Boosting in Univariate Nonparametric Maximum Likelihood Estimation

YunPeng Li and ZhaoHui Ye

***Abstract*—Nonparametric maximum likelihood estimation is intended to infer the unknown density distribution while making as few assumptions as possible. To alleviate the over parameterization in nonparametric data fitting, smoothing assumptions are usually merged into the estimation. In this letter a novel boosting-based method is introduced to the nonparametric estimation in univariate cases. We deduce the boosting algorithm by the second-order approximation of nonparametric log-likelihood. Gaussian kernel and smooth spline are chosen as weak learners in boosting to satisfy the smoothing assumptions. Simulations and real data experiments demonstrate the efficacy of the proposed approach.**

***Index Terms*—Boosting, kernel, nonparametric maximum likelihood estimation, second-order approximation, smoothing spline.**

I. Introduction

NONPARAMETRIC maximum likelihood estimation (NPMLE) [1]–[3] has received much attention in recent years. It has been successfully applied to various problems in signal processing, statistical learning, and pattern recognition. Given finite independent identically distributed (IID) random samples from an unknown distribution, the goal of NPMLE is to estimate the probability density with as few assumptions as possible. Unfortunately, NPMLE's optimization over an infinite-dimensional function space often leads to the unbounded likelihood and overfitting. The remedy to alleviate these defects is merging additional assumptions or constraints into the estimation. These assumptions confine the nonparametric density estimation to certain functional spaces. One of the most popular methods is to impose the smoothing constraint on the unknown distribution to restrict the estimation.

There are currently two common approaches to utilize the smoothing constraint: restriction methods and regularization methods. Conventional restriction methods control the smoothing degree via predetermined smoothing parameters (such as the number of bins in the histogram, the number of observations in the nearest-neighbor method, the bandwidth in kernel methods [4]–[6] and the local polynomials [7], [8]). Another kind of restriction methods supposes the unknown distribution owns special structures like mixture models [9]–[12], shape constraints (log-concavity [13] and monotonicity [14]). In regularization methods, penalty terms (such as roughness [15], [16], L_1 penalty [17], total variation [18]) are designed to control the smoothing degree. For roughness penalty, nonparametric maximum penalty likelihood is analyzed in the reproducing kernel Hilbert spaces [19] and one of its estimates is proven to be a positive exponential smooth spline [20] with knots only at the sample points [19], [21].

However, most of the restriction and selection methods have to determine the tuning parameters beforehand [6], [22], resulting in a lack of flexibility in inference. In this letter, a selection method is introduced to NPMLE in the boosting form. The proposed algorithm adaptively scans the function spaces and includes only those that contribute significantly to estimation.

Our contributions are as follows.

1) We derive the boosting algorithm from the second-order approximation of nonparametric log-likelihood.
2) We select several weak learners for boosting NPMLE. Different from the regularization methods, those weak learners share the fixed smoothing degree at each iteration. The only meta-parameter in boosting NPMLE is the number of boosting iterations.

II. Proposed Method

Let X be a random variable in $\mathbb{R}$ with probability density $p(x)$. Given N independent identically distributed samples $X_1, X_2, \ldots, X_N$, we model the density estimate $\hat{p}(x)$ in the form of Gibbs distribution.

$$\hat{p}(x) = \frac{e^{f(x)}}{\int e^{f(x)} dx} \tag{1}$$

where $f(x)$ is assumed to be a smooth function in $\mathbb{R}$. The log likelihood $L(f)$ is defined as the function of $f(x)$.

$$L(f) = \frac{1}{N} \sum_{j=1}^{N} log\, \hat{p}(X_j) \tag{2}$$

Given N samples $X_1, X_2, \ldots, X_N$, their unique elements in ascending order are $x_1, x_2, \ldots, x_n$, and the corresponding frequency q_i for x_i is,

$$q_i = \frac{1}{N} \sum_{j=1}^{N} \mathbb{I}(X_j = x_i) \tag{3}$$

where $\mathbb{I}$ is the indicator function. We restrict the support of $\hat{p}(x)$ in $[x_1, x_n]$. The trapezoidal rule is used for numerical integration in equation (1), where a_i is the coefficient concerning x_i. Then,

