Statistics and Computing **15**: 113–123, 2005
© 2005 Springer Science + Business Media, Inc. *Manufactured in The Netherlands.*

Kernel density classification and boosting: an L_2 analysis

M. DI MARZIO* and C.C. TAYLOR†

**University of Chieti-Pescara*
dimarzio@dmqte.unich.it
†*University of Leeds*
c.c.taylor@leeds.ac.uk

Received October 2003 and accepted December 2004

Kernel density estimation is a commonly used approach to classification. However, most of the theoretical results for kernel methods apply to estimation *per se* and not necessarily to classification. In this paper we show that when estimating the difference between two densities, the optimal smoothing parameters are *increasing* functions of the sample size of the complementary group, and we provide a small simluation study which examines the relative performance of kernel density methods when the final goal is classification.

A relative newcomer to the classification portfolio is "boosting", and this paper proposes an algorithm for boosting kernel density classifiers. We note that boosting is closely linked to a previously proposed method of bias reduction in kernel density estimation and indicate how it will enjoy similar properties for classification. We show that boosting kernel classifiers reduces the bias whilst only slightly increasing the variance, with an overall reduction in error. Numerical examples and simulations are used to illustrate the findings, and we also suggest further areas of research.

Keywords: cross-validation, discrimination, nonparametric density estimation, simulation, smoothing

1. Introduction

Consider data $x_1, \ldots, x_n$, as a realization of a random sample, and let an element of the set $\{f_j(x), j = 1, \ldots, J\}$ be the density associated with x_i. Let $\pi_j, j = 1, \ldots, J$ be the classes' prior probabilities, *i.e.* $\pi_j = P\left(x_i \in \Pi_j\right)$ where Π_j denotes the jth class. Then, using Bayes' Theorem, the posterior probability of the observation x_i being from the jth class, is:

$$P(x_i \in \Pi_j \mid x_i = x) = \frac{\pi_j f_j(x)}{\sum_{j=1}^{J} \pi_j f_j(x)}.$$

According to Bayes' rule, we allocate an observation to the class with highest posterior probability. Usually the values $\pi_j, j = 1, \ldots, J$ are estimated via the respective sample relative frequency, $\hat{\pi}_j = n_j/n$ with $\sum_j n_j = n$, associated with each class. As a consequence, the discrimination problem is essentially that of (jointly) estimating the probability density functions $f_j(x), j = 1, \ldots, J$.

There is a wide variety of approaches to discrimination, from parametric, normal-theory based linear and quadratic discrimination to neural networks; see Hastie *et al.* (2001). A flexible method uses kernel density estimation of $f_j(x)$ (Hand 1982). Given a random sample $X_1, \ldots, X_n$ from an unknown density f, the kernel density estimator of f at the point $x \in \mathbb{R}$ is (see, for example, Wand and Jones 1995, Ch. 4):

$$\hat{f}(x; h) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - X_i) \tag{1}$$

where h is a bandwidth or smoothing parameter, $K_h(x) = \frac{1}{h} K(\frac{x}{h})$, and the function $K : \mathbb{R} \to \mathbb{R}$, called a *kth-order* kernel, satisfies the following conditions: $\int K = 1$ and $\int x^j K \neq 0, \infty$ only for $j \geq k$.

The use of plain kernel density estimators has been shown to work well in a wide variety of real-world discrimination problems (see Habbema *et al.* 1974, Michie *et al.* 1994, Hall *et al.* 1995, Wright *et al.* 1995). Nevertheless, we note that in kernel-based classification problems we are not primarily interested in density estimation *per se*, but as a route to classification. We believe that the methodological impact of this different perspective

0960-3174 © 2005 Springer Science + Business Media, Inc.

has not yet been fully explored, although there are a few contributions; see, for example, Hall and Wand (1988).

It is worth considering the extent to which we should adapt the standard methodology of density estimation when applied to discrimination problems. An obvious difference is that density estimation usually considers Mean Integrated Squared Error, denoted as

$$\mathrm{MISE}(\hat{f}) = E\int (f(x) - \hat{f}(x))^2\,dx,$$

as a measure of the estimate's accuracy, whereas classification problems are more likely to use expected error rates. For example, many researchers avoid using higher-order kernels in density estimation because: the estimate is not itself a density; and, for moderate sample sizes, there is not much gain. However, for some classification problems, at least, the first reason may not be an obstacle.

In this paper we focus on the univariate case with two classes, i.e. $J = 2$; some multivariate extensions are contained in Di Marzio and Taylor (2004b). The information at hand is given in the bivariate dataset (x_i, Y_i), $i = 1, \ldots, n$ It will often be convenient to relabel the two classes 1, 2 as -1, 1 and in this case $Y_i : x_i \to \{-1, 1\}$ is an indicator of class membership. Our goal is to define a mapping $\delta : \mathbb{R} \to \{-1, 1\}$, called a classification rule. If $j \in \{-1, 1\}$, the point $x \in \Pi_j$ will be correctly classified if $\delta(x) = j$, misclassified if $\delta(x) \neq j$. If Π_1 and Π_2 are connected sets, then we all we require is an estimate of x_0 such that:

$$\delta(x) = \begin{cases} -1 & \text{if } x < x_0 \\ 1 & \text{otherwise.} \end{cases}$$

We use the above framework for the sake of simplicity, but note it can be early generalized if $J > 2$ or more complicated partitions of $\mathbb{R}$ occur. Extending some of the methods to higher dimensions is also straightforward.

Machine learning deals with automatic methods that, once trained on the basis of available data, are able to make predictions or classifications about new data. This subject, originating from artificial intelligence and engineering, has many intersections with statistics. Thus, in the last decade, it has gained a large amount of popularity among statisticians. Nowadays, many prominent researchers incorporate Machine Learning, several traditional statistical techniques related to classical regression and classification, and new computational procedures, into a superset known as *statistical learning*. Hastie *et al.* (2001) go deeply into this taxonomy. *Boosting* is a learning technique that has recently received a great deal of attention from statisticians; see Friedman *et al.* (2000), Friedman (2001) and Bühlmann and Yu (2003).

