GBHT: Gradient Boosting Histogram Transform for Density Estimation

Anonymous Authors[1]

Abstract

Boosting is a general method for improving the accuracy of a learning algorithm. Despite its successful implementations in supervised learning, the effectiveness of boosting in unsupervised learning tasks remains under-exploited. In this paper, we propose a density estimation algorithm called *Gradient Boosting Histogram Transform* (GBHT), where we adopt the *Negative Log Likelihood* as the loss function to make the boosting procedure available for the unsupervised tasks. From a learning theory viewpoint, we first prove fast convergence rates for GBHT with smoothness assumption that the underlying density function lies in the space $C^{0,\alpha}$. Then when the target density function lies in spaces $C^{1,\alpha}$, we present an upper bound for GBHT which is smaller than the lower bound of its corresponding base learner, in the sense of convergence rates. To the best of our knowledge, we make the first attempt to theoretically explain why boosting can enhance the performance of its base learners for density estimation problems. In experiments, we not only conduct performance comparisons with the widely used KDE, but also apply GBHT to anomaly detection to showcase a further application of GBHT.

1. Introduction

Regarded as one of the most important tasks in unsupervised learning, density estimation aims at inferring the true distribution of targeted unknown variables through limited samples. While basic statistical analysis can be directly carried out on density functions (Scott, 2015), density estimation is further regarded as an imperative cornerstone to more sophisticated tasks, such as anomaly detection (Nachman & Shih, 2020; Zhang et al., 2018; Amarbayasgalan et al., 2018) and clustering (Chen et al., 2020; Parmar et al., 2019; Ghaffari et al., 2019; Jang & Jiang, 2019).

On the other hand, as one of the most successful algorithms over two decades (Bühlmann & Yu, 2003), boosting attracts more and more attention in researches on machine learning (Mathiasen et al., 2019; Cortes et al., 2019; Parnell et al., 2020; Duan et al., 2020; Cai et al., 2020; Suggala et al., 2020) However, when boosting method shows its power and strength in the field of supervised learning, few studies focus on unsupervised learning, especially on the density estimation problem. Furthermore, previous attempts (Ridgeway, 2002; Rosset & Segal, 2003) focus more on methodology study instead of statistical theories. To the best of our knowledge, there remains little understood of the theoretical advantage of boosting over its base learners from the statistical learning point of view.

Under such background, by combing the boosting framework (Rosset & Segal, 2003) with the random histogram transforms (López-Rubio, 2013; Blaser & Fryzlewicz, 2016), this paper aims to establish a new boosting algorithm called *Gradient Boosting Histogram Transform* (*GBHT*) for density estimation, which not only has satisfactory performance but also has solid theoretical foundations. To be specific, we adopt the *Negative Log Likelihood* loss, which makes the boosting method, typically used in supervised learning tasks, available for density estimation which is an unsupervised problem. Moreover, through complete learning theory analysis, we for the first time provide theoretical supports to the benefit of the boosting procedure in the density estimation problem. GBHT starts with generating 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 is started with adding a sequence of random histogram transforms for minimizing empirical negative log-likelihood loss by a natural adaption of gradient descent boosting algorithm. Finally, after the iterative process, we inversely transform the partitioned space to the original and obtain the GBHT density estimator.

The contributions of this paper come from the model, theoretical, and experimental perspectives:

(i) While majority studies of boosting focus on supervised

[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.

learning, we exploit boosting to improve the accuracy in density estimation by taking an unsupervised loss function.

(ii) From a learning theory point of view, we prove the fast convergence rates of GBHT with assumptions that the underlying density functions lie in the Hölder space $C^{0,\alpha}$.

(iii) To our best knowledge, we are the first to explain the strength of boosting density estimation from the theoretical point of view. To be specific, in the space $C^{1,\alpha}$, we show that HT density estimator obtains lower bound as $O(n^{-2/2+d})$, which turns out to be greater than the upper bound for GBHT $O(n^{-2(1+\alpha)/4(1+\alpha)+d})$.

(iv) In experiments, we validate the performances of our algorithms through parameter analysis and real data comparisons. Moreover, we apply GBHT as part of a density-based anomaly detection algorithm, where the results show the promising compatibility of our GBHT.

2. Related Works

Density Estimation. The best-known and most traditional density estimation methods are histogram density estimation (HDE) and kernel density estimation (KDE), while the former one is often criticized for its lack of smoothness and the latter one is found weak against outlier and local adaptivity. In order to solve these problems, partition-based methods (Klemelä, 2009; Liu & Wong, 2014; López-Rubio, 2013), e.g. decision tree-based algorithms (Ram & Gray, 2011; Criminisi et al., 2011; Criminisi & Shotton, 2013) have been taken into consideration. However, partition-based algorithms inherently suffer from boundary discontinuity, i.e. the density estimation of adjacent partition cells may not correspond on their shared boundary. In this paper, inspired by histogram density estimation, we aim at solving the boundary discontinuity by aggregating random histogram transform density estimators with the help of boosting.

Boosting. Boosting is a widely used learning technique in machine learning. It boosts the performance of a base learner by combining multiple weak learners. In boosting, the weak learner in each iteration learns from the distance between truth and the estimated one, e.g. residuals in regression and wrong labels in classification. Based on these ideas, various boosting algorithms such as AdaBoost (Schapire & Freund, 1995; Freund & Schapire, 1997), GBDT and GBRT (Friedman, 2001), and XgBoost (Chen & Guestrin, 2016) become popular.