Manuscript received January 22, 2021; revised March 1, 2021; accepted March 8, 2021. Date of publication March 12, 2021; date of current version April 8, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiangui Kang. *(Corresponding author: YunPeng Li.)*

The authors are with the Department of Automation, Tsinghua University, Beijing 100084, China (e-mail: liyp18@mails.tsinghua.edu.cn; yezhaohui@mail.tsinghua.edu.cn).

Digital Object Identifier 10.1109/LSP.2021.3065881

the estimation $\hat{p}(x)$ and the log-likelihood $L(f)$ in equation (2) are transformed to the following form

$$\hat{p}(x) = \frac{e^{f(x)}}{\sum_{i=1}^{n} a_i e^{f(x_i)}} \tag{4}$$

$$\begin{aligned} L(f) &= \sum_{i=1}^{n} q_i log\, \hat{p}(x_i) \\ &= \sum_{i=1}^{n} q_i f(x_i) - log \sum_{i=1}^{n} a_i e^{f(x_i)} \end{aligned} \tag{5}$$

To avoid the summation in logarithm in equation (5), we replace the original $L(f)$ with a simpler surrogate $\mathcal{L}(f)$ according to the inequality $log\, v \leq -1 + v$.

$$\mathcal{L}(f) = \sum_{i=1}^{n} q_i f(x_i) - \sum_{i=1}^{n} a_i e^{f(x_i)} \tag{6}$$

$$\begin{aligned} L(f) &\geq 1 + \sum_{i=1}^{n} q_i f(x_i) - \sum_{i=1}^{n} a_i e^{f(x_i)} \\ &\geq 1 + \mathcal{L}(f) \end{aligned} \tag{7}$$

It can be proved that the original $L(f)$ and surrogate $\mathcal{L}(f)$ have an identical maximum point. Thus, we optimize the surrogate $\mathcal{L}(f)$ as the objective function in the NPMLE.

In the remaining part, We firstly derive the boosting algorithm to optimize $\mathcal{L}(f)$. Then we select several weak learners to validate the proposed method.

A. Boosting NPMLE

Boosting [23] is a technique of combining multiple weak learners to produce a powerful committee, whose performance is significantly better than any of the weak learners. It works by applying the weak learner sequentially to a dataset of weighted form. For applying the boosting principle to NPMLE, we express $f(x)$ as a combination of weak learner $b(x; \gamma_m)$

$$f(x) = \sum_{m=1}^{M} b(x; \gamma_m) \tag{8}$$

where M is the number of boosting iterations and m is the index of the single iteration. At each iteration, we train a single weak learner $b(x; \gamma_m)$ on the weighted data, characterized by a set of parameters γ_m. Thus, the maximum log-likelihood in $\mathcal{L}(f)$ is changed into a boosting form

$$\max_{\{\gamma_m\}_1^M} \quad \mathcal{L}\left(\sum_{m=1}^{M} b(x; \gamma_m)\right) \tag{9}$$

We define the $f(x)$ at m iteration as $f_m(x)$.

$$\begin{aligned} f_m(x) &= \sum_{i=1}^{m} b(x; \gamma_i) \\ &= f_{m-1}(x) + b(x; \gamma_m) \end{aligned} \tag{10}$$

The key of boosting is that no earlier parameters γ are adjusted at the current m iteration. To acquire $f_m(x)$, we optimize a subproblem based on former $f_{m-1}(x)$ sequentially.