Di Marzio and Taylor (2004b) have shown that boosting kernel classifiers can lead to a reduction in error rates for some real multivariate datasets. The main result of this paper is to explain *why* boosting kernel classifiers should be so successful. We firstly discuss some theory on bandwidth selection for standard kernel classification, and then propose a suitable implementation of boosting for the discrimination problem. We show that boosting is effective through an L_2 view of estimation in a neighbourhood of x_0.

This paper is organized as follows. Section 2 analyzes the standard case of kernel discrimination and deals with the *joint* selection of the smoothing parameters. Section 3 introduces boosting and considers how it may be adapted for use with kernel density discrimination. Section 4 makes a connection between boosting and a multiplicative bias reduction technique previously proposed in kernel density estimation, and we independently indicate why boosting should reduce the bias in kernel discrimination. In Section 5 we give some simulation and experimental results which illustrate the theory, make comparisons of boosting with simple kernel methods, and investigate the role of some of the parameter selections. A final section contains some concluding remarks, as well as a range of outstanding issues which may inform future research.

2. Estimating the difference between two densities

In this section we consider the goal of estimating a difference between two densities, say $g(x) = f_2(x) - f_1(x)$. In the case that $\pi_1 = \pi_2$, this would then lead to the classifier given by $\delta(x) = \operatorname{sign}\hat{g}(x)$. The reason for considering this is that it is similar to previously adopted implementations of kernel discrimination, and our objective is to indicate the effect on the choice of smoothing parameters when we estimate the *difference* between two densities.

2.1. *A L_2 risk function*

We are interested in solutions to $g(x) = 0$ given by x_0 such that $f_1(x_0) = f_2(x_0) = f(x_0)$, say. For simplicity here we suppose that $\pi_1 = \pi_2 = 1/2$, but we do not require equal sample sizes. Suppose the same kernel function K is used to estimate both f_1 and f_2; moreover let these standard assumptions hold (see, for example, Wand and Jones 1995, pp. 19–20):

(i) f_j'' is continuous and monotone in $(-\infty, -M) \cup (M, \infty)$, $M \in \mathbb{R}$; $\int (f_j'')^2 < \infty$;
(ii) $\lim_{n\to\infty} h = 0$ and $\lim_{n\to\infty} nh = \infty$;
(iii) K is bounded and $K(x) = K(-x)$.

Starting from the usual theory (see Wand and Jones 1995, p. 97), we obtain

$$\mathrm{E}\hat{g}(x) = f_2(x) - f_1(x) + \mu_2(K)\left(\frac{h_2^2}{2}f_2''(x) - \frac{h_1^2}{2}f_1''(x)\right) + o\left(h_1^2 + h_2^2\right)$$

and

$$\mathrm{Var}\,\hat{g}(x) = R(K)\sum_{j=1}^{2}\frac{f_j(x)}{n_j h_j} + o\left\{\sum_{j=1}^{2}(n_j h_j)^{-1}\right\}$$

where, for a real valued function t, $R(t) = \int t(x)^2\,dx$, $\mu_k(t) = \int x^k t(x)\,dx$, and h_i is the smoothing parameter used in the estimation of $f_i(x)$. Hence the mean squared error (MSE) of our estimate of the point x_0 such that $g(x_0) = 0$, is:

$$\text{MSE}\,\{\hat{g}(x_0)\} = \text{AMSE}\,\{\hat{g}(x_0)\} + o\left\{\sum_{j=1}^{2} h_j^4 + (n_j h_j)^{-1}\right\}$$

where

$$\text{AMSE}\,\{\hat{g}(x_0)\} = \mu_2(K)^2 \left\{\frac{h_2^2}{2} f_2''(x_0) - \frac{h_1^2}{2} f_1''(x_0)\right\}^2 + R(K)\sum_{j=1}^{2} \frac{f_j(x_0)}{n_j h_j} \tag{2}$$

is the asymptotic MSE, the usual large sample approximation consisting of the leading term in the expanded MSE. By integrating the pointwise measure in equation (2) we obtain a global measure, the asymptotic integrated mean squared error:

$$\text{AMISE}\,\{\hat{g}(\cdot)\} = \mu_2(K)^2\, R\left(\frac{h_2^2}{2} f_2'' - \frac{h_1^2}{2} f_1''\right) + R(K)\sum_{j=1}^{2} (n_j h_j)^{-1}. \tag{3}$$

2.2. *Pointwise estimation*

If we differentiate equation (2) with respect to h_i, $i = 1, 2$ and equate to zero we can solve to obtain:

$$h_1^5 = f(x_0)\big/\big(N_1 f_1''(x_0)^2 - (N_1 f_1''(x_0))^{5/3} N_2^{-2/3} f_2''(x_0)^{1/3}\big) \tag{4}$$

$$h_2^5 = f(x_0)\big/\big(N_2 f_2''(x_0)^2 - (N_2 f_2''(x_0))^{5/3} N_1^{-2/3} f_1''(x_0)^{1/3}\big) \tag{5}$$

where $N_j = n_j \mu_2^2(K)/R(K)$. The solution for one of the h_js will be negative in the case that $f_1''(x_0) f_2''(x_0) > 0$; this may give insight into a similar phenomenon noted by Hall and Wand (1988). In this case we can reduce the bias by taking a larger h_j and the asymptotic solution which minimizes the mean-squared error will need to use the next term ($O(h^4)$) in the Taylor series expansion.

Note that each h_j, $j = 1, 2$ depends on *both* sample sizes n_1 and n_2, as well as *both* densities and that they have the following relationship:

$$h_1 = h_2\left(\frac{-n_2 f_2''(x_0)}{n_1 f_1''(x_0)}\right)^{1/3} \tag{6}$$

Note that, by inspecting the second term in the denominator of equation (4), when n_1 is fixed we find h_1 *increases* with n_2, i.e. when n_1 is fixed and $n_2 \to \infty$, h_1 increases to $h_1 = \{f(x_0)/(N_1 f_1''(x_0)^2\}^{1/5}$, which is the usual asymptotic formula for a single sample. That the optimal smoothing parameters are *increasing* functions of the sample size of the complementary group may seem counter-intuitive at first, but it happens in this case because the sign of the bias is related to the sign of $f''(x_0)$.

