such that*

$$(f_{\mathrm{D},B}, h^*) = \operatorname*{arg\,min}_{f \in E,\, h \in \mathbb{R}^d} \Omega_\lambda(f) + \mathcal{R}_{L,\mathrm{D}}(f) \tag{10}$$

a boosted histogram transform for regression (BHTR) algorithm with respect to E, where $\Omega_\lambda(f)$ is defined by

$$\Omega_\lambda(f) := \lambda_1 \Omega_1(f) + \lambda_2 \Omega_2(f) := \lambda_1 \|f\|_E + \lambda_2 \underline{h}_0^{-2d}.$$

The regularization term consists of two components. The first term is motivated by the fact the early boosting methods such as Adaboost may overfit in the presence of label noise. It helps control the degree of overfitting by the L_2-norm of the weights of the composite estimators and helps achieve the consistency and convergence results. The second term is added to control the bin width of the histogram transform, which has been discussed in subsection 2.3. In fact, it is equivalent to adding the L_p-norm of the base learners f_t. since they are piecewise constant functions on the cells with volume no more than $\overline{h}_0^d$.

To conduct the theoretical analysis, we also need the infinite sample version of Definition 1. To this end, we fix a distribution P on $\mathcal{X} \times \mathcal{Y}$ and let the function space E be as in (9). Then every $f_{\mathrm{P},B} \in E$ satisfying

$$\Omega_\lambda(f_{\mathrm{P},B}) + \mathcal{R}_{L,\mathrm{P}}(f_{\mathrm{P},B}) = \inf_{f \in E} \Omega_\lambda(f) + \mathcal{R}_{L,\mathrm{P}}(f)$$

is called an infinite sample version of BHTR with respect to E and L. Moreover, the approximation error function $A(\lambda)$ is defined by

$$A(\lambda) = \inf_{f \in E} \Omega_\lambda(f) + \mathcal{R}_{L,\mathrm{P}}(f) - \mathcal{R}^*_{L,\mathrm{P}}. \tag{11}$$

With all these preparations, we now present a general form of algorithm for BHTR in Algorithm 1. Indeed, the randomness of histogram transform provides an effective procedure for carrying out boosting. With the help of HTR, we repeat the least squares fitting of residuals. Moreover, we introduce the learning rate ρ to dampen the move on the gradient descent update, which is related to the regularization through shrinkage.

Algorithm 1 Boosted Histogram Transform for Regression

Input: Training data $D := \{(x_1, y_1), \ldots, (x_n, y_n)\}$;
Bandwidth parameters $\underline{h}_0$, $\overline{h}_0$;
Learning rate $\rho > 0$;
Initialization: $U_i = y_i$, $i = 1, \ldots, n$. $\widehat{F}_0(x) = 0$.
for $t = 1$ **to** T **do**
Generate random affine transform matrix $H_n^t = R_n \cdot S_n^t$;
Apply data independent splitting to the transformed sample space;
Apply constant functions to each cell, that is, fit residuals with function f_t such that

$$f_t = \arg\min_{f \in \mathcal{F}_t} \frac{1}{n} \sum_{i=1}^{n} L(U_i, f(x_i)),$$

where $\mathcal{F}_t$ is defined as in (7) for H_n^t.
Update: $\widehat{F}_t(x) = \widehat{F}_{t-1}(x) + \rho f_t(x)$.
Compute residuals $U_i = U_i - \widehat{F}_t(x_i)$, $i = 1, \ldots, n$.
end for
Output: Boosted histogram transform estimator for regression is $f_{\mathrm{D}}(x) = \widehat{F}_T(x)$.

3. Theoretical Results

In this paper, the theoretical analysis is built on Hölder space $C^{k,\alpha}$ consisting of (k, α)-Hölder continuous functions of different order of smoothness.