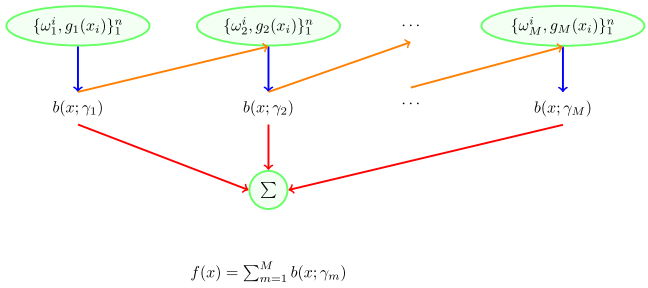


Fig. 1. Schematic of boosting in NPMLE. Weak learners are trained on the updated data depend on the performance of the previous iterations, and then combined to produce the final model.

$$\max_{\gamma_m} \quad \mathcal{L}(f_{m-1}(x) + b(x; \gamma_m)) \tag{11}$$

A second-order approximation $\mathcal{L}(f_m; f_{m-1})$ (equivalent to the Newton-Raphson method) is used to solve $\mathcal{L}(f_m)$ in equation (11),

$$\begin{aligned} \mathcal{L}(f_m; f_{m-1}) \approx\ & \mathcal{L}(f_{m-1}(x)) \\ & + \sum_{i=1}^{n} (q_i - a_i e^{f_{m-1}(x_i)})(f_m(x_i) - f_{m-1}(x_i)) \\ & + \sum_{i=1}^{n} \frac{1}{2}(-a_i e^{f_{m-1}(x_i)})(f_m(x_i) - f_{m-1}(x_i))^2 \end{aligned} \tag{12}$$

Maximizing $\mathcal{L}(f_m; f_{m-1})$ is equivalent to the minimizing of an iterative reweighted least squares (IRLS) [24], [25] problem as follow,

$$\min_{\gamma_m} \quad \sum_{i=1}^{n} \frac{1}{2} \omega_m^i (b(x_i; \gamma_m) - g_m(x_i))^2 \tag{13}$$

$$\omega_m^i = a_i e^{f_{m-1}(x_i)} \tag{14}$$

$$g_m(x_i) = \frac{q_i - \omega_m^i}{\omega_m^i} \tag{15}$$

where ω_m^i is the weight and $g_m(x_i)$ is the response of x_i in the current m iteration. For the next $m+1$ iteration, the updating rules concerning the weight and response are

$$\omega_{m+1}^i = \omega_m^i e^{b(x_i; \gamma_m)} \tag{16}$$

$$g_{m+1}(x_i) = \frac{q_i - \omega_{m+1}^i}{\omega_{m+1}^i} \tag{17}$$

The parameters γ_m in single $b(x; \gamma_m)$ are determined by equation (13). Once all the weak learners $b(x; \gamma)$ have been trained, $f(x)$ is the combination of whole M weak learners, as illustrated schematically in Fig. 1. The whole algorithm is summarized in Algorithm 1. Different from existing boosting algorithm in classification [26] and regression [27], boosting NPMLE updates the weight and response of data simultaneously.

Algorithm 1: Boosting NPMLE.

Initialization
$\omega_0^i \leftarrow \frac{1}{n}$
$b(x_i;\gamma_0) \leftarrow 0$
$f_0(x_i) \leftarrow 0$
for $m = 1$ to M **do**
  for $i = 1$ to n **do**
    $\omega_m^i \leftarrow \omega_{m-1}^i e^{b(x_i;\gamma_{m-1})}$
    $g_m(x_i) \leftarrow \frac{q_i - \omega_m^i}{\omega_m^i}$
  end for
  compute
  $\min_{\gamma_m} \sum_{i=1}^n \frac{1}{2}\omega_m^i (b(x_i;\gamma_m) - g_m(x_i))^2$
  $f_m(x) \leftarrow f_{m-1}(x) + b(x;\gamma_m)$
end for
output $f(x) \leftarrow f_M(x)$

B. Choices of Weak Learners

The choices of the weak learners $b(x;\gamma_m)$ and the number of boosting iterations M are the key to boosting NPMLE. Although conventional classification and regression trees (CART) [23], [26], [27] can solve equation (13) efficiently, CART cannot satisfy the smoothing constraint required in boosting NPMLE due to its feature of piecewise constant. Despite boosting method usually reduces training error as the increase of boosting iterations M, it can sometimes cause overfitting on future predictions.