2.3. *Global estimation*

If we use a Normal kernel and a Normal plug-in rule for separate estimation to minimize integrated mean squared error, then $h_j^5 = 4\sigma_j^5/(3n_j)$, $j = 1, 2$; see, for example, Silverman (1986, p. 45). Differentiating equation (3), we thus obtain the equations:

$$\frac{3h_1^5 n_1}{4\sigma_1^5} - 2h_1^3 h_2^2 n_1 \gamma - 1 = 0 \tag{7}$$

$$\frac{3h_2^5 n_2}{4\sigma_2^5} - 2h_1^2 h_2^3 n_2 \gamma - 1 = 0 \tag{8}$$

where

$$\gamma = \frac{D^2 + 3V^2 - 6DV}{(2V^9)^{1/2}} \exp\left(-\frac{D}{2V}\right),$$

with $D = (\mu_1 - \mu_2)^2$ and $V = \sigma_1^2 + \sigma_2^2$.

Although it is possible to find numerical solutions of equations (7) and (8) for h_i, $i = 1, 2$, we have been unable to obtain a simple closed form. So we now give an approximate solution which gives an indication of the difference of global *joint* estimation. As a first approximation for our joint estimation, let $h_j^5 = 4\sigma_j^5(1 + \alpha_j)/(3n_j)$, $j = 1, 2$. Expanding the resulting equations in a Taylor series in α_j and considering only first order terms we then have an approximate solution given by:

$$\alpha_1 = -\frac{\beta_2\big(\beta_1 - 15n_1^{2/5}\big)}{\big(\beta_1 - 9n_1^{2/5}\big)\big(\beta_2 - 9n_2^{2/5}\big) - 36n_1^{2/5}n_2^{2/5}} \tag{9}$$

$$\alpha_2 = -\frac{\beta_1\big(\beta_2 - 15n_2^{2/5}\big)}{\big(\beta_1 - 9n_1^{2/5}\big)\big(\beta_2 - 9n_2^{2/5}\big) - 36n_1^{2/5}n_2^{2/5}} \tag{10}$$

where $\beta_1 = 8\gamma\sigma_1^2\sigma_2^3 n_2^{2/5}$ and $\beta_2 = 8\gamma\sigma_1^3\sigma_2^2 n_1^{2/5}$. Given two samples, it would be quite straightforward to calculate the sample mean and variances and use the above equations (9) and (10) to derive a plug-in rule more suited to discrimination problems. We note that these adjustments (α_j) do not tend to zero as the sample sizes tend to ∞; in fact, if $n_1 = n_2$ then α_j do not depend on the sample size. The largest magnitude of α_j for the case $n_1 = n_2$ is when $D = V(5 - 10^{1/2}) = 1.838$ V and the ratio $\max \sigma_j / \min \sigma_j \approx 1.324$. The corresponding smoothing parameters will differ from the independent case by about 8–10%.

We conclude this section noting that being able to evaluate the bias and variance of $\hat{g}(x)$ near the solution $g(x_0) = 0$ is not the final goal. All these calculations deal with a vertical discrepancy rather than a *horizontal* discrepancy between x_0 and an estimator of it, say $\hat{x}_0$, i.e.: $\hat{x}_0 - x_0$. However, in a small simulation study we did find that the joint pointwise selection given in equations (4) and (5) were close to the pair (h_1, h_2) which minimized the

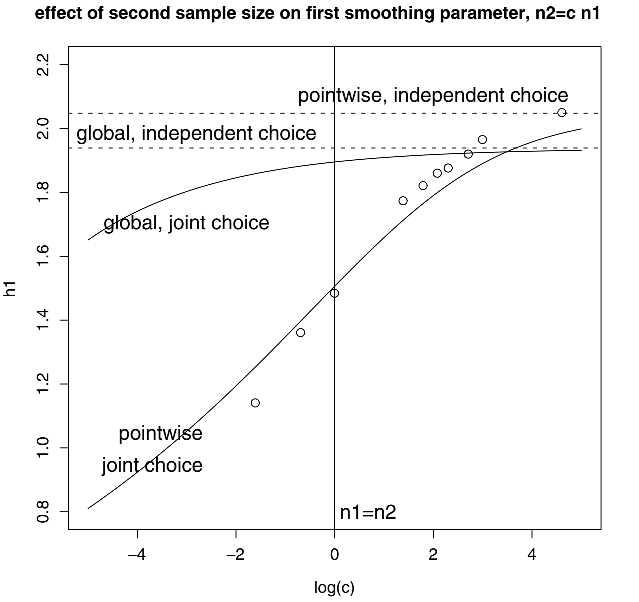


Fig. 1. *For $n_1 = 50$, and samples from $N(0,4^2)$, $N(4,1)$ the optimal h_1 (using the asymptotic equations) is shown for various criteria, as a function of the sample size $n_2 = cn_1$. The points show the values of h_1 to minimize (over pairs h_1, h_2) the average (over 20,000 simulations) squared error $(\hat{f}_1(x_0) - \hat{f}_2(x_0))^2$, where $x_0 = 2.243$*

misclassification rate. See Fig. 1 for a related example which illustrates the sample size dependence in equation (6), and the solutions to (7) and (8).

3. A boosting algorithm for kernel density discrimination

A boosting algorithm (Shapire 1990) repeatedly calls a "weak learner", which is essentially a crude classification method, M times to iteratively classify re-weighted data. The first weighting distribution is uniform, i.e. $w_1(i) = 1/n, i = 1, \dots, n$, whilst the mth distribution $\{w_m(i), i = 1, \dots, n\}$ with $m \in [2, \dots, M]$ is determined on the basis of the classification rule, say $\delta_{m-1}(x_i)$, resulting from the $(m-1)$-th call. The final sequence of decision rules, $\delta_m(x), m = 1, \dots, M$ is summarized into a single prediction rule which should have superior standards of accuracy.

