HIGHER-ORDER GAUSSIAN KERNEL IN BOOTSTRAP BOOSTING ALGORITHM

Ishiekwene, C. C [1] **and Afere, B.A.E** [2]

[1.] Department of Mathematics, University of Benin, Nigeria
[2.] Department of Mathematics & Statistics, Federal Polytechnic Idah, Nigeria

ABSTRACT
The bootstrap boosting algorithm has been shown to be a bias reduction scheme. This paper proposes the use of higher-order Gaussian kernel in a bootstrap boosting algorithm in kernel density estimation. The algorithm uses the higher-order Gaussian kernel instead of the regular fixed kernels. An empirical study for this scheme is conducted and the findings are compared with existing fixed kernel methods.

Keywords: Boosting, kernel density estimates, bias reduction, higher-order Gaussian kernel, meshsize, noises

INTRODUCTION
Boosting is a means of improving the performance of a 'weak learner'. Boosting does not only guarantee an error rate that is better than random guessing but also deals with the correction of 'noises' at the tails of the distribution or where we have sparse cluster of data within a given region.
Boosting in kernel density estimation was first proposed by Schapire (1990). Other authors like Freund (1995), Schapire and Singer (1999) but to mention a few have also made contributions. It is applied in this context using the higher-order Gaussian kernel.

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 ie

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

in which the denominator is a leave-one-out estimate of the density function. A theoretical explanation is also given by Mazio and Taylor (2004) to show how boosting is a bias reduction technique. That is a reduction in the bias term of the expression for the asymptotic mean integrated squared error (AMISE) giving as $AMISE = \int Bias^2 + \int Var$. See Silverman (1986) for more details.

METHODS
We shall see how the leave-one-out estimator of Mazio and Taylor (2004) 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 bootstrap is a resampling scheme that estimates the function $\hat{f}^{(-1)}(x_i)$ I the weight function of Mazio and Taylor (2004) . The only limitation on this bootstrap algorithm is that we must first determine B- the number of bootstrap sample which must be large (Ishiekwene *et al*., 2008). The need to use a bootstrap in place of the leave-one-out lies on the fact that bootstrap is a resampling technique which finds the required estimates unlike the leave-one-out with so much function evaluation. The bootstrap is a good substitute that approximates the leave-one-out estimate of the function (Duffy and Helmbold, 2000; Ratsch *et al.*, 2000; Mannor *et al.*, 2001: Hazelton and Turlach, 2007).
The new bootstrap algorithm is stated as:

Bootstrap boosting Algorithm for Higher-order Gaussian Kernel

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 $m = 1,2,\ldots,M$

Get $\hat{f}_m(x) = \sum_1^n \frac{W_m(i)}{h} \frac{1}{2\sqrt{2\pi}} \left[3-(t-t_i)^2\right]$

exp- $\frac{(t-t_i)^2}{2}$

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

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

STEP 4: Provide output

$$\prod_1^M \hat{f}_m(x)$$

and normalized to integrate to unity.

We can see that the weight function uses a bootstrap instead of the leave-one-out log ratio function of Mazio and Taylor (2004). The kernel function used is the higher-order Gaussian kernel unlike the fixed used in Ishiekwene *et al* (2008). The idea of higher-order kernels via bias reduction dates back to Parzen (1962) and Bartlett (1963). Schucany and Summers (1997) also applied the generalized jackknife to bias reduction in kernel density estimation and showed that it is equivalent to using higher-order kernels (Birke, 2009). The numerical verification of this algorithm would be seen in the discussion.

DISCUSSION

In this section, we shall use three sets of data to illustrate our algorithm and BASIC programming language is used. Table1 is a sample of size forty and is the lifespan of car batteries in years. Table 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. Table 3 is the scar length of patients randomly selected in millimeters (Ishiekwene and Afere, 2001; Ishiekwene and Osemwenkhae, 2006).

The results are shown in figures 3.1a – 3.3b. Figure 3.1a is the graph for Table 1 showing the bias reduction, Figure 3.1b for Table 1 showing the MISE. Figure 3.2a is the graph for Table 2 showing the bias reduction, Figure 3.2b for Table 2 showing the MISE. Figure 3.3a is the graph for Table 1 showing the bias reduction, Figure 3.3b for Table 3 showing the MISE. In all three data sets used in this paper,

