Journal of the Nigerian Association of Mathematical Physics
***Volume* 18 (May, 2011), pp 363 – 366**
© J. of NAMP

Adaptive Kernel In The Bootstrap Boosting Algorithm In KDE

[1]Ishiekwene, C. C and [2]Nwelih, E
[1]Department of Mathematics, University of Benin, Nigeria.
[2]Department of Computer Science, University of Benin, Nigeria.

Abstract

This paper proposes the use of adaptive kernel in a bootstrap boosting algorithm in kernel density estimation. The algorithm is a bias reduction scheme like other existing schemes but uses adaptive kernel instead of the regular fixed kernels. An empirical study for this scheme is conducted and the findings are comparatively interesting.

Keywords: Boosting, kernel density estimates, bias reduction, adaptive kernel, bootstrap.

1. Introduction:

Boosting in kernel density estimation was first proposed by [11] and other authors like [3, 12] but to mention a few also made their contributions. Boosting is a means of improving the performance of a 'weak learner'. It is applied in this context using the adaptive kernel, it would not only guarantee an error rate better than random guessing but also deals with correction of 'noises' at the tails of the distribution or where we have sparse cluster of data within a given region.

In 2004, Mazio and Taylor 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 a log function in which the denominator is a leave-one-out estimate of the density function. A theoretical explanation is also given to show how boosting is a bias reduction technique i.e a reduction of the bias term in the expression for the asymptotic mean integrated squared error (AMISE).

2. Methodology

Algorithm on Boosting Kernel Density Estimates and Bias Reduction

In this paper, we shall assume the data to be univariate and the kernel function is the adaptive kernel.

Bootstrap Boosting Algorithm

We shall see how the leave-one-out estimator of [9] in the weight function can be replaced by a bootstrap 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 bootstrap in its place. The only limitation on this bootstrap algorithm is that we must first determine the number of bootstraps (B – usually large) samples to be taken before finding the weight function [7]. The need to use a bootstrap 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 [2, 4, 8, 10,13]. The method been proposed here differs from [7] because the kernel in question here is not fixed unlike that of [7].

The new bootstrap algorithm is stated as:

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

Step 2: Select h (the smoothing parameter)

Step 3: For m = 1,2,…,M;

(i) Get

$$\hat{f}(x) \overset{\Delta}{=} \sum_{i=1}^{n} \frac{w_m(i)}{h} k_A\left(\frac{x - X_i}{h}\right) \qquad (2.1)$$

(ii) Update

$$W_{m+1}(i) = W_m(i) + \text{Log}\left\{\frac{\hat{f}_m(x_i)}{f_m^{(B)}(x_i)}\right\} \qquad (2.2)$$

Corresponding authors: Ishiekwene, Tel. +2348023348430

where $f_m^{(B)}(x_i)$ is the bootstrap estimate of the density at point i.

Step 4: Provide output $\prod_{m=1}^{M} \hat{f}_m(x)$ renormalized to integrate to unity.

3. Results/Discussion

In this section, we shall use three sets of data to illustrate our algorithm and BASIC programming language is used. Data 1 is a sample of size forty and is the lifespan of car batteries in years. Data 2 is a sample of size sixty-four and is the number of written words without mistakes in every 100 words by a set of students in a written essay. Data 3 is the scar length of patients randomly selected in millimeters [5, 6].

The results are shown in Figures 3.1 – 3.3. Figure 3.1 is the graph for Data 1, Figure 3.2 for Data 2 while Figure 3.3 is for Data 3. In all three charts shown in Figures 3.1 – 3.3, the three kernel methods are plotted on the same sheet for easy comparison at a glance (ie the classical fixed kernel method ,the adaptive kernel method and the boosted kernel method). The boosted version is obtained using the Bootstrap Boosting algorithm of [7].