The weighting distribution is designed to associate more importance to currently misclassified data through some *loss function*. Consequently, as the number of iterations increases the 'hard to classify' observations receive an increasing weight. Moreover, a simple majority vote criterion (Freund 1995), such as the sign of $\sum_{m=1}^{M} \delta_m(x)$, is commonly used to combine the 'weak' outputs. Finally, we note that, at present, there is no consolidated theory about a stopping rule, i.e. the value of M. This does not seem a particularly serious drawback because boosting is often characterized by some correlation between the training and test error.

Evidently, designing a *boosted* kernel classifier algorithm involves two main choices: (i) the weighting strategy, i.e. the way to 'give importance' to misclassified data; (ii) the version of boosting. Other issues, which will affect the accuracy, are: the existence of a kernel estimator and/or a bandwidth selector that are specifically suitable for boosting.

Concerning the weighting strategy, due to its nonparametric nature, kernel discrimination lends itself to several solutions. Two obvious criteria are: (i) locally adapting the bandwidths; and (ii) locally adapting the mass of the kernels by associating a weight to each observation. These correspond to undersmoothing and increasing the probability mass of kernels, respectively, for misclassified data.

A practical consideration can be helpful. Undersmoothing has the tendency to generate artificially numerous partitions of the feature space, especially if, as it usually happens, the data are sparse; in this case further investigation, to define an *ad hoc* bandwidth selector, is needed. Instead, varying the mass of the kernel seems a directly applicable solution. In this case, the traditional kernel estimator, that gives all observations the same mass, corresponds to the weak learner for $m = 1$.

Concerning an appropriate choice of boosting, we note that initial implementations of boosting used discrete decision rules, in our case: $\delta_m(x) : \mathbb{R} \to \{-1, 1\}$ (Shapire 1990, Freund and Shapire 1996), whilst recently Shapire and Singer (1998) and Friedman *et al.* (2000) suggest more efficient continuous mappings. In particular, Friedman *et al.* (2000) propose Real AdaBoosting in which the weak classifier yields membership probabilities, in our case $\delta_m(x) \propto p_m(x \in \Pi_j) = \hat{f}_{2,m}(x)/\{\hat{f}_{1,m}(x) + \hat{f}_{2,m}(x)\}$, for a fixed class Π_j. Its loss system gives x_i a weight proportional to

$$V_i = \left\{ \frac{\min\left(p(x_i \in \Pi_1), p(x_i \in \Pi_2)\right)}{\max\left(p(x_i \in \Pi_1), p(x_i \in \Pi_2)\right)} \right\}^{1/2}$$

if x_i is correctly classified, and proportional to V_i^{-1} if x_i is misclassified. Besides, it is to be noted that a continuous strong hypothesis is generated, preserving the analytical advantages of a kernel density estimate. Because kernel methods estimate densities in order to classify, Real AdaBoost seems the natural framework for boosting kernel discrimination , whereas discrete mappings do not employ the whole information generated by a kernel discrimination, but only the resulting sign.

Our pseudocode for Real AdaBoost kernel discrimination (*BoostKDC*) is given in Algorithm 1.

Algorithm 1 *BoostKDC*

```
1. Given $\{(x_i, Y_i), i = 1, \dots, n\}$, initialize $w_1(i) = 1/n, i = 1, \dots, n$.
2. Select $h_j, j = 1, 2$.
3. For $m = 1, \dots, M$ (the number of boosting iterations)
   (i) Obtain a weighted kernel estimate using
```

$$\hat{f}_{j,m}(x) = \sum_{i:Y_i=j} \frac{w_m(i)}{h_j} K\left(\frac{x - x_i}{h_j}\right) \qquad for j = 1, 2.$$

(ii) *Calculate*

$$\delta_m(x) = \frac{1}{2}\log\{p_m(x)/(1-p_m(x))\}.$$

where $p_m(x) = \hat{f}_{2,m}(x)/(\hat{f}_{1,m}(x) + \hat{f}_{2,m}(x))$

(iii) *Update:*

$$w_{m+1}(i) = w_m(i) \times \begin{cases} \exp(\delta_m(x_i)) & \textit{if} \quad Y_i = 1 \\ \exp(-\delta_m(x_i)) & \textit{if} \quad Y_i = 2 \end{cases}$$

4. *Output*

$$H(x) = \text{sign}\left\{\sum_{m=1}^{M} \delta_m(x)\right\}$$

Note that $\hat{f}_{j,m}(x)$ does not integrate to 1 even for $m = 1$; so in effect we are considering $\pi_j f_j(x)$, with $\pi_j = n_j/n$, in our estimation. Note also that we do not need to renormalize the weights because we consider the ratio $\hat{f}_{2,m}(x)/\hat{f}_{1,m}(x)$ so any normalization constant will cancel.

Considering the accuracy of the method we need to explore the *overfitting* phenomenon in boosting. A weak learner overfits data when it concentrates too much on a few misclassified observations, i.e. heavily bases the fitting on them, being unable to correctly classify them. Thus, after a value M^*, consecutive overfitted decision rules $\delta_{M^*+1}(x), \delta_{M^*+2}(x)\ldots$ can worsen the performance of the final classifier. A simple and general approach to prevent overfitting is cross-validation: M^* is estimated by observing the corresponding loss function when the boosting algorithm is carried out on a subsample.

However, if a flexible base learner is employed, we would expect small values of M^*. An illuminating description of this phenomenon is provided by Ridgeway (2000): on a dataset where a 'stump' works reasonably well, a more complex tree with four terminal nodes overfits from $M = 2$. Here the decision boundary is efficiently estimated in the first step, the other steps can only overfit misclassified data without varying the estimated boundary, so degrading the general performance. In order to reduce the risk of overfitting, a low variance base learner is suggested, so

> *…Each stage makes a small, low variance step, sequentially chipping away at the bias.*

Obviously a kernel discrimination is a flexible base learner, whatever its formulation is. Then, in a first approximation we can adopt the criterion suggested by Ridgeway (2000) by significantly oversmoothing, using as a bandwidth a multiple of the optimal value as obtained from classical methods.