Despite its great success in supervised learning, very few studies focus on exploiting the effectiveness of boosting in unsupervised learning, especially in density estimation problems. For instance, (Rosset & Segal, 2003) considers boosting as a gradient descent search method, and transform density estimation problem into a supervised learning problem by rationally adjusting the loss function. (Ridgeway, 2002) brings EM algorithm in to conduct boosting density estimation. These authors suggest that more researches can be done with boosting for density estimation problems, since present researches about boosting density estimation focus mainly on methodology following the derivation process of gradient descent and none of the above-mentioned boosting works present a satisfactory explanation from the statistical optimization view.

This paper aims at filling the blank in studies of boosting in unsupervised learning, and at providing sound theoretical analysis to explain why boosting can enhance the performance of its base learners for density estimation problems.

3. Methodology

3.1. Notations

Throughout this paper, we assume that $\mathcal{X} \subset \mathbb{R}^d$ is compact and non-empty. For any fixed $r > 0$, we denote B_r as the centered hyper-cube of $\mathbb{R}^d$ with size $2r$, that is, $B_r := [-r, r]^d := \{x = (x_1, \dots, x_d) \in \mathbb{R}^d : x_i \in [-r, r], i = 1, \dots, d\}$, and for any $r' \in (0, r)$, we write $B^+_{r,r'} := [-r + r', r - r']^d$. Recall that for $1 \leq p < \infty$, the L_p-norm of $x = (x_1, \dots, x_d)$ is defined by $\|x\|_p := (|x_1|^p + \dots + |x_d|^p)^{1/p}$, and the L_∞-norm is defined by $\|x\|_\infty := \max_{i=1,\dots,d} |x_i|$.

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 $\lfloor x \rfloor := (\lfloor x_i \rfloor)_{i=1}^d$, $x^{-1} := (x_i^{-1})_{i=1}^d$, $\log(x) := (\log x_i)_{i=1}^d$, $\overline{x} = \max_{i=1,\dots,d} x_i$, and $\underline{x} = \min_{i=1,\dots,d} x_i$.

3.2. Negative Log Likelihood Loss

Let f be the underlying density function of an unknown probability measure P on $\mathcal{X}$. Based on a dataset $D := \{x_1 \dots, x_n\}$ consisting of i.i.d. observations drawn from P, our goal in the density estimation is to construct a measurable function $\hat{f} : \mathcal{X} \to [0, \infty)$ satisfying $\int_{\mathcal{X}} \hat{f}(x)\, dx = 1$ to approximate f properly. To evaluate the quality of $\hat{f}$, we use the *Negative Log Likelihood* loss $L : \mathcal{X} \times [0, \infty) \to \mathbb{R}$ defined by

$$L(x, \hat{f}) := -\log \hat{f}(x). \qquad (1)$$

Then the risk is defined by $\mathcal{R}_{L,\mathrm{P}}(\hat{f}) := \int_{\mathcal{X}} L(x, \hat{f})\, d\mathrm{P}(x)$ and the empirical risk is defined by $\mathcal{R}_{L,\mathrm{D}}(\hat{f}) := \frac{1}{n} \sum_{i=1}^n L(x_i, \hat{f}(x_i))$. The Bayes risk, which is the smallest possible risk with respect to P and L, is given by $\mathcal{R}^*_{L,\mathrm{P}} := \inf\{\mathcal{R}_{L,\mathrm{P}}(\hat{f}) | \hat{f} : \mathcal{X} \to [0, \infty)$ measurable and $\int_{\mathcal{X}} \hat{f}(x)\, dx = 1\}$. It is easy to verify that the $\hat{f}(x)$ that maximizes $\mathcal{R}_{L,\mathrm{P}}$ is indeed the true density. Therefore, it is reasonable to consider the framework that using gradient-based functional optimization algorithms to generate density estimators.

3.3. Histogram Transform (HT) for Density Estimation

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 $R^\top = R^{-1}$ and $\det(R) = 1$; 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, we have $\det(S) = \prod_{i=1}^d s_i$. Moreover, we denote $s = (s_i)_{i=1}^d$, and the bin width vector defined on the input space is given by $h = s^{-1}$; $b \in [0,1]^d$ is a d dimensional vector named translation vector. Then we define the histogram transform $H : \mathcal{X} \to \mathcal{X}$ by

$$H(x) := R \cdot S \cdot x + b. \tag{2}$$

Figure 1 illustrates 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.

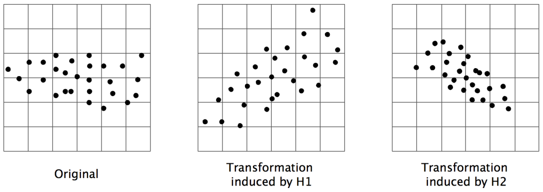


Figure 1. Two possible histogram transforms in 2-D.

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 $A'_H(x) := \{H(x') \mid \lfloor H(x') \rfloor = \lfloor H(x) \rfloor\}$ and the corresponding histogram bin containing $x \in \mathcal{X}$ in the input space is $A_H(x) := \{x' \mid H(x') \in A'_H(x)\}$. 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 B_r. 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 $h_i \in [\underline{h}_0, \overline{h}_0] = [\overline{s}_0^{-1}, \underline{s}_0^{-1}]$, $i = 1, \ldots, d$.