The results as shown in Figures 3.1 - 3.3 reveals that the classical fixed kernel density estimation method oversmooths the curves by obscuring some important features in the data. The adaptive kernel method showed a clearer picture of the nature of the data around the tails. The boosted kernel method was close to the adaptive kernel method in all three data used thus showing that this method is clearer than the classical fixed kernel method in terms of revealing data features. It does not only reveal features at the tails but is a bias reduction scheme as shown theoretically above and in Table 3.1[1].

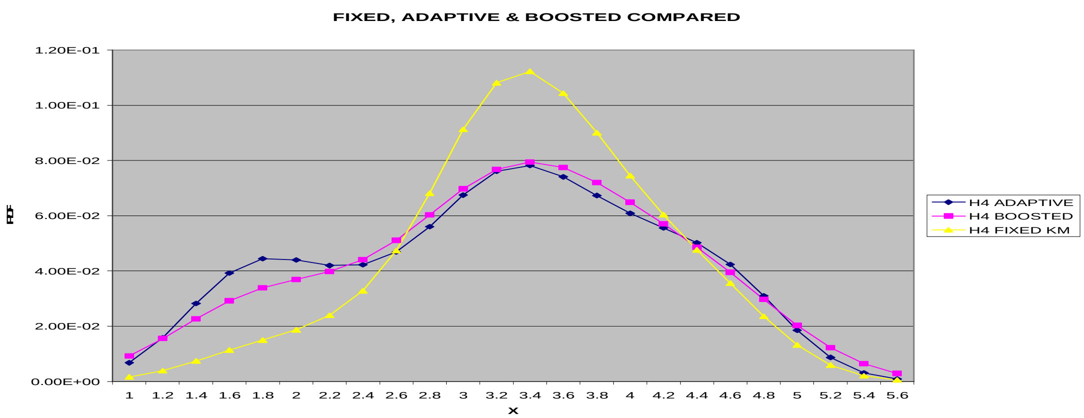

Fig 3.1. Chart showing the three techniques Using Data 1

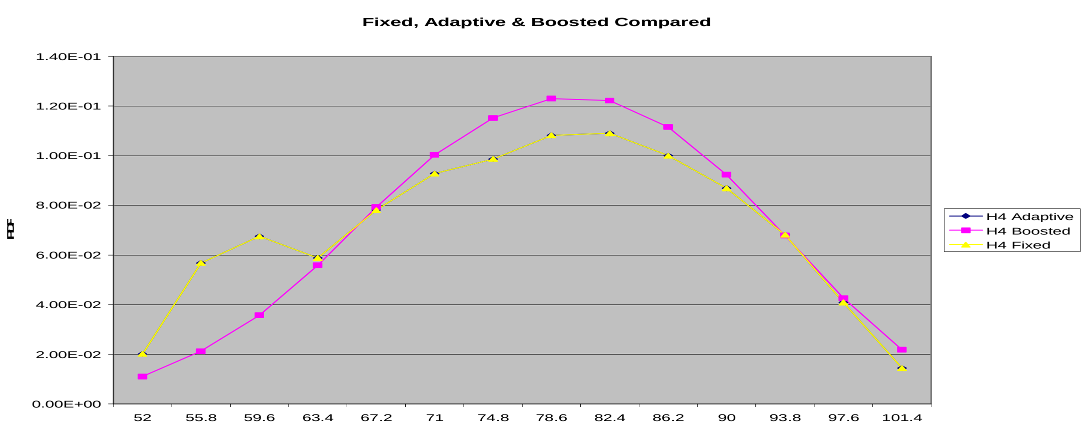

Fig 3.2. Chart showing the three techniques Using Data 2

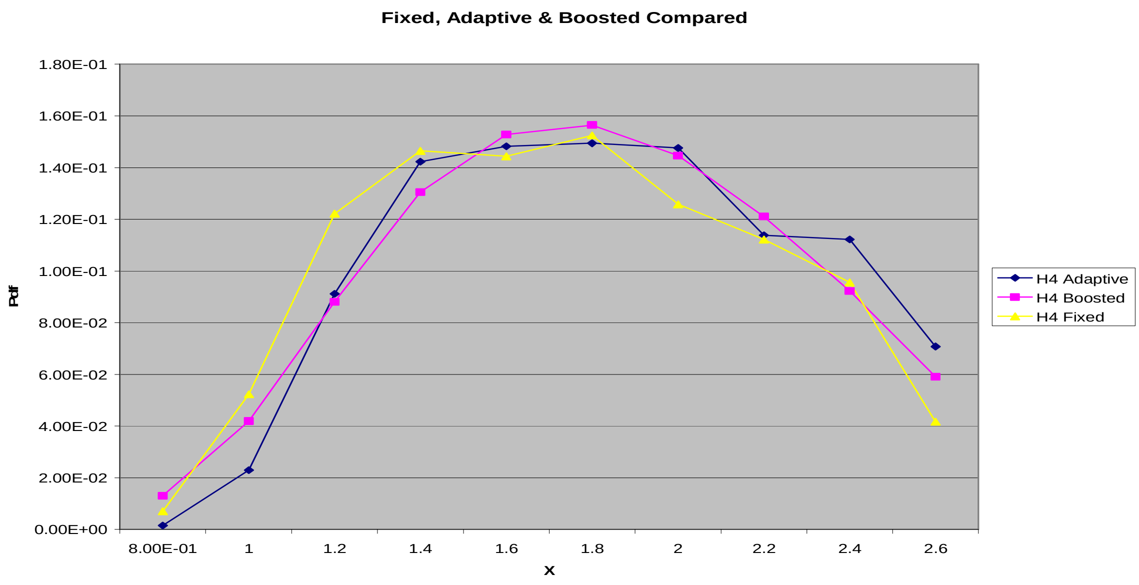

Fig 3.3. Chart showing the three techniques Using Data 3

4. Conclusion

We have shown that the adaptive kernel can be used in place of the classical fixed kernel in boosting in kernel density estimation. The charts- figs. 3.1 – 3.3 and table 3.1 clearly reveals that the adaptive kernel method does better than the classical fixed kernel method in kernel density estimation. It is therefore recommended for use in place of the classical fixed kernel method in boosting in KDE having exhibited the qualities of bias reduction and revealing the data features at the tails.

REFERENCES

