

## **A MESH SIZE BOOSTING ALGORITHM IN KERNEL DENSITY ESTIMATION**

**C.C. Ishiekwene, S.M. Ogbonmwan and J.E. Osemwenkhae**  
*Department of Mathematics,  
University of Benin, Benin City, Nigeria*

### **ABSTRACT**

*This paper proposes a new algorithm for boosting in kernel density estimation (KDE). This algorithm enjoys the property of a bias reduction technique like other existing boosting algorithms and also enjoys the property of less function evaluations when compared with other boosting schemes. Numerical examples are used and compared with existing algorithm and the findings are comparatively interesting.*

**Keywords:** *Boosting, kernel density estimation, bias reduction, boosting algorithm.*

### **INTRODUCTION**

Kernel density estimation (KDE) is a process of constructing a density function from a given set of data. The process depends on the choice of the smoothing parameter which is not easily found (Silverman, 1986; Jones and Signorini, 1997; Wand and Jones, 1995). As a result of this problem of finding a smoothing parameter, the idea of boosting in KDE is born.

The area of boosting is relatively new in kernel density estimation. It was first proposed by Schapire (1990). Other contributors are Freund (1995), Freund and Schapire (1996), Schapire and Singer (1999). Boosting is a means of improving the performance of a "weak learner" in the sense that given sufficient data, it would be

guaranteed to produce an error rate which is better than "random guessing".

Mazio and Taylor (2004) proposed an algorithm in which a kernel density classifier is boosted by suitably re-weighting the data. This weight placed on the kernel estimator is a ratio of the log function in which the denominator is a leave-one-out estimate of the density function. Also, Mazio and Taylor's (2004) work theoretically gave an explanation and finally showed how boosting is able to reduce the bias in the asymptotic mean integrated squared error (AMISE) of Silverman (1986).

In section 2, we shall see how the Mazio and Taylor's (2004) algorithm is a bias reduction technique. Section 3 shall show a proposed meshsize algorithm and its limitations. Numerical examples

are used to justify this algorithm in section 4 and compare the results with that of Mazio and Taylor (2004).

**Algorithm on boosting kernel density estimates and bias reduction**

Throughout this paper, we shall assume our data to be univariate. The algorithm of Mazio and Taylor (2004) is briefly summarized in algorithm 1.

**Algorithm 1**

- Step 1: Given  $\{x_i, i=1,2,\dots,n\}$ , initialize  $W_i=1/n$
- Step 2: Select  $h$  (the smoothing parameter).
- Step 3: For  $m =1, 2, \dots, M$ , obtain a weighted kernel estimate

$$\hat{f}_m(x) = \sum_{i=1}^n \frac{W_m(i)}{h} k\left(\frac{x-x_i}{h}\right) \tag{2.1}$$

where  $x$  can be any value within the range of the  $x_i$ 's,  $k$  is the kernel function and  $w$  is a weight function; and then update the weights according to

$$W_{m+1}(i) = W_m(i) + \log\left\{\frac{\hat{f}_m(x_i)}{\hat{f}_m^{(-1)}(x_i)}\right\} \tag{2.2}$$

Step 4: Provide output as

$$\prod_{m=1}^M \hat{f}_m(x) \text{ renormalized to integrate to unity}$$

For full implementation of this algorithm see Marzio and Taylor (2004).

**Boosting as a Bias Reduction in Kernel Density Estimation**

Suppose we want to estimate  $f(x)$  by a multiplicative estimate. We also suppose that we use only "weak" estimates which are such that  $h$  does not tend to zero as  $n \rightarrow \infty$ . Let us use a population version instead of sample in which our weak learner, for  $h > 0$  is given by

$$\hat{f}_m(x) = \int \frac{1}{h} W_m(y) K\left(\frac{x-y}{h}\right) f(y) dy \tag{2.3}$$

where  $W_i(y)$  is taken to be 1. We shall take our kernel function to be Gaussian (since all distributions tend to be normal as  $n$ , the sample size, becomes large through central limit theory (Towers, 2002). The first approximation in the Taylor's series, valid for  $h < 1$  provided that the derivatives of  $f(x)$  are properly behaved, is

$$\hat{f}_{(1)}(x) = f(x) + \frac{h^2 f''(x)}{2}$$

and so we observe the usual bias of order  $O(h^2)$  of Wand and Jones (1995). If we now let

$$W_2(x) = \hat{f}_{(1)}(x)^{-1}$$

$$\begin{aligned} \hat{f}_2(x) &= \int k(z) \left\{ f(x+zh) + h^2 \frac{f''(x+zh)}{2} + O(h^4) \right\}^{-1} f(x+zh) dz \\ &= 1 - \frac{h^2 f''(x)}{2f(x)} + O(h^4) \end{aligned} \tag{2.4}$$