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

$$\mathcal{F}_H := \left\{ \sum_{j \in \mathcal{I}_H} c_j \mathbf{1}_{A_j} \,\middle|\, c_j \geq 0, \sum_{j \in \mathcal{I}_H} c_j \mu(A_j) = 1 \right\}. \tag{3}$$

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 (HT) density estimator can be produced by the regularized empirical risk minimization (RERM) over $\mathcal{F}_H$, i.e.

$$(f_{\mathrm{D},H}, h^*) = \operatorname*{arg\,min}_{f \in \mathcal{F}_H,\, h \in \mathbb{R}^d} \Omega(h) + \mathcal{R}_{L,\mathrm{D}}(f), \tag{4}$$

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.

3.4. Gradient Boosting Histogram Transform (GBHT) for Density Estimation

In this work, we mainly focus on the boosting algorithm equipped with histogram transform density estimators 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 (3). Then we define the function space E by

$$E := \left\{ f : B_r \to \mathbb{R} \,\middle|\, f = \sum_{t=1}^T w_t f_t,\ f_t \in \mathcal{F}_t \text{ s.t. } \sum_{t=1}^T w_i = 1 \right\}. \tag{5}$$

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 (5) and L be the negative log-negative loss. Given $\lambda > 0$, we call a learning method that assigns to every $D \in (\mathcal{X} \times \mathcal{Y})^n$ a function $f_{\mathrm{D},\lambda} : \mathcal{X} \to \mathbb{R}$ such that*

$$(f_{\mathrm{D},\lambda}, h^*) = \operatorname*{arg\,min}_{f \in E,\, h \in \mathbb{R}^d} \Omega(h) + \mathcal{R}_{L,\mathrm{D}}(f) \tag{6}$$

a gradient boosting histogram transform (GBHT) algorithm for density estimation with respect to E, where $\Omega(h) := \lambda \underline{h}_0^{-2d}$.

The regularization term is added to control the bin width of the histogram transform, which has been discussed in Section 3.3. 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 $\overline{h}_0^d$.

With all these preparations, we now present the gradient boosting algorithm GBHT to solve the optimization problem (6) in Algorithm 1.

Algorithm 1 Gradient Boosting Histogram Transform (GBHT) for Density Estimation

```
Input: Training data $D := \{x_1, \dots, x_n\}$;
       Bandwidth parameters $\underline{h}_0$, $\overline{h}_0$;
       Number of iterations $T$.
Initialization: $F_0$ is set to be uniformly distributed on cells $A_j \in \pi_H$ satisfying $A_j \cap D \neq \emptyset$.
for $t = 1$ to $T$ do
    Set the sample weight $\omega_{t,i} = 1/F_{t-1}(x_i)$;
    For random histogram transformation $H_t$ (2):
    Find $f_t = \arg\max_{f \in \mathcal{F}_t} \sum_{i=1}^n \omega_{t,i} f(x_i)$;
    Find $\alpha_t := \arg\min_\alpha \sum_{i=1}^n - \log\big((1-\alpha)F_{t-1}(x_i) + \alpha f_t(x_i)\big)$;
    Update $F_t = (1-\alpha_t)F_{t-1} + \alpha_t f_t$;
end for
return $F_T$.
```

The algorithm proceeds iteratively, that is, for $t = 1, \dots, T$, $F_t(x_i) = (1-\alpha_t)F_{t-1} + \alpha_t f_t$, where F_t denotes the density estimator after t iterations, $f_t \in \mathcal{F}_t$ denotes the t-th base learner, and $\alpha_t \in (0,1)$. Obviously we have $F_t = w_{t,0}F_0 + \sum_{j=1}^t w_{t,j} f_j$, where $w_{t,j} = (1-\alpha_t)\cdots(1-\alpha_{j+1})\alpha_j$ for $j = 1, \dots, t$, and $w_{t,0} = \prod_{j=1}^t (1-\alpha_j)$. By initiating $F_0 \in \mathcal{F}_0 := \mathcal{F}_H$, we have $F_t \in E$. Then we aim to search the base learner f_t under partition H_t and step size α_t to result in F_t with lower empirical risk $\mathcal{R}_{L,\mathrm{D}}(F_t)$ in each iteration. In the t-th iteration, for every $\alpha_t \in (0,1)$, the minimization of $\mathcal{R}_{L,\mathrm{D}}(F_t)$ equals to the minimization of $\sum_{i=1}^n - \log(F_{t-1}(x_i) + \varepsilon_t f_t(x_i))$, where $\varepsilon_t = \alpha_t/(1-\alpha_t)$. Using Taylor expansion, we get

$$\begin{aligned}&\sum_i - \log(F_{t-1}(x_i) + \varepsilon_t f_t(x_i))\\&\quad = \sum_i - \log(F_{t-1}(x_i)) - \varepsilon_t \cdot \omega_{t,i} f_t(x_i) + O(\varepsilon_t^2),\end{aligned}$$

where $\omega_{t,i} := 1/F_{t-1}(x_i)$. For sufficiently small ε_t (or α_t), we can ignore the higher order term and find the maximum gradient $\max_{f_t \in \mathcal{F}_t} \sum_{i=1}^n \omega_{t,i} f_t(x_i)$. Then we determine the step size α_t by line search, which ensures that the updated F_t remains to be a probability distribution.

It is worth mentioning that GBHT enjoys two advantages. First, the algorithm can be locally adaptive by applying random rotations, stretchings, and translations to the original input data. Regular density estimators such as KDE adopt uniform bandwidth, regardless of the fact that the local structures of real-world data usually vary from area to area. On the contrary, it is well known that boosting algorithms take local data structures into consideration by updating its vulnerable part in each iteration, and the adopted histogram transform catches exactly various local features of the input data. Thus, good combinations of random weak learners and the boosting procedure can lead to great local adaptivity. Second, the boosting procedure brings smoothness to histogram-based density estimators, thanks to the randomness of base learners. Through iteration, GBHT adds more information obtained by the base learners into the boosting estimator, and it turns out to be the weighted average of all random base learners with different partition boundaries. As a result, it can be more smooth than regular histogram density estimators, which will also be theoretically verified in Section 4 and experimentally validated by numerical simulations in Section 5.3.