Table 3.1: Showing the various higher-order window widths, bias, variance and MISE for three data sets

m	n = 40				n = 64				n = 110			
	h_{opt}^m	$\int \text{Bias}^2 dx$	$\int \text{Var} dx$	MISE	h_{opt}^m	$\int \text{Bias}^2 dx$	$\int \text{Var} dx$	MISE	h_{opt}^m	$\int \text{Bias}^2 dx$	$\int \text{Var} dx$	MISE
2	0.508056	0.00273569	0.0234243	0.0261542	5.81862	0.00015979	0.00127832	0.00143811	0.233918	0.00231255	0.0185004	0.020813
4	0.511906	0.00213714	0.0232481	0.0253764	5.8684	0.000105623	0.00126747	0.0013731	0.240328	0.00150058	0.018007	0.0195076
6	0.523919	0.00141969	0.0227151	0.0241348	6.10333	0.0000761679	0.00121869	0.00129485	0.252412	0.00107156	0.017145	0.0182165
8	0.540169	0.00110159	0.0220318	0.0231333	6.32586	0.0000587908	0.00117582	0.00123461	0.263207	0.000822088	0.0164418	0.0172639
10	0.55443	0.000894378	0.0214651	0.0223595	6.51615	0.0000475616	0.00114148	0.00118904	0.272246	0.000662328	0.0158959	0.0165582
12	0.566618	0.000750119	0.0210033	0.0217535	6.67669	0.0000397868	0.00111403	0.00115382	0.279788	0.000552406	0.0154674	0.0160198
14	0.577034	0.000644507	0.0206242	0.0212687	6.8128	0.000034118	0.00109178	0.00112589	0.286139	0.000472628	0.0151241	0.0155967
16	0.585995	0.000564134	0.0203088	0.020873	6.92926	0.0000298174	0.00107343	0.00110324	0.291547	0.00041232	0.0148435	0.0152559
18	0.593773	0.00050107	0.0200428	0.0205439	7.02994	0.0000264513	0.00105805	0.0010845	0.296206	0.000365252	0.0146101	0.0149753
20	0.600582	0.000450354	0.0198156	0.0202659	7.1178	0.0000237498	0.00104499	0.00106874	0.30026	0.000327564	0.0144128	0.0147404

Table 1

2.2	3.4	2.5	3.3	4.7
4.1	1.6	4.3	3.1	3.8
3.5	3.1	3.4	3.7	3.2
4.5	3.3	3.6	4.4	2.6
3.2	3.8	2.9	3.2	3.9
3.7	3.1	3.3	4.1	3.0
3.0	4.7	3.9	1.9	4.2
2.6	3.7	3.1	3.4	3.5

Table 2

88	69	70	74	70	86	76	74
58	84	68	79	75	83	93	78
92	85	69	67	81	79	97	83
77	78	84	68	80	69	87	69
81	79	88	96	77	83	75	91
86	72	89	90	79	73	83	88
90	86	82	66	80	75	81	82
67	94	75	69	91	85	76	80

Table 3

1.2	2.0
1.4	2.3
2.6	1.4
2.0	1.8
1.4	1.64
1.7	2.0
1.6	2.3
1.5	1.2
1.48	1.3
1.6	1.9
2.2	2.0
1.35	2.4
1.35	2.0
1.2	2.6
1.6	1.3
1.2	1.7
1.6	1.6
1.2	1.5
2.0	1.9
1.4	2.4
1.7	2.1
1.6	2.3
2.0	1.8
2.4	1.4
1.8	1.9
1.6	2.0
1.64	1.3
1.3	1.9
2.0	1.42
1.9	1.47
1.4	1.4
2.0	1.9
1.4	2.0
1.7	2.0
1.9	2.4
1.6	1.9
2.0	2.0
2.4	2.4
1.8	2.0
1.6	1.98
1.64	2.2
1.3	1.6
1.4	2.4
2.4	2.6
1.6	2.0
2.4	1.6
2.0	1.7
1.4	1.9
1.6	2.2
1.8	1.86
1.2	1.4
2.0	1.9
2.2	1.7
1.8	1.6
1.9	2.3