[1] Birke, Melanie (2009) – Shape constrained KDE. Journal of Statistical planning & inference, vol **139,** issue 8 , August 2009, pg 2851 – 2862.

[2] Duffy, N. and Hemlbold, D. (2000) – Potential boosters? Advances in Neural info. Proc. Sys. **12**, 258 – 264.

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

[4] Hazelton, M.L & Turlach,B.A (2007) – Reweighted KDE. Computational Statistics & Data Analysis,Vol **51**,issue 6,March 2007,pg 3057 – 3069.

[5] Ishiekwene, C.C and Afere, B.A.E. (2001) – Higher Order Window Width Selectors for Empirical Data. Journal of Nigerian Statistical Association (J.N.S.A), **14**, 69 – 82.

[6] Ishiekwene, C. C. and Osemwenkhae, J. E. (2006) – A comparison of fourth order window width selectors in Kernel Density Estimation (A Univariate case), ABACUS, **33**; 14 - 20.

[7] Ishiekwene**,** C.C**;** Odiase, J. I and Ogbonmwan, S.M (2007) - Bootstrap in Boosting Kernel Density Estimates. Int'l Journal of Natural & Applied Sciences, **3** (4), 531 – 536.

[8] Mannor, S.; Meir, R. and Mendelson, S. (2001) – On consistency of boosting algorithms; submitted to Adv. In Neural info. Proc. Sys.

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

[10] 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.

[11] Schapire, R. (1990) – The strength of weak learnability. Machine learn. **5**,pg 313 – 321.

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

[13] Taha, H. A. (1971) – Operations Research: An Introduction. Prentice Hall, New Jersey.