4. Theoretical Results

Our theoretical analysis is built on the fundamental assumption on the smoothness of the underlying density function. Recall that a function $f : \mathcal{X} \to \mathbb{R}$ is (k, α)-Hölder continuous, $\alpha \in (0, 1]$, $k \in \mathbb{N}_0$, if there exists a constant $c_L \in (0, \infty)$ such that

$$\|\nabla^\ell f\| \le c_L \text{ for all } \ell \in \{1, \dots, k\} \text{ and} \tag{7}$$

$$\|\nabla^k f(x) - \nabla^k f(x')\| \le c_L \|x - x'\|^\alpha \tag{8}$$

for all $x, x' \in B_r$. The set of such functions is denoted by $C^{k,\alpha}(B_r)$. Note that the functions contained in the space $C^{k,\alpha}$ with larger k enjoy a higher level of smoothness. Throughout this paper, we make the following assumptions on the bin width h.

Assumption 1 *Let the bin width $h \in [\underline{h}_0, \overline{h}_0]$ and 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, $h_n \in [\underline{h}_{0,n}, \overline{h}_{0,n}]$, we still have $c_0 \overline{h}_{0,n} \le \underline{h}_{0,n} \le c_0^{-1} \overline{h}_{0,n}$.*

Assumption 1 indicates that the upper and lower bounds of the bin width h are of the same order. In other words, we assume that under a certain partition, the extent of stretching in each dimension cannot vary too much.

Furthermore, to remove the boundary effect on the convergence rate, we denote $L_{\overline{h}_0}(x, t)$ as the negative log loss function restricted to $B^+_{R,\sqrt{d}\cdot\overline{h}_0}$, that is,

$$L_{\overline{h}_0}(x, t) := \mathbf{1}_{B^+_{R,\sqrt{d}\cdot\overline{h}_0}}(x) L(x, t), \tag{9}$$

where $L(x, t)$ is the negative log loss.

4.1. Convergence Rates for GBHT in $C^{0,\alpha}$

Theorem 1 *Let $f_{\mathrm{D},\lambda}$ be as in (6) and the density function $f \in C^{0,\alpha}(B_r)$. Then for all $\tau > 0$ and for any $\delta \in (0,1)$, there exists a constant N_0 such that for all $n \geq N_0$, there holds*

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

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

Theorem 1 presents the fast convergence rates of the GBHT density estimator in the sense of "with high probability", which is a stronger claim than the convergence results "in expectation". Moreover, convergence rates, a finite sample property of GBHT, also indicate the consistency of $\mathcal{R}_{L,\mathrm{P}}(f_{\mathrm{D},\lambda})$ when $n \to \infty$.

4.2. Convergence Rates for GBHT in $C^{1,\alpha}$

Theorem 2 *Let $f_{\mathrm{D},\lambda}$ be as in (6) and the density function $f \in C^{1,\alpha}(B_r)$. Moreover, let $L_{\overline{h}_0}(x,t)$ be the restricted negative log loss as in (9). Then for all $\tau > 0$ and $\delta \in (0,1)$, there exists a constant N_1 such that for all $n \geq N_1$, by choosing $T_n \gtrsim n^{2\alpha/(2(1+\alpha)(2-\delta)+d)}$, there holds*

$$\mathcal{R}_{L_{\overline{h}_0},\mathrm{P}}(f_{\mathrm{D},\lambda}) - \mathcal{R}^*_{L_{\overline{h}_0},\mathrm{P}} \lesssim n^{-\frac{2(1+\alpha)}{2(1+\alpha)(2-\delta)+d}} \qquad (10)$$

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

In Theorem 2, the excess risk decreases as T_n grows at first, and when T_n achieves a certain level, the algorithm achieves the best convergence rate. Moreover, comparing with Theorem 1, when the underlying density function turns more smooth, GBHT achieves a better convergence rate with $f \in C^{1,\alpha}(B_r)$ than that with $f \in C^{0,\alpha}(B_r)$, where a relatively large T_n helps the density estimator to achieve asymptotic smoothness.

4.3. Lower Bound for HT Density Estimation in $C^{1,\alpha}$

Theorem 3 *Let $f_{\mathrm{D},H}$ be as in (4) and suppose that the density function $f \in C^{1,\alpha}(B_r)$. Then there exists a constant N_2 such that for all $n \geq N_2$, there holds*

$$\sup_{f \in C^{1,\alpha}} \mathcal{R}_{L,\mathrm{P}}(f_{\mathrm{D},H}) - \mathcal{R}^*_{L,\mathrm{P}} \gtrsim n^{-\frac{2}{2+d}}, \qquad (11)$$

in expectation with respect to $\mathrm{P}^n \otimes \mathrm{P}_H$.

Recall that in Theorem 2, as $n \to \infty$, the upper bound for our GBHT attains asymptotically convergence rate which is slightly faster than $n^{-2(1+\alpha)/(4(1+\alpha)+d)}$. When comparing Theorem 3 with Theorem 2, we find that for any $\alpha \in (0,1]$, if $d \geq 2(1+\alpha)/\alpha$, the upper bound of the convergence rate (10) for GBHT turns out to be smaller than the lower bound (11) for HT density estimators, which explains the benefits of the boosting procedure from the perspective of convergence rates.