This gives an overall estimator at the second step as

$$\begin{aligned} \hat{f}_{(1)}(x) \cdot \hat{f}_{(2)}(x) &= f(x) \left\{ 1 + h^2 \frac{f''(x)}{2f(x)} + O(h^4) \right\} \left\{ 1 - \frac{h^2 f''(x)}{2f(x)} + O(h^4) \right\} \\ &= f(x) + O(h^4) \end{aligned} \tag{2.5}$$

which is clearly of order four and so we can see a bias reduction from order two to order four.

**Proposed meshsize algorithm in boosting**

In this section, we shall see how the leave-one-out estimator of the (2.2) in the weight function can be replaced by a meshsize estimator due to the time complexity involved in the leave-one-out estimator. In the leave-one-out estimator, we require  $(n+(n-1)).n$  function evaluations of the density for each boosting step. Thus, we are using a meshsize in its place. The only limitation on this meshsize algorithm is that we must first determine the quantity  $1/nh$  so as to know what the meshsize that would be placed on the weight function of (2.2) would be. The need to use a meshsize in place of the leave-one-out lies on the

fact that boosting is like the steepest-descent algorithm in unconstrained optimization and thus a good substitute that approximates the leave-one-out estimate of the function (Duffy and Helmbold, 2000; Taha, 1971; Ratsch et al., 2000; Mannor et al., 2001).

The new meshsize algorithm is stated as:

**Algorithm 2**

Step 1: Given  $\{x_i, i=1,2,\dots,n\}$ , initialize  $W_1=1/n$ , initialize

Step 2: Select  $h$  (the smoothing parameter).

Step 3: For  $m=1, 2, \dots, M$ ,

i) Get

$$\hat{f}_m(x) = \sum_{i=1}^n \frac{W_m(i)}{h} k\left(\frac{x-x_i}{h}\right)$$

where  $x$  can be any value within the range of the  $x_j$  's,  $k$  is the kernel function and  $w$  is a weight function

ii) Update

$$W_{m+1}(i) = W_m(i) + mesh$$

Step 4: Provide output

$$\prod_{m=1}^M \hat{f}_m(x)$$

and normalize to integrate to unity

We can see that the weight function uses a meshsize instead of the leave-one-out log ratio function of Mazio and Taylor (2004).

The numerical verification of this algorithm would be seen in section 4.

**NUMERICAL RESULTS**

In this section, we shall use three sets of data (data 1 is the lifespan of a car battery in years, data 2 is the number of written words without mistakes in every 100 words by a set of students and data 3 is the scar length of patients randomly sampled. See Ishiekwene and Osemwenkhae, 2006) which were suspected to be normal to illustrate algorithms 1 and 2 and compare their results. BASIC programming language is used to generate the results for a specified  $h$  (Ishiekwene and Osemwenkhae (2006) and their corresponding graphs shown in figures 1a – 1c for data 1(n=40), figures 2a – 2c for data 2(n=64) while figures 3a – 3c is for data 3(n=110). Also, table 4.1 shows the bias reduction we discussed in section 2 for the three sets of data used.

We can clearly see the bias reduction between the normal kernel estimator and the boosted kernel estimator. We also compared our boosted kernel estimator with the fourth-order kernel estimator of Jones and Signorini (1997) which is arguably the best smoothing parameter choice and an equivalent of Ishiekwene and Osemwenkhae's (2006) higher-order choice at order four.

**CONCLUSION**

From the graphs in figures 1A – 3C for the three sets of data , we can clearly see that our proposed

**Table 4.1 Showing bias reduction**

	NORMAL (conventional order two)			H4 (as in Jones and Signorini 1997 and Ishiekwene and Osemwenkhae 2006)			BOOSTED (proposed scheme)		
	Bias2	Var	AMISE	Bias2	var	AMISE	Bias2	var	AMISE
Data 1	0.005276637	0.019811685	0.250883225	0.002071803	0.014789245	0.016861048	0.002078009	0.015168456	0.017246655
Data 2	0.000293946	0.001130402	0.001424348	0.000108591	0.000809307	0.000917898	0.000108715	0.000822153	0.000930868
Data 3	0.004697604	0.016623515	0.021321119	0.001767617	0.011342162	0.013109779	0.001768218	0.011446219	0.013214437