Another regularization strategy, adopted to restrict the variance inflation due to high values of M, is to reduce the contribution of $\delta_m(x)$ to $H(x)$. This philosophy is proposed by Friedman (2001) for a different boosting algorithm where the contribution of each step is reduced by 94%. Observing experimental evidence, he finds an inverse relation between M^* and the 'Learning Rate Parameter' (LRP), and suggests a very low LRP and a very high M. Friedman can't justify the good practical performances of this strategy, considering the phenomenon to be 'mysterious'. In Real AdaBoost we can follow this approach identifying as LRP the exponent of the probabilities ratio in the loss function. Then, a strategy could be to replace the value $1/2$ in step 3(ii) by a value $1/T$ with $T > 2$. For larger values of T the less aggressive will be the algorithm, becoming similar to discrete AdaBoost as $T \to \infty$.

4. The first boosting step ($m = 2$)

In this section we firstly point out an an interesting link between boosting kernel discrimination and previous work on bias reduction in density estimation. This work was totally independent of the boosting paradigm. Then we derive the bias of the difference estimator $\hat{g}(x) = \hat{f}_1(x) - \hat{f}_2(x)$, involved in $H(x)$, at the point x_0 such that $f_1(x_0) = f_2(x_0) = f(x_0)$, and show that while it is initially $O(h^2)$- biased (standard kernel method), boosting reduces the bias to $O(h^4)$ in the special case when $h_1 = h_2$.

4.1. *Relationship to previous work*

The final classifier output by Algorithm 1 is of the form

$$H(x) = \text{sign}\left\{\sum_{m=1}^{M} \delta_m(x)\right\} = \text{sign}\left[\sum_{m=1}^{M} \frac{1}{2}\log\left\{\frac{\hat{f}_{2,m}(x)}{\hat{f}_{1,m}(x)}\right\}\right].$$

For $M = 2$ we see the decision boundary is defined by points x such that

$$\sum_{m=1}^{2} \delta_m(x) = 0$$

which is equivalent to

$$\tilde{f}_1(x)\sum w_1 K\left(\frac{x-x_i}{h_1}\right) = \tilde{f}_2(x)\sum w_2 K\left(\frac{x-x_i}{h_2}\right)$$

where

$$w_1 = \left(\frac{\tilde{f}_2(x_i)}{\tilde{f}_1(x_i)}\right)^{1/2} \qquad w_2 = \left(\frac{\tilde{f}_1(x_i)}{\tilde{f}_2(x_i)}\right)^{1/2}$$

and $\tilde{f}_j$, $j = 1, 2$ are the initial density estimates for the two groups. Thus the classification boundary can be seen as the intersection points of two multiplicative kernel estimators.

Note that this is very similar to the variable-kernel density estimator of Jones *et al.* (1995):

$$\hat{f}(x) = \hat{f}_b(x)\frac{1}{n}\sum_{i=1}^{n}\hat{f}_b^{-1}(x_i)\frac{1}{h}K\left(\frac{x-x_i}{h}\right), \qquad (11)$$

where $\hat{f}_b$ is the classical estimator with the bandwidth b. We can see that equation (11) is simply the product of an initial estimate, and a (re-)weighted kernel estimate, with the weights depending on the first estimate. This is of the same form as the boosted classifier at $m = 2$. The idea behind (11) is that the leading bias in $\hat{f}_b(x)$ should cancel with the leading bias in $\hat{f}(x_i)$ and their paper showed that this was an effective method of nonparametric density estimation. In its simplest form, $b = h$. A recent semiparametric modification of this method was proposed by Jones *et al.* (1999).

Di Marzio and Taylor (2004a) showed that kernel density *estimates* could be directly boosted by defining a loss function in terms of a leave-one-out estimate (see Silverman 1986, p. 49), and they established a link between this version of boosting and the bias-reduction technique of Jones *et al.* (1995).

4.2. *Boosting reduces the bias*

In order to gain some insights into the behaviour of boosting we consider a population version: this corresponds to the situation in which there is an infinite amount of data, but the smoothing parameter is bounded away from 0. We examine the weights and classifiers for learners which are "weak" in the sense that our estimate of $f(x)$ is given by:

$$\hat{f}_{j,m}(x) \propto \int \frac{1}{h_j} K\left(\frac{x-y}{h_j}\right) w_{j,m}(y) f_j(y)\, dy \quad \text{for } j = 1, 2.$$

The first approximation in the Taylor series expansion (which we use for the initial estimate, when $m = 1$) is

$$\hat{f}(x) = f(x) + h^2 f''(x)/2 \tag{12}$$

for some $h > 0$. So the initial classifier then uses

$$\begin{aligned}\delta_1(x) &\propto \frac{1}{2}\{\log \hat{f}_{2,1}(x) - \log \hat{f}_{1,1}(x)\}\\ &= \frac{1}{2}\left[\log\left\{\frac{f_2(x)}{f_1(x)}\right\} + \frac{h_2^2 f_2''(x)}{2f_2(x)} - \frac{h_1^2 f_1''(x)}{2f_1(x)} + O\left(h_1^4\right)\right.\\ &\quad \left. + O\left(h_2^4\right)\right].\end{aligned}$$

Thus at x_0 we have a bias given by:

$$\Delta_1(\hat{g}(x_0)) = \frac{h_2^2 f_2''(x_0) - h_1^2 f_1''(x_0)}{4f(x_0)}. \tag{13}$$

which is of order $O(h^2)$.