5. Numerical Experiments

5.1. Generation Methods of Histogram Transforms

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 an orthogonal matrix R and an 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 the definition of R. In this case, we can change the sign of the first column of R to construct a new rotation matrix R^+.

We 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.

The translation vector b is drawn from the uniform distribution over the hypercube $[0,1]^d$.

5.2. Evaluation Criteria

Mean absolute error (*MAE*). The first criterion of evaluating the accuracy of density estimator is the mean absolute error, defined by $\mathit{MAE}(\widehat{f}) = \frac{1}{M}\sum_{j=1}^M |\widehat{f}(x_j) - f(x_j)|$, where $x_1, \ldots, x_M$ are test samples. It is used in synthetic data experiments where the true density function is known.

Average negative log-likelihood (*ANLL*). Another effective measure of estimation accuracy, especially when facing real data and the true density function is unknown, is the average negative log-likelihood, defined by $\mathit{ANLL}(\widehat{f}) = -\frac{1}{M}\sum_{j=1}^M \log \widehat{f}(x_j)$, where $\widehat{f}(x_j)$ represents the estimated probability density for the test sample x_j and M is the size of test samples. Note that the lower the *ANLL* is, the better estimation we obtain.

5.3. Empirical Understandings

In this part, we conduct simulations concerning GBHT for density estimation. Based on several synthetic datasets, we show the power of boosting procedure through simulations, and we illustrate a possible explanation for the enhancement in accuracy, i.e. the asymptotic smoothness achieved. Then we study a pair of important parameters for histogram transforms, $s_{\min}$ and $s_{\max}$.

5.3.1. Synthetic Data Settings

We base the simulations on four different types of synthetic distributions, each with dimension $d \in \{2, 5, 7\}$, respectively. The premise of constructing data sets is that we assume that the components $X_i \sim f_i$, $i = 1 \ldots, d$, of the random vector $X = (X_1, \ldots, X_d)$ are independent of each other. To be specific, Type I density function, representing a bimodal Gaussian distribution, enjoys high order of smoothness, while those for Types II and III are not continuous. Moreover, Types II and III represent density functions with bounded support and unbounded support, respectively. Finally, Type IV represents the case where the marginal distributions of each dimension are not identical. More detailed descriptions and visual illustrations are shown in Section C.1 of the supplementary material.

In the following experiments, we generate $2,000$ and $10,000$ i.i.d samples as training and testing data respectively from each type of synthetic datasets, and each with dimension $d \in \{2, 5, 7\}$.

5.3.2. The Power of Boosting

To show the behavior of T, we carry out the experiments with $T \in \{1, 5, 10, 20, 50, 100, 500, 1000\}$, and the other two hyper-parameters are chosen by 3-fold cross-validation. We pick $s_{\min}$ from the set $\{-3 + 0.5k, k = 0, \ldots, 12\}$ and $s_{\max} - s_{\min}$ is chosen from the set $\{0.5 + 0.5k, k = 0, \ldots, 5\}$. For each T we repeat this procedure for 10 times.

As can be seen in Figure 2, as T grows, the accuracy performance of GBHT (both *MAE* and *ANLL*) first enhances dramatically when T grows from 1 to $1,000$, but as T continues to grow, a steady state will be reached. This coincides with Theorem 2, where the convergence rate attains the optimum when T_n is greater than a certain value. Moreover, fewer iterations are required to make GBHT convergence when the dimension of input space is lower. A large number of iterations lead to a more accurate model but bring about the additional burden of computation.

For a possible explanation of the enhancement in estimation accuracy under the boosting procedure, we conduct simulations to show that GBHT achieves asymptotic smoothness with T increasing. For the sake of more clear visualization, we utilize a toy example with $2,000$ samples i.i.d. generated from the one-dimensional standard normal distribution, and use GBHT to conduct density estimation, where the number of trees T is set to $1, 5, 20, 50$, respectively.

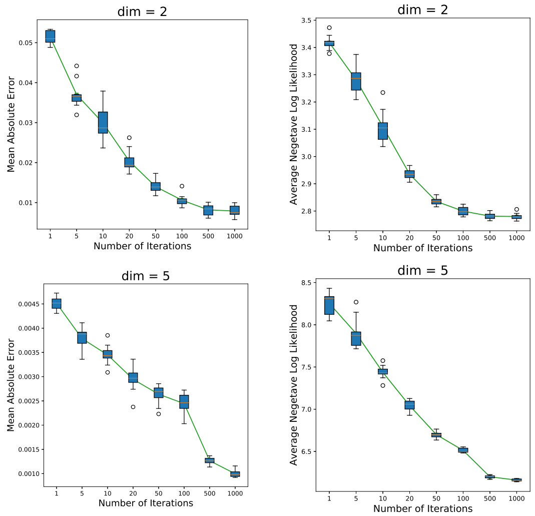
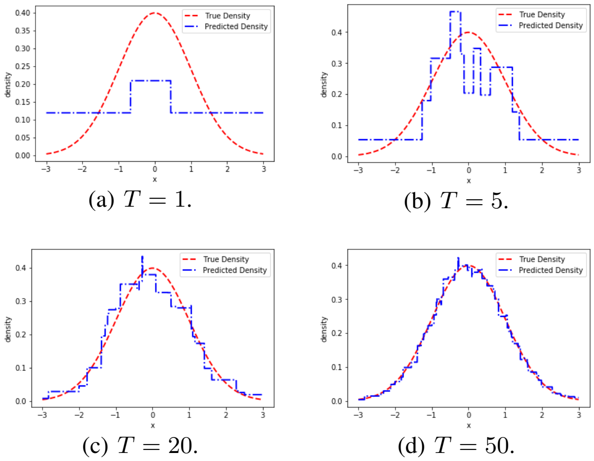


Figure 2. The study of parameter T on GBHT of Type I synthetic distribution, where the first row illustrates the low-dimensional results with dimension $d = 2$, and the second row indicates the high-dimensional results with dimension $d = 5$. The left column indicates how *MAE* varies along parameters T, and the right column shows the variation of *ANLL*.