Fig. 1A (Leave-one out Algorithms)

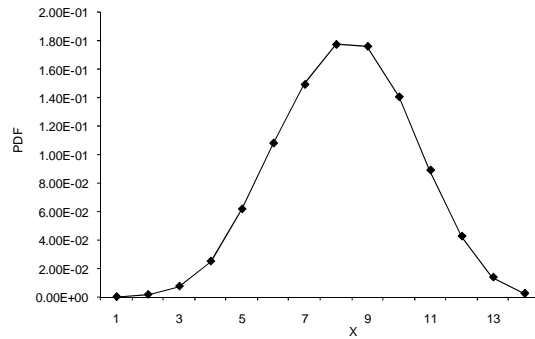


Fig. 2A (Leave-One-Out Algorithms)

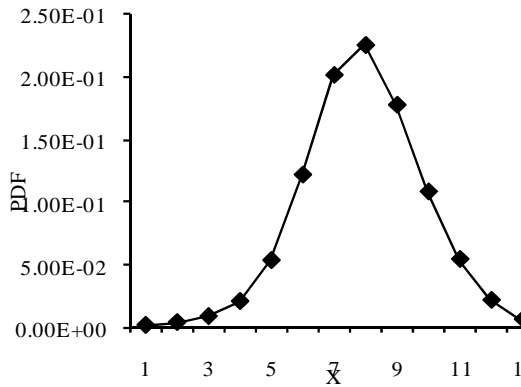


Fig. 1B (Meshsize Algorithms)

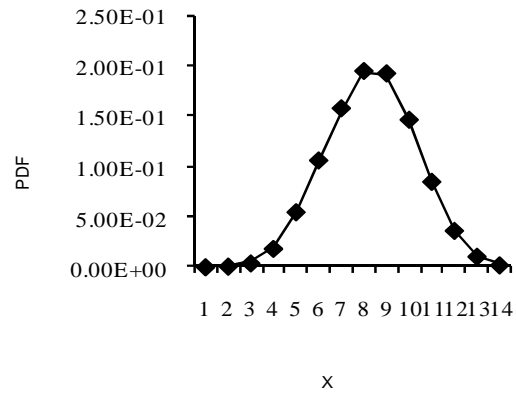


Fig. 2B (Meshsize Algorithms)

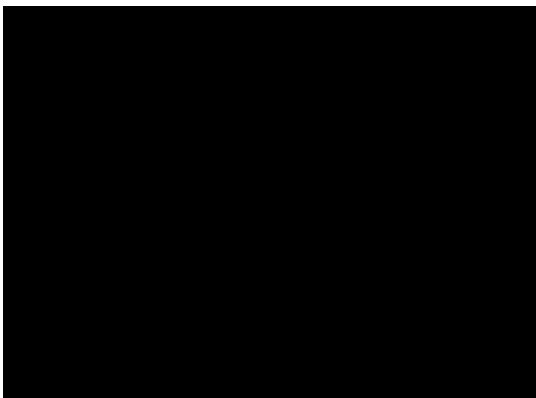


Fig. 1C (Comparison of both Algorithms)

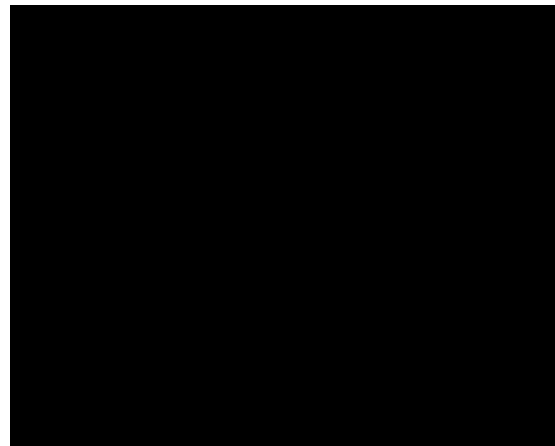


Fig. 2C; (Comparison of Both Algorithms)

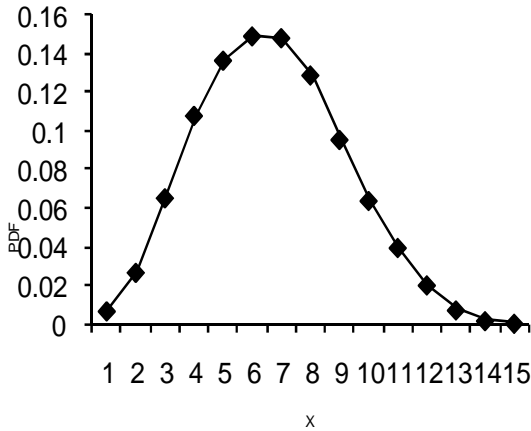


Fig. 3A (Leave-One-Out Algorithms)