We then obtain, for $m = 2$

$$\hat{f}_{1,2}(x) \propto \int \frac{1}{h_1} K\left(\frac{x-y}{h_1}\right)\left(\frac{\hat{f}_{2,1}(y)}{\hat{f}_{1,1}(y)}\right)^{1/2} f_1(y)\, dy \tag{14}$$

$$\hat{f}_{2,2}(x) \propto \int \frac{1}{h_2} K\left(\frac{x-y}{h_2}\right)\left(\frac{\hat{f}_{1,1}(y)}{\hat{f}_{2,1}(y)}\right)^{1/2} f_2(y)\, dy \tag{15}$$

By substituting equation (12) into equations (14) and (15), expanding in a Taylor series and making the change of variable we eventually obtain an approximation up to terms of order $h_i^2, i = 1, 2$:

$$\begin{aligned}\hat{f}_{1,2}(x) &= (f_1(x)f_2(x))^{1/2}\left[1 + \left\{\frac{f_2''(x)}{4f_2(x)} + \frac{f_1'(x)f_2'(x)}{4f_1(x)f_2(x)}\right.\right.\\ &\quad \left.\left. - \frac{(f_2'(x))^2}{8f_2^2(x)} - \frac{(f_1'(x))^2}{8f_1^2(x)}\right\} h_1^2 + \frac{f_2''(x)}{4f_2(x)} h_2^2\right]\\ \hat{f}_{2,2}(x) &= (f_1(x)f_2(x))^{1/2}\left[1 + \left\{\frac{f_1''(x)}{4f_1(x)} + \frac{f_2'(x)f_1'(x)}{4f_1(x)f_2(x)}\right.\right.\\ &\quad \left.\left. - \frac{(f_1'(x))^2}{8f_1^2(x)} - \frac{(f_2'(x))^2}{8f_2^2(x)}\right\} h_2^2 + \frac{f_1''(x)}{4f_1(x)} h_1^2\right].\end{aligned}$$

From this we can compute up to terms of order $h_j^3, j = 1, 2$:

$$\begin{aligned}\delta_2(x) = \frac{1}{2}&\left[\frac{\left(h_1^2 - h_2^2\right)}{8}\left\{\frac{f_1'(x)}{f_1(x)} - \frac{f_2'(x)}{f_2(x)}\right\}^2\right.\\ &\left. + \frac{\left(h_2^2 + h_1^2\right)}{4}\left\{\frac{f_1''(x)}{f_1(x)} - \frac{f_2''(x)}{f_2(x)}\right\}\right]\end{aligned}$$

which gives an updated classifier which uses $\delta_1(x) + \delta_2(x)$. Thus at x_0 we have bias given by

$$\begin{aligned}\Delta_2(\hat{g}(x_0)) = &\frac{\Delta_1(\hat{g}(x_0))}{2} + \frac{h_2^2 f_1''(x_0) - h_1^2 f_2''(x_0)}{8f(x_0)}\\ &+ \left(h_1^2 - h_2^2\right)\left(\frac{f_1'(x_0) - f_2'(x_0)}{4f(x_0)}\right)^2\end{aligned}$$

If we now set $h_1 = h_2$ we see that $\Delta_2(\hat{g}(x_0)) = O(h^4)$ so boosting gives bias reduction. That boosting reduces the bias comes as no surprise, but it is somewhat counter-intuitive that the bias reduction is enhanced by taking equal smoothing parameters.

Simple closed form expressions for the variance have eluded us, but we believe that, in common with other applications of boosting, the variance will increase rather slowly with M.

5. Numerical and simulation experiments

We will not address the issue of automatic bandwidth selection for kernel classification. Even in the regular (non-boosting) situation, this is not straightforward. Cross-validation could be a possible solution to finding good pairs (h_1, h_2), but in our simple experiments, the surface often has local minima, in which the loss, given by the number misclassified observations, is a discrete function. However, it is worth reiterating that the automatic or data-based choices of smoothing parameter that have been developed for density estimation (Jones and Signorini 1997) are unlikely to be optimal in the classification setting.

Our case studies consist of four simple discrimination problems. The models are: two gaussian cases, with equal or different variance (M1, M2); a limited support case (M3) and a heavy-tailed case (M4). In particular: M1 := $N(0, 4^2)$, $N(4, 1^2)$, M2 := $N(0, 3^2)$, $N(4, 3^2)$, M3 := $N(4, 1^2)$, exp(2) and M4 := $N(4, 1^2)$, $t(2)$.

As our loss function we will consider the root mean squared error of the estimator $\hat{x}_0 = x : \hat{f}_1(x) = \hat{f}_2(x)$, calculated on B samples with equally sized groups (with $n_1 = n_2$):

$$\widehat{\mathrm{RMSE}}\,(\hat{x}_0) = \left\{ \frac{\sum_{b=1}^{B} (\hat{x}_{0,b} - x_0)^2}{B} \right\}^{1/2}.$$

The reasons for focussing on $\hat{x}_0$ are twofold. Firstly, the above risk criterion allows us to examine the behaviour of the two contributions: namely the bias and variance of the estimator $\hat{x}_0$. Secondly, it re-enforces the fact that the source of inflated error rates is due to poor estimation of the decision boundaries. Connections between x_0 and the error rate are further explored in Friedman (1997).

The secondary solutions for x_0, where they existed, contribute very little to the error rate, and so were simply ignored for simplicity. Note that in some cases the secondary solution was such that $f_1''(x_0)f_2''(x_0) > 0$ which requires special attention; see equation (13) and the discussion in Section 2.2. A further potential problem is that, particularly for small choice of h, we could get multiple solutions to $\hat{f}_1(x) = \hat{f}_2(x)$ in the vicinity of x_0. However, for the values of h considered here, this never occurred in any of our simulations.

In our simulation studies two main aspects are explored. In Section 5.1 we consider using separate estimation, a simple benchmark for kernel density discrimination. Obviously, a discrimination based on independent estimations uses J independent estimates which leads to a partition of $\mathbb{R}$ generated on the basis of the x_0s as defined above. The performance of a number of current estimators are compared. Here the end is threefold: firstly investigating if there is an estimator that behaves better than others in classification (as opposed to density estimation); secondly, to establish whether the bias-reduction properties of higher-order bias kernel methods transfer to the estimation of x_0; thirdly, benchmark accuracy values are established for the subsequent analysis. Figure 2 shows the relationship between the intersection point and the error rate for the models used in the simulations. It appears that the intersection point will give more sensitivity in assessing the performance of our methods. In Sections 5.2 and 5.3 we investigate the performance of the BoostKDC algorithm. In Section 5.2 we numerically investigate the empirical behaviour to check what we formally found for $M = 2$. In Section 5.3 the consequences of various tuning choices of parameters, such as the bandwidths and the number of iterations to be carried out, are explored.