In boosting NPMLE, ideal weak learners should meet several requirements:

1) the weak learners should satisfy the smoothing constraint in NPMLE to avoid over parameterization.
2) the weak learners can solve the weighted least squares in equation (13) efficiently.
3) the model complexity of the weak learners can be easily restricted during each boosting iteration m.
4) the weak learners should be robust to the large choice of boosting iterations M.

We select the Gaussian kernel and the smooth spline as weak learners in boosting NPMLE for the following reasons:

1) Gaussian kernel: kernel functions are basis functions for nonlinear mapping, determined by kernel choice and bandwidth. An L_2 penalty is added to equation (13) to control their model complexity by the lagrange multiplier $lambda$. These models change from the simple fit to ordinary least squares as the decrease of $lambda$. We select the extremely large choice of $lambda$ ($lambda = 10^4$) in boosting NPMLE to constrain their model complexity [28].
2) smooth spline: different from CART method, smooth spline is piecewise cubic polynomials under the smoothing constraint. It has been applied and analyzed in nonparametric estimation in regularization methods [19], [21]. We fix the complexity of smooth spline via a parameter named degree of freedom df ($df = 3$). With the increase of df from 2 to n, the $b(x;\gamma_m)$ changes from a simple line fit to ordinary least squares interpolation.

The selection methods in the proposed paper do not focus on the smoothing parameters for single $b(x;\gamma_m)$. We only determine these weak learners to be extremely simple in each iteration by the fixed $lambda$ (Gaussian kernel) or df (smooth spline). This is the fundamental difference between the existing regularization methods [15]–[18]. As a result, our algorithm avoids the difficult choices of tuning parameters. Besides, the extreme choices of large $lambda$ and small df enable boosting NPMLE robust to the large boosting iterations M.

TABLE I
IMPLEMENTATION DETAILS

weak learners	package	parameters
CART	rpart	$minsplit$ = 30.
Gaussian kernel	density;glmnet	$kernel$ ="gaussian";$alpha$ = 0, $family$ = "gaussian",$lambda$ = 10^4.
smooth spline	smooth.spline	df = 3.

III. Experiments and Results

In this section, simulations, and experiment on real data are designed to verify the performance of boosting NPMLE in univariate cases. The only tuning parameter is the number of boosting iterations M, more details are shown in Table I.

A. Improvement in Data Fitting

In the first simulation, we apply boosting NPMLE to density estimation of different distributions, ranging from discontinuous to continuous, the sample size N is 500. As can be seen in Fig. 2(a), when $M = 1$, Gaussian kernel and smooth spline do not fit well in all cases, while CART performs well only in uniform distribution owe to its feature of piecewise constant. After $M = 200$ boosting iterations, in Fig. 2(b), we find the estimation results of Gaussian kernel and smooth spline become closer to the ground-truth for all distributions, with performance surpassing CART. We can conclude that the performance of data fitting is significantly improved as the increase of boosting iterations M for Gaussian kernel and smooth spline, and the CART is actually not appropriate to be the weak learners in boosting NPMLE.