Definition 2 *Let $k \in \mathbb{N}_0 := \mathbb{N} \cup \{0\}$, $\alpha \in (0, 1]$, and $R > 0$. We say that a function $f : B_R \to \mathbb{R}$ is (k, α)-Hölder continuous, if there exists a finite constant $c_L > 0$ such that $\|\nabla^\ell f\| \le c_L$ for all $\ell \in \{1, \ldots, k\}$ and $\|\nabla^k f(x) - \nabla^k f(x')\| \le c_L \|x - x'\|^\alpha$ for all $x, x' \in B_R$. The set of such functions is denoted by $C^{k,\alpha}(B_R)$.*

From Definition 2 we see that the functions contained in the space $C^{k,\alpha}$ with larger k enjoy high level of smoothness. In particular, for the special case $k = 0$, the corresponding function space $C^{0,\alpha}(B_R)$ coincides with the commonly used α-Höder continuous function space $C^\alpha(B_R)$.

Throughout this paper, we make the following assumptions on the bin width h.

Assumption 1 *Let the bin width $h \in [\underline{h}_0, \overline{h}_0]$ be defined as in (3), assume that there exists some constant $c_0 \in (0, 1)$ such that $c_0 \overline{h}_0 \le \underline{h}_0 \le c_0^{-1} \overline{h}_0$. Moreover, if the bin width h depends on the sample size n, that is, $h_n \in [\underline{h}_{0,n}, \overline{h}_{0,n}]$, assume that there exist constants $c_{0,n} \in (0, 1)$ such that $c_{0,n} \overline{h}_{0,n} \le \underline{h}_{0,n} \le c_{0,n}^{-1} \overline{h}_{0,n}$.*

Assumption 1 requires that the upper and lower bounds of the bin width h are assumed to be of the same order. In other words, the extent of stretching in each dimension can not vary too much.

Finally, to leave out the boundary effect on the convergence rate, we denote $L_{\overline{h}_0}(x, y, t)$ as the least squares loss function restricted to $B^+_{R, \sqrt{d} \cdot \overline{h}_0}$, that is,

$$L_{\overline{h}_0}(x, y, t) := \mathbf{1}_{B^+_{R, \sqrt{d} \cdot \overline{h}_0}}(x) L(x, y, t), \tag{12}$$

where $L(x, y, t)$ is the least squares loss.

3.1. Convergence Rates for BHTR in $C^{0,\alpha}$

Theorem 1 *Let the histogram transform H_n be defined as in (4) with bin width h_n satisfying Assumption 1, and $f_{\mathrm{D},B}$ be defined in (10). Furthermore, suppose that the Bayes decision function $f^*_{L,\mathrm{P}} \in C^{0,\alpha}$. Moreover, let $\{\lambda_{1,n}\}$, $\{\lambda_{2,n}\}$ and $\{\overline{h}_{0,n}\}$ be chosen as*

$$\lambda_{1,n} := n^{-\frac{2\alpha}{(4-2\delta)\alpha + d}}, \quad \lambda_{2,n} := n^{-\frac{2(\alpha+d)}{(4-2\delta)\alpha+d}},$$
$$\overline{h}_{0,n} := n^{-\frac{1}{(4-2\delta)\alpha+d}},$$

where $\delta := (\underline{h}_{0,n}/c_d)^d$, then for all $\tau > 0$, we have

$$\mathcal{R}_{L,\mathrm{P}}(f_{\mathrm{D},B}) - \mathcal{R}^*_{L,\mathrm{P}} \lesssim n^{-\frac{2\alpha}{(4-2\delta)\alpha+d}},$$

holds with probability $\mathrm{P}^n \otimes \mathrm{P}_H$ at least $1 - 3e^{-\tau}$.

3.2. Convergence Rates for BHTR in $C^{1,\alpha}$

Theorem 2 *Let the histogram transform H_n be defined as in (4) with bin width h_n satisfying Assumption 1 and T_n be the number of iterations. Furthermore, let $f_{\mathrm{D},B}$ be defined in (10) and suppose that the Bayes decision function $f^*_{L,\mathrm{P}} \in C^{1,\alpha}$ and P_X is the uniform distribution. Moreover, let $L_{\overline{h}_0}(x, y, t)$ be the restricted least squares defined as in (12) and the sequences $\{T_n\}$, $\{\lambda_{1,n}\}$, $\{\lambda_{2,n}\}$, and $\{\overline{h}_{0,n}\}$ be chosen as*

$$\lambda_{1,n} := n^{-\frac{2}{2(1+\alpha)(2-\delta)+d}}, \quad \lambda_{2,n} := n^{-\frac{2(\alpha+d+1)}{2(1+\alpha)(2-\delta)+d}},$$
$$\overline{h}_{0,n} := n^{-\frac{1}{2(1+\alpha)(2-\delta)+d}}, \quad T_n := n^{\frac{2\alpha}{2(1+\alpha)(2-\delta)+d}},$$