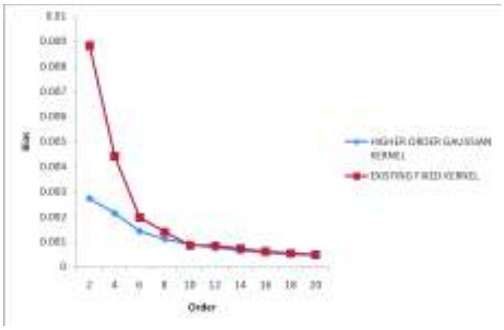

Fig 3.1a: Graph Showing the Bias for Table 1

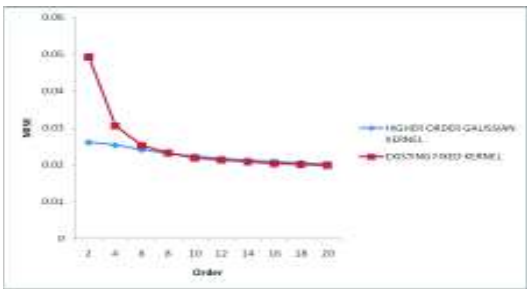

Fig 3.1b: Graph Showing the MISE for Table 1

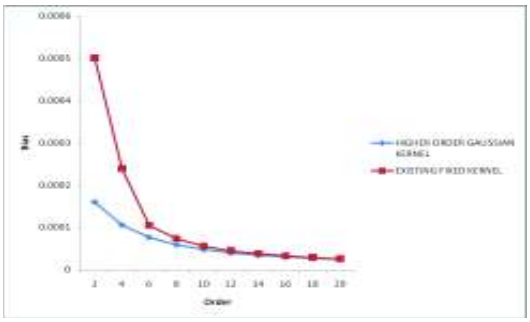

Fig 3.2a: Graph Showing the Bias for Table 2

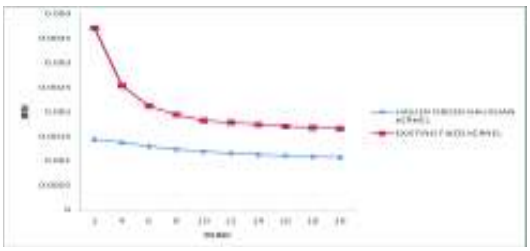

Fig 3.2b: Graph Showing the MISE for Table 2

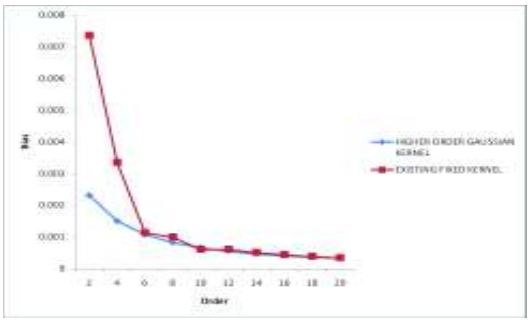

Fig 3.3a: Graph Showing the Bias for Table 3

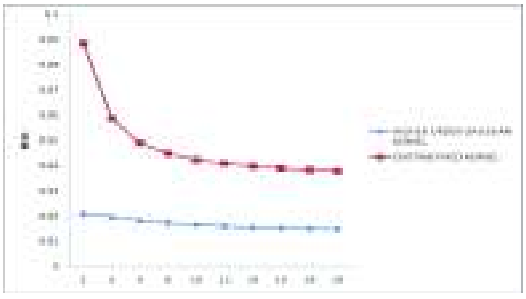

Fig 3.3b: Graph Showing the MISE for Table 3

we can clearly see the bias reduction which in turn translates to a reduction in the MISE. Table 3.1 shows the various window widths, bias^2,variance and the MISE for all three data sets (Ishiekwene and Nwelih, 2011).

CONCLUSION

We have shown that the higher-order Gaussian kernel can be used in place of the classical fixed kernel in boosting kernel density estimates. The charts- figs. 3.1a – 3.3b and table 3.1 clearly reveals that the higher-order Gaussian 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 which translates to a reduction in the overall MISE(ie bias^2 + var).

REFERENCES

Bartlett, M.S. (1963). Statistical estimation of density functions. Sankhyā series A, **25**: 245 – 254.

Birke, M. (2009). Shape constrained KDE. *Journal of Statistical Planning & Inference* **139 (8):** 2851 – 2862.

Duffy, N. and Hemlbold, D. (2000). Potential bosters? *Advances in Neural Information Processing Systems* **12**: 258 – 264.

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

Hazelton, M.L and Turlach, B.A (2007). Reweighted KDE. *Computational Statistics & Data Analysis* **5 (6):** 3057 – 3069.

Ishiekwene, C. C. and Osemwenkhae, J. E. (2006). A comparison of fourth order window width selectors in Kernel Density Estimation (A Univarate case), ABACUS, **33**: 14 - 20.

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

Ishiekwene, C.C., Ogbonmwan, S.M and Osemwenkhae, J.E. (2008). A Meshsize Boosting Algorithm in Kernel Density Estimation. *Journal of Science & Technology, Ghana.* **28 (2):** 69 – 74.

Ishiekwene, C.C. and Nwelih, E. (2011). Adaptive Kernel in Meshsize Boosting Algorithm in KDE. *African Research Review* **5(1):** 438-446.

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

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

Parzen, E. (1962). On the estimation of a probability function and the nodes. *Annals of Mathematical Statistics.* **33**: 1065 – 1076.

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 learning* **5**: 313 – 321.

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

Schucany, W.R. and Sommers, J.P. (1977). Improvement of kernel-type density estimators. *Journal of American Statistical Association* **72**: 420 -423.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London

Wand, M.P. and Jones, M.C. (1995). Kernel Smoothing. Chapman and Hall/CRC, London.