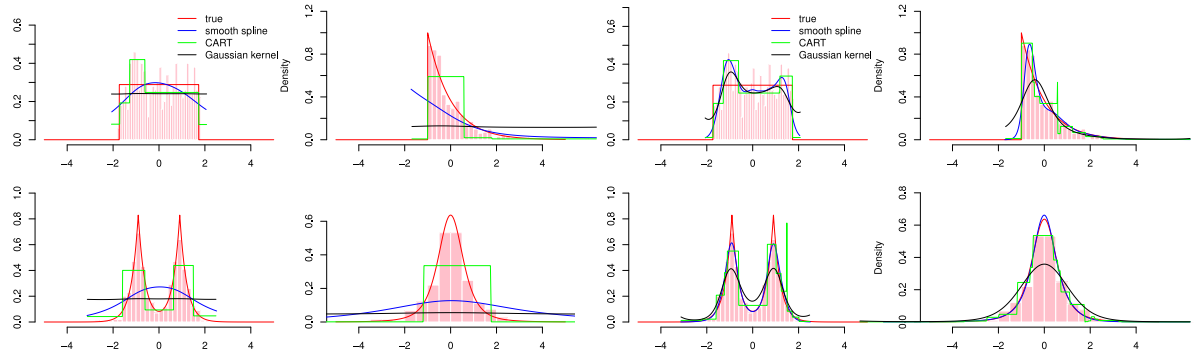
B. Robustness to the Choice of M

Although large M strikingly improves the data fitting in train stage, inappropriate choice of large M usually leads to overfitting in prediction for ordinary boosting methods. Another simulation is conducted to evaluate the robustness of boosting NPMLE concerning M. In this simulation, we use the Gaussian kernel and smooth spline as weak learners and fix the true distribution $p(x)$ as a Gaussian Mixture Model (GMM), where the sample size is $N = 500$.

$$p(x) = \beta\phi(x;\mu_1,\sigma_1^2) + (1-\beta)\phi(x;\mu_2,\sigma_2^2) \quad (18)$$

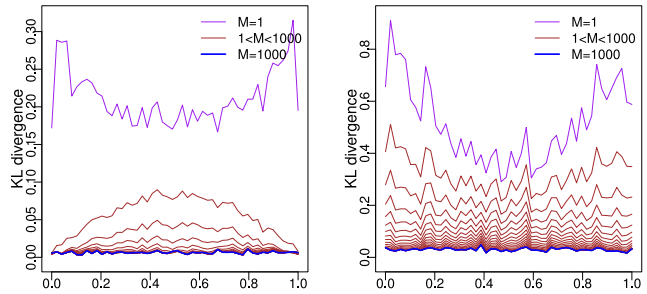
$$\phi(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (19)$$

where $\beta \in [0,1]$, $\mu_{1,2} = \pm 2.5$, $\sigma_{1,2}^2 = 2$. We increase the β from 0 to 1 and choose different M in boosting NPMLE. The KL divergence $D_{KL}(p||\hat{p})$ is used to measure the distance between



(a) Boosting iteration $M = 1$ (b) Boosting iteration $M = 200$

Fig. 2. Application of boosting to the density estimation with different boosting iterations. Smooth spline (blue), CART (green), and Gaussian kernel (black) work as weak learners to estimate the true distributions (red), the histograms (pink) are presented for comparing. (Top left) uniform distribution; (Top right) exponential distribution; (Bottom left) mixture of two double exponential distributions; (Bottom right) student distribution.



(a) smooth spline (b) Gaussian kernel

Fig. 3. Average KL divergence of boosting NPMLE in GMM based on 50 simulations. The ordinate is the KL divergence $D_{KL}(p||\hat{p})$, and the abscissa represents the increasing sequence concerning β. In the left figure, smooth spline works as weak learners, while Gaussian kernel works as weak learners in right figure. In both figures, the corresponding weak learners have different number of iterations M such as $M = 1$ (purple), $1 < M < 1000$ (brown), $M = 1000$ (blue).