where $\delta := (\underline{h}_{0,n}/c_d)^d$ with c_d depending only on d. Then, for all $\tau > 0$, the boosted histogram transform regressor satisfies

$$\mathcal{R}_{L_{\overline{h}_0},\mathrm{P}}(f_{\mathrm{D},B}) - \mathcal{R}^*_{L_{\overline{h}_0},\mathrm{P}} \lesssim n^{-\frac{2(1+\alpha)}{2(1+\alpha)(2-\delta)+d}} \tag{13}$$

with probability P^n not less than $1 - 4e^{-\tau}$ in expectation with respect to P_H.

Note that as $n \to \infty$, we have $\underline{h}_{0,n} \to 0$, and thus the upper bound for our BHTR attains asymptotically convergence rate which is slightly faster than

$$n^{-\frac{2(1+\alpha)}{4(1+\alpha)+d}}. \tag{14}$$

Moreover, the excess risk decreases as T_n increases at the beginning, and when T_n achieves a certain level, the algorithm achieves the optimal learning rate.

Theorem 3 *Let the histogram transform H_n be defined as in (4) with bin width h_n satisfying Assumption 1 with $\underline{h}_{0,n} \leq 1$, and T_n be the number of iterations. Furthermore, let the histogram transform regressor f_{D} be defined as in (8) and the regression model be defined by $Y := f(X) + \varepsilon$, where P_X is the uniform distribution over B_R and ε is independent of X such that $\mathrm{E}(\varepsilon|X) = 0$ and $\mathrm{Var}(\varepsilon|X) = \sigma^2 < \infty$. Moreover, assume that $f \in C^{1,\alpha}$ and there exists a constant $\underline{c}_f \in (0, \infty)$ such that $\|\nabla f\|_\infty \geq \underline{c}_f$. Then for all $n > N_1$, there holds*

$$\mathcal{R}_{L,\mathrm{P}}(f_{\mathrm{D}}) - R^*_{L,\mathrm{P}} \gtrsim n^{-2/(2+d)} \tag{15}$$

in expectation with respect to $\mathrm{P}^n \otimes \mathrm{P}_H$, where the constant N_1 is specified in the proof.

Note that for any $\alpha \in (0, 1]$, if $d \geq 2(1+\alpha)/\alpha$, then the upper bound of the convergence rate (13) for BHTR will be smaller than the lower bound (15) for HTR, which explains the benefits of the boosting procedure.

We mention that all the proofs in this paper can be found in supplementary material.

4. Numerical Experiments

4.1. Experimental Setup

We generate the random rotation matrix R in the manner described in Section 2.3 and apply the well-known Jeffreys prior for scale parameters (Jeffreys, 1946). To be specific, we draw $\log(s_i)$ from the uniform distribution over intervals $[\log(\underline{h}_0), \log(\overline{s}_0)]$. Recall that $h = s^{-1}$ stands for the bin width vector measured in the input space, we choose $\underline{s}_0$ and $\overline{s}_0$, recommended by (López-Rubio, 2013), as $\widehat{h} = 3.5\sigma n^{-1/(2+d)}$, where $\sigma := \sqrt{\mathrm{trace}(V)/d}$ is the standard deviation defined by $V := \frac{1}{n-1}\sum_{i=1}^n (x_i - \overline{x})(x_i - \overline{x})^\top$ and $\overline{x} := \frac{1}{n}\sum_{i=1}^n x_i$. Then we can transform the bin width vector to obtain this scale parameter $\widehat{s} = (\widehat{h})^{-1} = (3.5\sigma)^{-1} n^{\frac{1}{2+d}}$, which can be further refined as

$$\log(\underline{s}_0) := s_{\min} + \log(\widehat{s}), \quad \log(\overline{s}_0) := s_{\max} + \log(\widehat{s}),$$