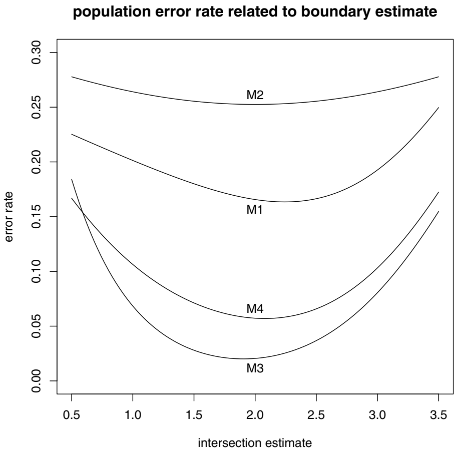


Fig. 2. *Relationship between the error rate and the estimate of x_0 for the four models considered*

5.1. *Separate estimation*

We have compared the performances of five estimators: the linear discriminant (LD), the classic kernel estimator (CK) given by equation (1), two adaptive estimators: the algorithms by Abramson (1982) (AB) and Jones *et al.* (1995) (JLN), and finally the jackknifed higher order kernel (HO). JLN was discussed in Section 4; a brief description of AB and HO follows.

Consider the general formulation

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

where there is a different bandwidth for every sample element. Due to verified practical performance and some optimal analytic properties, a good choice is to take $h(x_i)$ proportional to $\tilde{f}(x_i)^{-1/2}$, where $\tilde{f}$ is a pilot estimator of f (Abramson 1982). This estimator has been closely studied and some theoretical drawbacks have been found (Terrell and Scott 1992, Hall and Turlach 1999); however Abramson's solution is still very appealing for its simplicity and effectiveness. A higher order kernel estimator uses a kernel with order $k > 2$. Since k is the order of bias, there are obvious theoretical reasons to use $k > 2$. Concerning the order of the kernel, there is general agreement that good improvements can often be obtained with $k = 4$. One of the principal reasons why they do not have a greater usage in practice is because they take negative values, so the resultant estimate is not itself a density. In a discrimination setting, this defect is not particularly serious, because we are not primarily interested in a density estimate. In fact, our goal is to determine whether $f_1(x) > f_2(x)$ for given x. However, other drawbacks, such as a difficult choice of bandwidth and the poor enhancements for reasonably sized samples (Marron and Wand 1992, Jones and Signorini 1997), could be still valid in a discrimination framework. Following Jones and Signorini (1997), we consider the generalized-jackknife estimator given by

$$\hat{f}\,(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h} K_{(4)} \left(\frac{x - x_i}{h} \right),$$

with

$$K_{(4)}(u) = \frac{(\mu_4(K) - \mu_2(K)u^2)K(u)}{(\mu_4(K) - \mu_2(K)^2)}$$

where $\mu_j\,(K) = \int x^j K(x)\,dx$.

Concerning the implementation details, we have used two step versions of AB and JLN. This is because in both cases the second step affects the major bias deletion, while the residual

Table 1. *Accuracy values for 5 separate estimations of x_0 such that $f_1(x_0) = f_2(x_0)$ with $n_i = 50$, $i = 1, 2$. The models for the f_i are: M1 N(0, 4^2), N(4, 1^2); M2 N(0, 3^2) ,N(4, 3^2); M3 N(4, 1^2),* exp *(2); and M4 N(4,1^2), t(2)*

	M1					M2				
	LD	CK	HO	AB	JLN	LD	CK	HO	AB	JLN
bias	−.1943	−.1630	−.0793	−.0174	.0161	.0315	.0033	.0073	.0101	.0011
s.d.	.3055	.2354	.2332	.2346	.2390	.3116	.5443	.5349	.5183	.5457
RMSE	.3620	.2863	.2463	.2353	.2396	.3132	.5443	.5350	.5184	.5457
	M3					M4				
bias	.4570	.0148	.0553	.0713	.0865	.0685	.0899	.1355	.1069	.1184
s.d.	.1598	.1603	.1583	.1686	.1736	.2471	.1718	.1812	.1854	.1909
RMSE	.4841	.1610	.1677	.1831	.1940	.2564	.1939	.2263	.2140	.2246

bias is slowly reduced across the successive steps at the expense of a significant variance inflation. Moreover we have used a normalized version of JLN.

We have used the simple normal scale rule $h = 1.06\hat{\sigma}n^{-1/5}$ for two reasons. From a population point of view, we have almost always unimodal symmetric populations, the only exception being the exponential population that, however, is not particularly concentrated near the boundary. From the estimator's point of view, AB and JLN, because of their iterative nature, are robust to the bandwidth selection step, moreover, higher order kernel theory is not particularly developed for bandwidth selection. We use small sample sizes to indicate the effectiveness of the asymptotic arguments with real datasets.

The bias, s.d. and RMSE of the $\hat{x}_0'$s for $n_i = 50$, $i = 1, 2$, and $B = 500$ are reported in Table 1. In problem M1 the $O(h^4)$-biased estimators perform drastically better than CK. A large bias reduction (around 90%) is obtained without a variance inflation. AB and JLN exhibit very similar accuracy values, while HO reduces the bias more modestly (around 51%) but exhibits the smallest variance. In the estimation of problem M2 there is not a bias problem, but AB is a little more stable than the other estimators, note that JLN has the smallest bias and the biggest variance. As expected, LD gives the smallest RMSE since it is optimal for such distributions. Curiously, in problem M3 and M4 CK gives the best results. In problem M3, AB and JLN perform so poorly because their pilot estimation is $O(h)$ biased near zero, HO performs similarly to CK. In problem M4, due to the sparseness of the data in the tails of the t distribution, larger sample sizes are required in order to make effective the properties of $O(h^4)$-biased estimators. However, it should be noted that in the models M3 and M4 there is a nearly symmetric pattern in a wide neighbourhood of x_0. Obviously LD performs very poorly when the population variances are quite different.