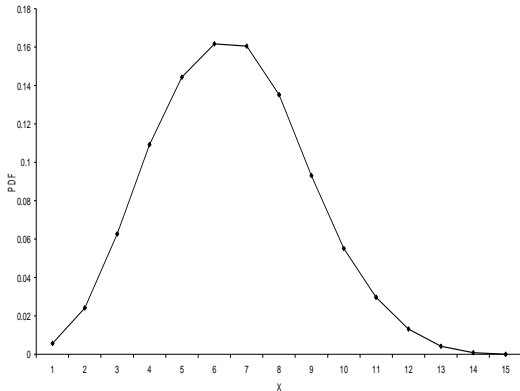


Fig. 3B (Meshsize Algorithms)

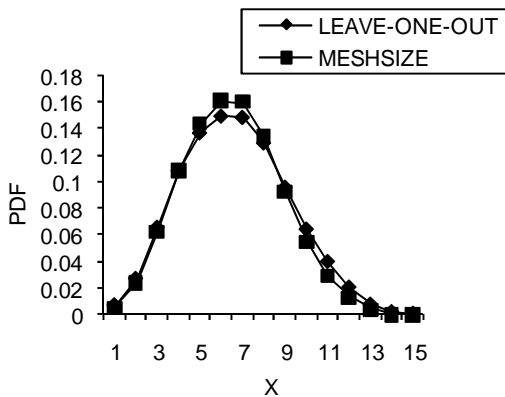


Fig. 3C (Comparison of both Algorithms)

algorithm compares favorably if not arguably better than that of Mazio and Taylor (2004). Also, the results of Table 4.1 has revealed the bias reduction targeted by both algorithms.

Since our algorithm does the same task with fewer function evaluations and in fact better in terms of time complexity, we are recommending it for use in boosting in KDE.

REFERENCES

Duffy, N. and Hembold, D. (2000). Potential boosters? *Advances in Neural info. Proc. Sys.* 12: 258 – 264.

Freund, Y. (1995). Boosting a weak learning algorithm by majority. *Info Comp.* 121: 256 - 285.

Freund, Y. and Schapire, R. (1996). Experiments with a new boosting algorithm. *In machine learning: proceedings of the 12<sup>th</sup> Intl. conf.* Ed. L. Saite 148 - 156.

Ishiekwene, C. C. and Osemwenkhae, J. E. (2006). A comparison of fourth order window width selectors in KDE (A Univariate case). *ABACUS* 33(2A): 14 – 20.

Jones, M.C. and Signorini, D.F. (1997). A comparison of Higher-order Bias Kernel Density Estimators. *JASA* 92: 1063 – 1072.

Mannor, S., Meir, R. and Mendelson, S. (2001). On consistency of boosting algorithms. (Unpublished Manuscript).

Mazio, D. M. and Taylor, C.C. (2004). Boosting Kernel Density estimates: A bias reduction technique? *Biometrika* 91: 226 - 233.

Ratsch, G. Mika, S., Scholkopf, B. and Muller, K. K. (2000). Construction boosting algorithms from SVM's: An application to one-class classification. *GMD Tech Report* No. 1.

Schapire, R. (1990). The strength of weak learnability. *Machine learn.* 5: 313 – 321.

Schapire, R. and Singer, Y. (1999). Improved boosting algorithm using confidence rated prediction. *Machine learn.* 37: 297 – 336.

- Silverman, B.W. (1986). Density Estimation for Statistics and Data analysis. Chapman and Hall, London; 34 – 74.
- Taha, H. A. (1971). Operations Research: An Introduction Prentice Hall, New Jersey; 781 – 814.
- Towers, S. (2002). Kernel Probability density estimation methods. In; M. Whalley, L. Lyons (Eds), Advanced Statistical techniques in particle Physics, Durham, England, 112 – 115.
- Wand, M.P. and Jones, M.C. (1995). Kernel Smoothing. Chapman and Hall/CRC, London; 10 – 88.
- Wenxin, J. (2001). Some theoretical aspects of boosting in the presence of noisy data. *Neural Computation* 14(10): 2415-2437.