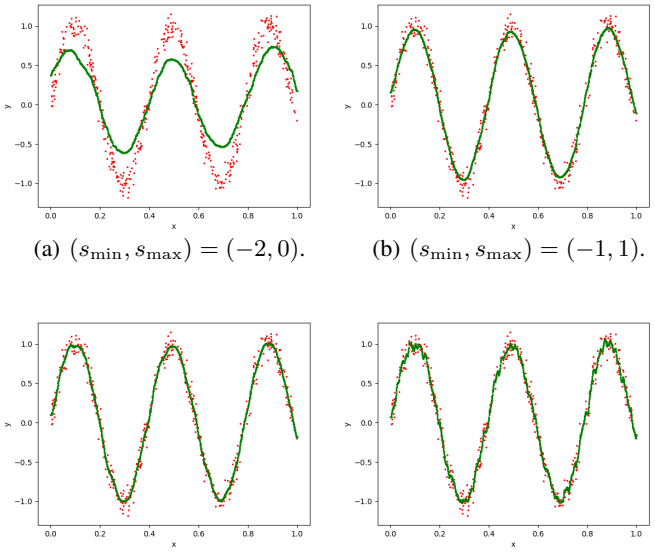
where $s_{\min} < s_{\max}$ are tunable parameters. Finally, to measure the performance for regression estimators, we adopt the mean squared error (*MSE*) defined by $MSE(\widehat{f}) = \frac{1}{n}\sum_{j=1}^n (y_j - \widehat{f}(x_j))^2$ over test set $\{(x_j, y_j)\}_{j=1}^n$.

4.2. Parameter Analysis for Histogram Transforms

In this subsection, we mainly conduct experiments dealing with the parameters of histogram transform for our BHTR algorithm, namely the lower and upper scale parameters $s_{\min}, s_{\max} \in \mathbb{R}$. To this end, we consider the following model:

$$Y = \sin(16X) + \varepsilon, \tag{16}$$

where $X \sim \mathrm{Unif}[0, 1]$ and $\varepsilon \sim \mathcal{N}(0, 0.1^2)$. Recall that the scale parameters $s_{\min}$ and $s_{\max}$ of the stretching matrix S control the size of histogram bins. For the regions with complex data structure, smaller bins are required while those with simple structure calls for larger bins. A narrower range of bin sizes are accommodated to cope with the varying scales while to preserve an homogeneous structure. We perform experiments with $n = 500$ training data and then predict 2000 test observations with four pairs of scale parameter $(s_{\min}, s_{\max}) \in \{(-2, 0), (-1, 1), (0, 2), (1, 3)\}$. In addition, we select $\rho = 0.01$ and $T = 500$. The results are shown in Figure 2.



(a) $(s_{\min}, s_{\max}) = (-2, 0)$. (b) $(s_{\min}, s_{\max}) = (-1, 1)$.

(c) $(s_{\min}, s_{\max}) = (0, 2)$. (d) $(s_{\min}, s_{\max}) = (1, 3)$.

Figure 2. Red points represent the training sample and green ones denote the predictive values on the test sample.

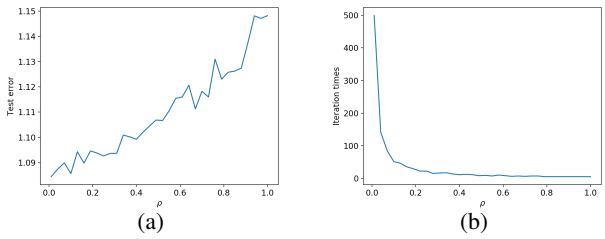
As we can see, lower values of these parameters leads to a coarser approximation for the underlying Bayes decision function, which results in the loss of precision. From Figure 2(a) we can see that the regressor is underfitting when the bin width is too large. On the contrary, with the bin width being too small, there are few samples lying in most of the histogram bins and thus resulting in overfitting, as is shown in Figure 2(d). Therefore, it is of great importance to properly choose the values of $s_{\min}$ and $s_{\max}$, where we the grid search procedure can be adopted.

4.3. Learning Rate

From the theory of boosting, it is well-known that weak learners should be underfitting, then the bias and variance

will be decreased and increased accordingly during the iterations. For an estimator with high-level underfitting, more boosting iterations are needed in order to achieve a competitive performance compared with other efficient learning algorithms. Therefore, if HTRs are used as base learners, then we should select a relatively small learning rate ρ in Algorithm 1. In this subsection, we conduct simulations to verify this argument and show the relationship between the generalization ability and the learning rate in Algorithm 1.