(a) $T = 1$.

(b) $T = 5$.

(c) $T = 20$.

(d) $T = 50$.

Figure 3. The study of parameter T on GBHT of the Standard Normal distribution. The red line represents the underlying density while the blue one represents density estimator returned by GBHT.

From Figure 3 we see that with $T = 1$, the base estimator turns out to be a step function with discontinuous boundaries, and the estimation is far from satisfactory. Nevertheless, as the iteration T increases, the boosting estimator becomes more continuous and smooth with the corresponding accuracy enhancing greatly. With $T = 50$, our GBHT is nearly smooth and achieves high estimation accuracy.

Table 1. Average *ANLL* and *MAE* over simulated datasets

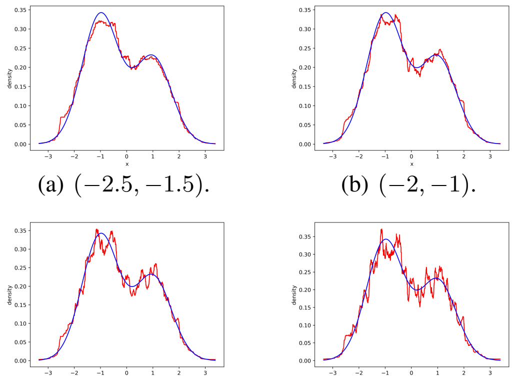
Method	d	Type I		Type II		Type III		Type IV	
		ANLL	*MAE*	*ANLL*	*MAE*	*ANLL*	*MAE*	*ANLL*	*MAE*
GBHT (Ours)	5	$\mathbf{6.26}$	$\mathbf{2.41e{-}3}$	$\mathbf{-0.80}$	$\mathbf{10.31}$	$\mathbf{8.23}$	$\mathbf{6.61e{-}4}$	$\mathbf{3.85}$	$\mathbf{0.14}$
KDE		6.33	$2.36e{-}3$	-0.32	12.40	8.65	$8.27e{-}4$	3.86	0.15
GBHT (Ours)	7	$\mathbf{8.36}$	$\mathbf{4.33e{-}4}$	$\mathbf{-0.45}$	$\mathbf{34.91}$	$\mathbf{10.81}$	$\mathbf{5.30e{-}5}$	$\mathbf{5.10}$	$\mathbf{0.18}$
KDE		8.77	$5.13e{-}4$	0.03	40.74	12.48	$6.05e{-}5$	5.16	0.18

* The best results are marked in **bold**.

5.3.3. Parameter Analysis

Here we mainly conduct experiments concerning the parameters of histogram transforms, namely the lower and upper scale parameters $s_{\min}, s_{\max} \in \mathbb{R}$. To this end, for the sake of clear visualization, we consider the Type I synthetic dataset of 1 dimension to see how these parameters affect the performance of GBHT.

Recall that the scale parameters $s_{\min}$ and $s_{\max}$ of the stretching matrix S control the size of histogram bins. Smaller bins are required for the regions with complex structures of the density function while those with simple structure calls for larger bins. A narrower range of bin size is accommodated to cope with the varying scales while to preserve a homogeneous structure. We conduct experiments over four pairs of scale parameters $(s_{\min}, s_{\max}) \in \{(-2.5, -1.5), (-2, -1), (-1.5, -0.5), (-1, 0)\}$. We select $T = 500$ to make the density estimator convergence with sufficient boosting iterations.



(a) $(-2.5, -1.5)$. (b) $(-2, -1)$.

(c) $(-1.5, -0.5)$. (d) $(-1, 0)$.

Figure 4. The study of parameter $s_{\min}$ and $s_{\max}$ on GBHT of the Type I synthetic distribution. The red line represents the density estimator returned by GBHT algorithm while the blue one represents the underlying density function. And the tuples in subtitle represent $(s_{\min}, s_{\max})$.

As is shown in Figure 4, lower values of these parameters lead to a coarser approximation of the underlying density function, which results in the loss of precision. Figure 4(a) implies that the density estimator is underfitting when the bin width is too large. On the contrary, if the bin width is too small, then there are few samples lying in most of the histogram bins and thus overfitting occurs as shown in Figure 4(d). Therefore, it is of great importance to choose $s_{\min}$ and $s_{\max}$ properly.

5.4. Performance Comparisons

In this section, we conduct performance comparisons on both synthetic and real datasets.

5.4.1. Synthetic Data Comparisons

Following the experimental settings in Section 5.3, we conduct empirical comparisons between GBHT and the prevailing KDE to further demonstrate the desirable performance of GBHT under synthetic datasets. Table 1 records average *ANLL* and *MAE* over simulation data sets for KDE and GBHT with $T = 1{,}000$. For higher dimensions $d = 5$ and $d = 7$, our GBHT always outperforms KDE in terms of *ANLL* and *MAE*.