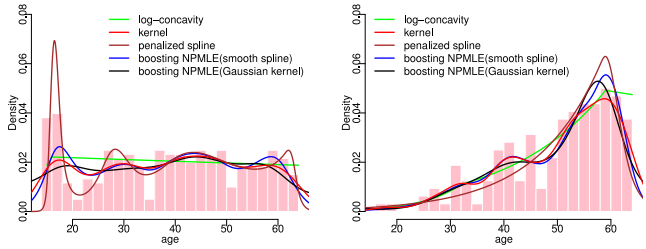
the true distribution $p(x)$ and the estimation $\hat{p}(x)$.

$$D_{KL}(p||\hat{p}) = \int p(x) log \frac{p(x)}{\hat{p}(x)} dx \tag{20}$$

In Fig. 3, with the increase of M, the KL divergences approach zero and their envelopes become surprisingly denser, which indicates that a parsimonious updating strategy is adopted by our weak learners to alleviate the risk of overfitting in the remaining iterations.

C. Evaluation on Pattern Classification

We evaluate our algorithm on the South African Heart Disease dataset [29] for pattern classification, which contains 462 patterns (70% for the training set and 30% for the testing set). We use only the quantitative input feature age (age at onset) to predict the binary response chd (coronary heart disease). The conditional probabilities $p(age|chd)$ are estimated by the proposed algorithms (smooth spline and Gaussian kernel) in Fig. 4. We use bayesian classifiers to compare boosting NPMLE with other NPMLE methods including log-concavity [30], kernel [31], and



(a) $chd = 0$ (b) $chd = 1$

Fig. 4. Estimated conditional densities $p(age|chd)$ to the South African Heart Disease dataset by boosting NPMLE. Histograms of age for the binary response chd separately.

TABLE II
COMPARISONS OF PERFORMANCE OF DIFFERENT NPMLE METHODS ON SOUTH AFRICAN HEART DISEASE DATASET

NPMLE method	Misclassification Rate(%)	
	Training set	Testing set
log-concavity	31.38(±1.59)	32.67(±3.33)
kernel	30.82(±1.55)	32.63(±3.37)
penalized spline	30.91(±1.60)	33.89(±3.55)
boosting NPMLE(smooth spline)	**30.99(±1.70)**	**32.58(±3.63)**
boosting NPMLE(Gaussian kernel)	**30.98(±1.74)**	**32.38(±3.77)**

penalized spline [32] (both default parameters). Thanks to the robustness of boosting NPMLE, boosting iterations can be selected extremely large ($M = 2000$). The average misclassification rate on 100 random splits is recorded in Table II. Our algorithm is consistent with other NPMLE methods in this task.

IV. CONCLUSION

In this letter, a novel selection algorithm based on boosting has been proposed to solve NPMLE. We derive the boosting NPMLE by second-order approximation to log-likelihood. Different from ordinary boosting in supervised learning, our algorithm adjusts both the weight and response during the sequential routine. Several weak learners are chosen to comply with the smoothing assumptions required in NPMLE. Simulations and classification experiment validate the effectiveness of the proposed algorithm.

REFERENCES

[1] J. Kiefer and J. Wolfowitz, "Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters," *Ann. Math. Statist.*, vol. 27, no. 4, pp. 887–906, 1956.

[2] S. J. Bean and C. P. Tsokos, "Developments in nonparametric density estimation," *Int. Statist. Rev.*, vol. 48, no. 3, pp. 267–287, 1980.

[3] A. J. Izenman, "Review papers: Recent developments in nonparametric density estimation," *J. Amer. Statist. Assoc.*, vol. 86, no. 413, pp. 205–224, 1991.

[4] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, 1962.

[5] E. A. Nadaraya, "On non-parametric estimates of density functions and regression curves," *Theory Probability Appl.*, vol. 10, no. 1, pp. 186–5, 1965.

[6] J. Racine, "An efficient cross-validation algorithm for window width selection for nonparametric kernel regression," *Commun. Statist. Simul. Comput.*, vol. 22, no. 4, pp. 1107–1114, 1993.

[7] C. Loader, *Local Regression and Likelihood* (Series Statistics and Computing). 1st ed. Springer, 1999.

[8] M. D. Cattaneo, M. Jansson, and X. Ma, "Simple local polynomial density estimators," *J. Amer. Statist. Assoc.*, vol. 115, no. 531, pp. 1449–1455, 2020.