In the simulation, we choose the pair of scale parameter $(s_{\min}, s_{\max})$ to be $(-1, 1)$ and the learning rate ρ over the set $\{0.01 + 0.03k, k = 0, \ldots, 33\}$. Then we perform experiments with 350 training data and 150 validation data for the model (16) with ε independently drawn from the Gaussian distribution $\mathcal{N}(0, 1)$. For each learning rate parameter, we run BHTR with training set until T reaches 500. We repeat this procedure for 30 times, denote the corresponding estimator as $\{\widehat{F}_T^i(x)\}_{i=1}^{30}$, and compute the average of these estimators, that is, $\widehat{F}_T(x) = \frac{1}{30}\sum_{i=1}^{30} \widehat{F}_T^i(x)$. For each ρ, we estimate the error over the validation set $\{(x_j, y_j)\}_{j=1}^{150}$ by $\frac{1}{150}\sum_{j=1}^{150}(y_j - \widehat{F}_T(x_j))^2$. Then we record the optimal T with respect to the validation data and the corresponding test error on 2000 data. Figure 3 reports test errors and iteration numbers versus learning rate parameters.



(a) (b)

Figure 3. Figure (a) shows test errors of BHTR versus learning rate parameters, while Figure (b) shows the optimal iteration numbers.

From Figure 3 we see that the test error grows as the learning rate ρ increases from 0.01 to 1. Furthermore, when ρ is larger than some value, e.g. 0.2 in this simulation, then BHTR needs nearly the same number of steps to achieve the optimal test error. Moreover, Figure 3 shows that the more underfitted HTR is, the more boosting iterations are required. Therefore, it is reasonable to use HTR to build weak learners for boosting. And in this case, a smaller learning rate ρ is necessary to achieve better experimental performance.

4.4. Behavior of BHTR

In this subsection, we give a more comprehensive understanding of the behavior of BHTR. Since BHTR focuses on fixed learning rate parameter and varying iteration times, we perform the following experiment to study how the test error would behave as a function of iteration times.

In this simulation, The learning rate ρ is picked from the set $\{0.002, 0.01, 0.25, 0.5\}$ and other experimental setups are the same as those in Section 4.3. In Figure 4, test error versus iteration times for each learning rate is plotted. It can be apparently seen that the test error typically decreases until iteration times increases to certain value, and then it increases slowly, which reflects the trade-off between the approximation error and the sample error of the theoretical results in Section 3.

Notice that a too small ρ are likely to make the test error converge too slowly and bring about the additional burden of computation. Therefore, it is necessary to select an appropriate learning rate. Furthermore, Figure 4 shows a stable relation between the generalization performance and iteration numbers for some small ρ, e.g. $\rho < 0.05$ in this experiment. This tells us that overfitting does not seem to occur even though we run the boosting with plentiful of iterations. In addition, it is easily seen that the number T of iterations at which the test error achieves the minimal value would increase as ρ increases. Last but not least, lower test error implies better performance of BHTR than HTR in terms of accuracy for a wide range of ρ.

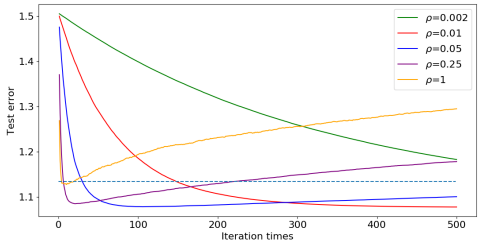


Figure 4. Test errors versus the number of iterations for different learning rates. The horizontal line indicates the test error for HTR.

4.5. Synthetic and Real Data Analysis

In this section, our experiments are carried out on three benchmark synthetic datasets and eight real datasets.

For our BHTR, We first scale each feature individually to the range $[0, 1]$ on the training set, and then impose a gird of size 4 on learning rate $\rho \in \{0.2, 0.1, 0.05, 0.01\}$, a grid of size 3 on $s_{\min} \in \{-4, -3, -2\}$ and a grid of size 2 on $s_{\max} - s_{\min} \in \{1, 2\}$. For each element from the Cartesian product of these grids, we run BHTR with iteration times $T = 3000$. The optimal iteration times $t \leq T$ and optimal parameters ρ, $s_{\min}$ and $s_{\max}$ are chosen by 5-fold cross-validation. The comparison are conducted among

- Gradient Boosting Regression Tree (GBRT): proposed by Friedman (2001). We utilize the parkage `sklearn`

in `python` with iteration times `n_estimators = 3000` and other parameters being default.