5.4.2. Real Data Comparisons

We conduct real data comparisons on real datasets from the UCI repository. We put the detailed description of datasets in Section C.2 of the supplement.

Experimental Settings. In order to evaluate the performance of density estimators on datasets with various dimensions, we apply the following data preprocessing pipeline. Firstly, we remove duplicate observations as well as those with missing values. Then each dimension of the datasets is scaled to $[0, 1]$ and each dataset is reduced to lower dimensions d' through PCA, e.g. to 10%, 30%, 50% and 70% of the original dimension d, respectively. Finally, in each dataset, we randomly select 70% of the samples for training and the remaining 30% for testing.

The number of iterations T is set to be 100 and the other two hyper-parameters $s_{\min}$ and $s_{\max} - s_{\min}$ are chosen from $\{-2 + 0.5k, k = 0, \ldots, 8\}$ and $\{0.5 + 0.5k, k = 0, \ldots, 5\}$, respectively, by 3-fold cross-validation. We repeat this procedure 10 times to evaluate the standard deviation for *ANLL*. The average *ANLL* on test sets are recorded in Table 2.

Since real density often resides in a low-dimensional manifold instead of filling the whole high-dimensional space, it

Table 2. Average *ANLL* over real data sets

Datasets	d'	GBHT (Ours)	KDE	Datasets	d'	GBHT (Ours)	KDE
Adult	2	$-\mathbf{1.2371}$ (0.0312)	-0.7402 (0.0027)	Diabetes	1	$-\mathbf{0.7057}$ (0.1253)	-0.2627 (0.0111)
	4	$-\mathbf{1.9312}$ (0.0667)	-0.3075 (0.0032)		3	$-\mathbf{1.5982}$ (0.1011)	-0.4042 (0.0403)
	8	$-\mathbf{5.5922}$ (0.1097)	-2.2970 (0.0108)		4	$-\mathbf{1.8605}$ (0.1424)	-0.8353 (0.0773)
	10	$-\mathbf{6.0740}$ (0.1044)	-3.4372 (0.0110)		6	$-\mathbf{2.6134}$ (0.2310)	-1.9693 (0.1550)
Australian	2	$-\mathbf{0.7966}$ (0.0904)	1.3155 (0.0234)	Ionosphere	3	$\mathbf{2.8681}$ (0.0917)	2.9544 (0.0423)
	4	$-\mathbf{5.8510}$ (0.2947)	0.8518 (0.0291)		10	$\mathbf{4.1625}$ (0.2150)	4.6447 (0.4448)
	8	$-\mathbf{3.7957}$ (0.5823)	0.6879 (0.1056)		17	$\mathbf{3.8920}$ (0.4198)	5.3236 (0.9654)
	10	$-\mathbf{1.3659}$ (0.4382)	0.4995 (0.1748)		24	$\mathbf{2.1412}$ (0.6710)	4.5570 (1.3684)
Breast-cancer	1	$\mathbf{0.3580}$ (0.0561)	0.6907 (0.0394)	Parkinsons	2	$-\mathbf{0.9465}$ (0.0402)	-0.0847 (0.0094)
	3	$-\mathbf{0.5446}$ (0.1887)	0.1743 (0.1268)		7	$-\mathbf{5.7700}$ (0.1439)	-2.1513 (0.0189)
	6	$-\mathbf{3.2099}$ (0.6068)	-1.1397 (0.2788)		11	$-\mathbf{10.0932}$ (0.1492)	-7.8291 (0.0340)
	8	$-\mathbf{6.4362}$ (0.8144)	-2.1110 (0.3906)		15	$-\mathbf{16.9316}$ (0.2151)	-16.8767 (0.1025)

* The best results are marked in **bold**, and the standard deviation is reported in the parenthesis.

is reasonable to study the density estimation problem after dimensionality reduction. Therefore, in data preprocessing, all data sets are reduced to various lower dimensions through PCA. However, we need to take the to-be-reduced dimension as a hyper-parameter, since in general, the dimension of the manifold is unknown.

Experimental Results. In Table 2, we summarize the comparisons with the state-of-the-art density estimator KDE on six real datasets, which demonstrates the accuracy of our GBHT algorithm. For most of the redacted datasets, GBHT shows its superiority on accuracy, whereas the standard deviation of GBHT is slightly larger than that of KDE due to the randomness of histogram transforms.

5.5. Gradient Boosted Histogram Transform (GBHT) for Anomaly Detection

To showcase a potential application of GBHT, we propose a density-based method for anomaly detection. Given a density level ρ, we regard the sample points with low density estimation $\{x_i \in D \mid f_{\mathrm{D},\lambda}(x_i) \leq \rho\}$ as anomaly points. Based on GBHT density estimation, we are able to present the *Gradient Boosting Histogram Transform* (*GBHT*) for anomaly detection in Algorithm 2.

Algorithm 2 GBHT for Anomaly Detection

```
Input: Training data $D := \{x_1, \dots, x_n\}$;
       Density threshold parameters $\rho$.
Compute GBHT $f_{\mathrm{D},\lambda}$ (6).
Output: Recognize anomalies as
        $\{x_i \in D \mid f_{\mathrm{D},\lambda}(x_i) \leq \rho\}$.
```

We conduct numerical experiments to make a comparison between our GBHT and several popular anomaly detection algorithms such as the forest-based Isolation Forest (iForest) (Liu et al., 2008), the distance-based k-Nearest Neighbor (k-NN) (Ramaswamy et al., 2000) and Local Outlier Factor (LOF) (Breunig et al., 2000), and the kernel-based one-class SVM (OCSVM) (Schölkopf et al., 2001), on 20 real-world benchmark outlier detection datasets from the ODDS library. Detailed experimental settings and comparison results are shown in Section C.3.