[9] N. Laird, "Nonparametric maximum likelihood estimation of a mixing distribution," *J. Amer. Statist. Assoc.*, vol. 73, no. 364, pp. 805–811, 1978.

[10] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Rev.*, vol. 26, no. 2, pp. 195–239, 1984.

[11] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *J. Amer. Statist. Assoc.*, vol. 90, no. 430, pp. 577–588, 1995.

[12] P. R. Hahn, R. Martin, and S. G. Walker, "On recursive Bayesian predictive distributions," *J. Amer. Statist. Assoc.*, vol. 113, no. 523, pp. 1085–1093, 2018.

[13] R. J. Samworth, "Recent progress in log-concave density estimation," *Statist. Sci.*, vol. 33, no. 4, pp. 493–509, 2018.

[14] M.-Y. Cheng, T. Gasser, and P. Hall, "Nonparametric density estimation under unimodality and monotonicity constraints," *J. Comput. Graph. Statist.*, vol. 8, no. 1, pp. 1–21, 1999.

[15] I. J. Goodd and R. A. Gaskins, "Nonparametric roughness penalties for probability densities," *Biometrika*, vol. 58, no. 2, pp. 255–277, Aug. 1971.

[16] B. W. Silverman, "On the estimation of a probability density function by the maximum penalized likelihood method," *Ann. Statist.*, vol. 10, no. 3, pp. 795–810, 1982.

[17] F. Bunea, A. Tsybakov, and M. Wegkamp, "Sparse density estimation with l1 penalties," in *Proc. Int. Conf. Comput. Learn. Theory,* Jun. 2007, pp. 530–543.

[18] R. Koenker and I. Mizera, "Density estimation by total variation regularization," in *Proc. Adv. Stat. Model. Inf. Essaysin Honor Kjell A Doksum.*, 2007, pp. 613–634.

[19] G. F. de Montricher, R. A. Tapia, and J. R. Thompson, "Nonparametric maximum likelihood estimation of probability densities by penalty function methods," *Ann. Statist.*, vol. 3, no. 6, pp. 1329–1348, Nov. 1975.

[20] C. D. Boor, *A Practical Guide to Splines* (Series Applied Mathematical Sciences). vol. 27, Revised Edition, Springer, 2001.

[21] E. J. Wegman and I. W. Wright, "Splines in statistics," *J. Amer. Statist. Assoc.*, vol. 78, no. 382, pp. 351–365, 1983.

[22] C. Gu and J. Wang, "Penalized likelihood density estimation: Direct cross-validation and scalable approximation," *Statisti. Sinica*, vol. 13, pp. 811–826, 2003.

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

[24] R. Wolke and H. Schwetlick, "Iteratively reweighted least squares: Algorithms, convergence analysis, and numerical comparisons," *SIAM J. Sci. Statist. Comput.*, vol. 9, no. 5, pp. 907–921, 1988.

[25] D. P. O'Leary, "Robust regression computation using iteratively reweighted least squares," *SIAM J. Matrix Anal. Appl.*, vol. 11, no. 3, pp. 466–480, 1990.

[26] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

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

[28] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for Cox's proportional hazards model via coordinate descent," *J. Statist. Softw.*, vol. 39, no. 5, pp. 1–13, 2011.

[29] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009, ch. 6, pp. 214–215.

[30] L. Dümbgen and K. Rufibach, "logcondens: Computations related to univariate log-concave density estimation," *J. Statist. Softw.*, vol. 39, no. 6, pp. 1–28, 2011.

[31] T. Duong, "KS: Kernel smoothing," R Package version 1.12.0, 2021. [Online]. Available: https://cran.r-project.org/package/ks/index.html

[32] C. Kooperberg, "Logspline: Routines for Logspline Density Estimation," R Package version 2.1.16, 2020. [Online]. Available: https://cran.r-project.org/web//packages/logspline/index.html