- Random Forest (RF): proposed by Breiman (2001). The `sklearn` package in `python` is applied with `n_estimators = 100` and other parameters being default.

- Boosted Kernel Ridge Regression (BKRR): proposed by Lin et al. (2019). They combine L_2-Boosting with the kernel ridge regression. The regularization parameter λ is chosen from grid `np.logspace(0, 4, 5)` and the bandwidth σ is chosen from grid `np.logspace(-1, 1, 3)`. The iteration times are set to be 3000 and the hyper-parameters is chosen by 5-fold cross-validation.

Comparisons on Synthetic Datasets

We choose Friedman's benchmark functions (Friedman, 1991), which are widely and frequently employed models in regression problems (Smola & Schölkopf, 2004; Brown et al., 2005; Feng et al., 2015), listed as follows:

1. $f_1(x) = \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5$;
2. $f_2(x) = \sqrt{(x_1)^2 + (x_2 x_3 - 1/(x_2 x_4))^2}$;
3. $f_3(x) = \arctan(1/x_1(x_2 x_3 - 1/(x_2 x_4)))$.

For f_1 we have $x = (x_1, \ldots, x_{10})$, where $x_j \sim \text{Unif}[0,1]$, $j = 1, \ldots, 5$, and x_j, $j = 6, \ldots, 10$, are noise variables. For f_2 and f_3, we have $x = (x_1, \ldots, x_4)$, where $x_1 \sim \text{Unif}[0, 100]$, $x_2 \sim \text{Unif}[40\pi, 560\pi]$, $x_3 \sim \text{Unif}[0,1]$, and $x_4 \sim \text{Unif}[1, 11]$. Moreover, we assume that the noise added to the function f is drawn from the standard normal distribution. For each function, 1000 observations are generated for training and another 1000 are for testing. The cross-validation procedure is adopted for hyper-parameter selection.

Table 1. Average *MSE* over synthetic datasets

Dataset	BHTR	GBRT	RF	BKRR
`Fried 1`	*3.55*	**2.01**	3.87	9.41
`Fried 2`	**258.81**	313.76	*310.08*	10448
`Fried 3`	*1.09*	1.43	1.10	**1.03**

* The best results are marked in **bold**, and the second best are marked in *italic*.

Table 1 shows that on the second dataset `Fried 2`, BHTR performs the best while on the remaining two datasets, it ranks the second.

Comparisons on Real Datasets

Eight real datasets are listed in Table 2. Further information can be found in the supplementary material.

Table 2. Description of real datasets

DATASET	N	d	DATASET	N	d
`ABA`	4177	8	`MPG`	392	7
`BOD`	252	14	`PYR`	74	27
`HOU`	506	13	`SPA`	3107	6
`MG`	1385	6	`TRI`	186	60

Table 3. Average *MSE* over real datasets

Dataset	BHTR	GBRT	RF	BKRR
`ABA`	**4.60**	5.73	*4.83*	6.43
`BOD`	$1.65e-05$	**9.51e-06**	*9.53e-06*	$3.56e-04$
`HOU`	13.03	**10.03**	*12.33*	46.39
`MG`	*0.016*	0.017	**0.015**	0.021
`MPG`	**6.98**	8.47	*7.67*	11.54
`PYR`	**0.0063**	0.0067	*0.0065*	0.010
`SPA`	*0.0123*	**0.0120**	0.0137	0.061
`TRI`	0.021	*0.019*	**0.016**	0.023

* The best results are marked in **bold**, and the second best are marked in *italic*.

Table 3 shows that on three datasets `ABA`, `MPG`, and `PYR`, our BHTR has the best accuracy and on another two datasets `MG` and `SPA`, BHTR ranks the second.

From the results in Tables 1 and 3 we come to the conclusion that our method shows promising performance compared to the efficient algorithms GBRT and RF.