In the aspect of best performance, our method GBHT wins in 7 out of 20 datasets while the iForest and OCSVM, wins both 4 out of 20 datasets, respectively. Moreover, our GBHT ranks the second on 5 datasets. Finally, in the aspect of the average performance of benchmark datasets, our method has the lowest rank sum. Overall, our experiments on benchmark datasets show that our method has favorable performance among competitive anomaly detection algorithms.

6. Conclusion

In this paper, we propose an algorithm called *Gradient Boosting Histogram Transform* (*GBHT*) for density estimation with novel theoretical analysis under the RERM framework. It is well-known that boosting methods are hard to apply in unsupervised learning. Therefore, we turn the density estimation into a supervised learning problem by changing the loss function to *Negative Log Likelihood* loss, which measures the proximity between the estimated density and the true one. In each iteration of boosting methods, histogram transform first randomly stretches, rotates and translates the feature space for acquiring more information and then an additional density function is attached on the estimated one with weights, which guarantees that the result is a density function with integral equals to 1. For theoretical achievements, we prove convergence properties of our algorithm under mild assumptions. It should be highlighted that we are the first to explain the benefits of the boosting procedure for density estimation algorithms. Last but not least, numerical experiments of both synthetic data and real data are carried out to verify the promising performance of GBHT with applications to anomaly detection.

References

Amarbayasgalan, T., Jargalsaikhan, B., and Ryu, K. H. Unsupervised novelty detection using deep autoencoders with density based clustering. *Applied Sciences*, 8(9): 1468, 2018.

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.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. Lof: identifying density-based local outliers. In *ACM Sigmod Record*, volume 29, pp. 93–104. ACM, 2000.

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

Cai, Y., Hang, H., Yang, H., and Lin, Z. Boosted histogram transform for regression. In *International Conference on Machine Learning*, pp. 1251–1261. PMLR, 2020.

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.

Chen, Y., Hu, X., Fan, W., Shen, L., Zhang, Z., Liu, X., Du, J., Li, H., Chen, Y., and Li, H. Fast density peak clustering for large scale data based on knn. *Knowledge-Based Systems*, 187:104824, 2020.

Cortes, C., Mohri, M., and Storcheus, D. Regularized gradient boosting. *Advances in Neural Information Processing Systems*, 32:5449–5458, 2019.

Criminisi, A. and Shotton, J. *Decision Forests for Computer Vision and Medical Image Analysis*. Springer Science & Business Media, 2013.

Criminisi, A., Shotton, J., and Konukoglu, E. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. *Microsoft Research Technical Report 2011–114*, 2011.

Duan, T., Anand, A., Ding, D. Y., Thai, K. K., Basu, S., Ng, A., and Schuler, A. Ngboost: Natural gradient boosting for probabilistic prediction. In *International Conference on Machine Learning*, pp. 2690–2700. PMLR, 2020.

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

Ghaffari, M., Lattanzi, S., and Mitrović, S. Improved parallel algorithms for density-based network clustering. In *International Conference on Machine Learning*, pp. 2201–2210. PMLR, 2019.

Jang, J. and Jiang, H. Dbscan++: Towards fast and scalable density clustering. In *International Conference on Machine Learning*, pp. 3019–3029. PMLR, 2019.

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.

Klemelä, J. Multivariate histograms with data-dependent partitions. *Statistica Sinica*, 19(1):159–176, 2009.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. Isolation forest. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 413–422, 2008.

Liu, L. and Wong, W. H. Multivariate density estimation via adaptive partitioning (I): sieve MLE. *arXiv preprint arXiv:1401.2597*, 2014.

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

Mathiasen, A., Larsen, K. G., and Grønlund, A. Optimal minimal margin maximization with boosting. In *International Conference on Machine Learning*, pp. 4392–4401. PMLR, 2019.

Nachman, B. and Shih, D. Anomaly detection with density estimation. *Physical Review D*, 101(7):075042, 2020.

Parmar, M., Wang, D., Zhang, X., Tan, A.-H., Miao, C., Jiang, J., and Zhou, Y. Redpc: A residual error-based density peak clustering algorithm. *Neurocomputing*, 348: 82–96, 2019.

Parnell, T., Anghel, A., Łazuka, M., Ioannou, N., Kurella, S., Agarwal, P., Papandreou, N., and Pozidis, H. Snapboost: A heterogeneous boosting machine. *Advances in Neural Information Processing Systems*, 33, 2020.

Ram, P. and Gray, A. G. Density estimation trees. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 627–635. ACM, 2011.

Ramaswamy, S., Rastogi, R., and Shim, K. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 427–438, 2000.

Ridgeway, G. Looking for lumps: Boosting and bagging for density estimation. *Computational Statistics & Data Analysis*, 38(4):379–392, 2002.

Rosset, S. and Segal, E. Boosting density estimation. In *Advances in Neural Information Processing Systems*, pp. 657–664, 2003.

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.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7): 1443–1471, 2001.

Scott, D. W. *Multivariate Density Estimation*. John Wiley & Sons, Inc., Hoboken, NJ, second edition, 2015.

Suggala, A., Liu, B., and Ravikumar, P. Generalized boosting. *Advances in Neural Information Processing Systems*, 33, 2020.

Zhang, L., Lin, J., and Karim, R. Adaptive kernel density-based anomaly detection for nonlinear systems. *Knowledge-Based Systems*, 139:50–63, 2018.