We mention that to improve the performance, Friedman (2002) proposed the stochastic gradient boosting algorithm using subsampling to reduce the variance of the base learners. Therefore, in the future work, we believe that to further improve the performance of BHTR, the combination of BHTR with subsampling should be explored.

5. Conclusion

In the present paper, we propose the boosting algorithm called *boosted histogram transform regression* (BHTR) based on histogram transforms consisting of random rotations, stretchings, and translations. By conducting a theoretical analysis within the framework of regularized empirical risk minimization in learning theory, we are able to prove the fast convergence rates of BHTR when the target function $f^*_{L,\mathrm{P}}$ lies in the Hölder space $C^{0,\alpha}$. Moreover, when $f^*_{L,\mathrm{P}} \in C^{1,\alpha}$, it is the first time that we successfully derive the upper bound of convergence rates for the boosted predictor BHTR and the lower bound for the base predictor HTR, which further demonstrate the benefits of the boosting procedure. In the experiments, numerical simulations and real data comparisons with other state-of-the-art methods including GBRT, random forest, and kernel-based boosting algorithms are provided to support the theoretical results and justify the performance of BHTR.

References

Blanchard, G., Lugosi, G., and Vayatis, N. On the rate of convergence of regularized boosting classifiers. *The Journal of Machine Learning Research*, 4(Oct):861–894, 2003.

Blaser, R. and Fryzlewicz, P. Random rotation ensembles. *The Journal of Machine Learning Research*, 17(1):126–151, 2016.

Breiman, L. Random forests. *Machine Learning*, 45(1): 5–32, 2001.

Brown, G., Wyatt, J. L., and Tiňo, P. Managing diversity in regression ensembles. *The Journal of Machine Learning Research*, 6(Sep):1621–1650, 2005.

Bühlmann, P. and Yu, B. Boosting with the L2 loss: regression and classification. *Journal of the American Statistical Association*, 98(462):324–339, 2003.

Chen, T. and Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016.

Feng, Y., Huang, X., Shi, L., Yang, Y., and Suykens, J. Learning with the maximum correntropy criterion induced losses for regression. *The Journal of Machine Learning Research*, 16:993–1034, 2015.

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

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

Friedman, J., Hastie, T., and Tibshirani, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, 28(2):337–407, 2000.

Friedman, J. H. Multivariate adaptive regression splines. *The Annals of Statistics*, pp. 1–67, 1991.

Friedman, J. H. Greedy function approximation: a gradient boosting machine. *The Annals of statistics*, pp. 1189–1232, 2001.

Friedman, J. H. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

Jeffreys, H. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

Koltchinskii, V. and Panchenko, D. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

Lin, S.-B., Lei, Y., and Zhou, D.-X. Boosted kernel ridge regression: Optimal learning rates and early stopping. *The Journal of Machine Learning Research*, 20(46):1–36, 2019.

López-Rubio, E. A histogram transform for probabilitydensity function estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):644–656, 2013.

Lugosi, G. and Vayatis, N. On the Bayes-risk consistency of regularized boosting methods. *The Annals of Statistics*, 32(1):30–55, 2004.

Mannor, S., Meir, R., and Zhang, T. The consistency of greedy algorithms for classification. In *International Conference on Computational Learning Theory*, pp. 319–333. Springer, 2002.

Mease, D. and Wyner, A. Evidence contrary to the statistical view of boosting. *The Journal of Machine Learning Research*, 9(Feb):131–156, 2008.

Park, B., Lee, Y., and Ha, S. L2 boosting in kernel regression. *Bernoulli*, 15(3):599–613, 2009.

Schapire, R. and Freund, Y. A decision-theoretic generalization of on-line learning and an application to boosting. In *Second European Conference on Computational Learning Theory*, pp. 23–37, 1995.

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

Smola, A. J. and Schölkopf, B. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

Steinwart, I. and Christmann, A. *Support Vector Machines*. Information Science and Statistics. Springer, New York, 2008.

Wyner, A. J., Olson, M., Bleich, J., and Mease, D. Explaining the success of adaboost and random forests as interpolating classifiers. *The Journal of Machine Learning Research*, 18(1):1558–1590, 2017.

Zhang, T. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32(1):56–85, 2004.

Zhang, T. and Yu, B. Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, 33(4): 1538–1579, 2005.