5.2. *Two boosting iterations*

In this subsection we have implemented BoostKDC using the standard kernel density estimator, given by equation (1), and the normal scale rule to select $h = 1.06\hat{\sigma}n^{-1/5}$. This very simple automatic choice, which is well-known to oversmooth, should make clear the effect of boosting and should satisfy the requirements of a "weak learner" especially since the normal scale rule tends to oversmooth for non-normal data. Our objective was to observe the reduction in bias, theoretically derived in Section 4.2, and to confirm that a common smoothing parameter ($h_1 = h_2 = h$) was asymptotically superior to separate smoothing parameters. So two bandwidth selection strategies were adopted: the *separate* and the *common* strategy. Two BoostKDC estimators result: one using separate bandwidths, with first step a classical kernel (CK), and second step referred to as 2KB*s*; a second estimator where the same bandwidth $h_{\mathrm{KB}} = (h_1 \times h_2)^{1/2}$ is employed to estimate both f_1 and f_2 (1KB*c* and 2KB*c*). We have chosen the selector h_{KB} for its simplicity and because it will again weaken the learner by oversmoothing. However, in the light of theory of Section 4 we note that the bandwidth selection task should not be crucial for 2KB*c*, provided that the unique bandwidth employed is able to control the effects of higher order bias terms. Actually, we observed numerical evidence to support this hypothesis.

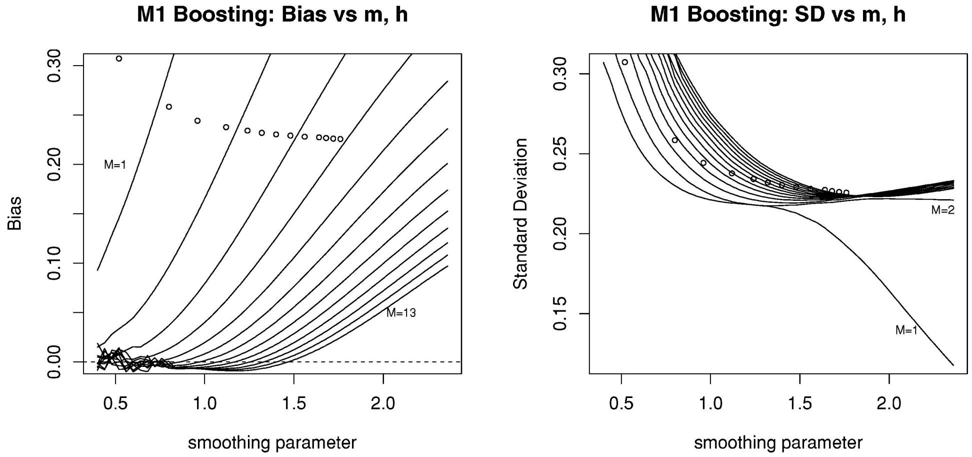
The numerical experiment consists of the estimation of models M1–M4 and the three sample sizes: 50, 100, and 500. The accuracy values of CK, 2KB*s*, 1KB*c* and 2KB*c* are contained in Table 2.

We expect that data from model M1 will generate heavily biased estimates because f_1 and f_2 exhibit quite different curvatures near x_0. However, in correspondence of each sample size our boosting algorithms are clearly less biased than CK. Comparing boosting algorithms, we note that 2KB*c* increases its accuracy values faster than 2KB*s* as n increases. Specifically, comparing the bias magnitudes at $n = 50$, we have −80% for 2KB*c* and −75% for 2KB*s*; while for the sd respectively −65% versus −59%.

In model M4, because of the presence of a heavily tailed distribution, the bias of CK does not decrease for large samples, boosting shows an even more marked ability to reduce it. Here 2KB*c* is substantially unbiased, while 2KB*s* is decidedly less biased than CK, in fact the bias ratios are 0.40 for $n = 50$ and 0.27 for $n = 500$. Moreover, for $n = 500$ 2KB*c* appears more stable than CK.

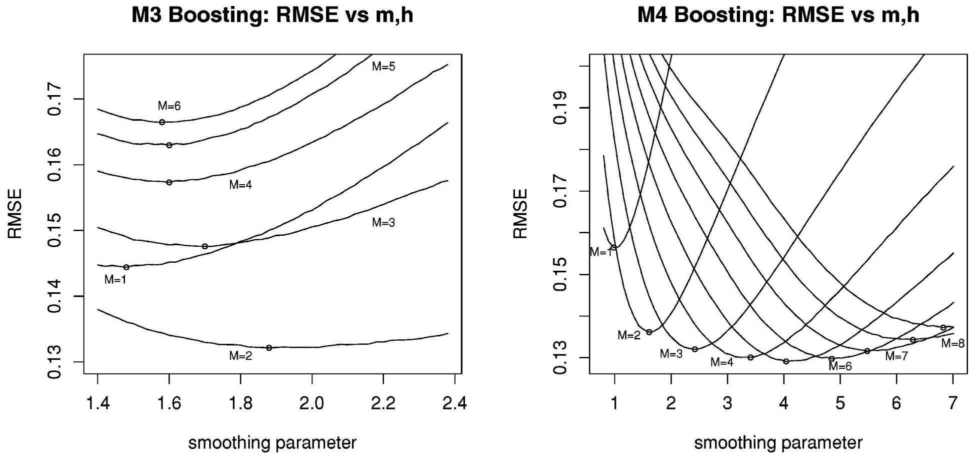
Table 2. *Bias of (i) the classical estimator (CK) and the second boosting step of* BoostKDC *with separate (2KBs) bandwidths and (ii) common bandwidth selection with one (1KBc) and two iterations (2KBc) of* BoostKDC*. Different sample sizes for each of models M1–M4*

	M1				M2			
n_j	CK	2KB*s*	1KB*c*	2KB*c*	CK	2KB*s*	1KB*c*	2KB*c*
50	−.1630	−.1103	−.3973	−.1588	.0003	−.0084	.0062	.0014
100	−.1040	−.0530	−.3057	−.0948	−.0219	−.0358	−.0174	−.0268
200	−.0904	−.0469	−.2437	−.0646	.0165	.0154	.0261	.0164
500	−.0651	−.0265	−.1756	−.0323	.0252	.0275	.0263	.0285
	M3				M4			
50	.0152	−.0306	−.0408	.0044	.0899	−.0359	−.0583	.0001
100	.0027	−.0351	−.0500	−.0082	.0888	−.0374	−.0609	−.0017
200	.0033	−.0248	−.0417	−.0039	.0921	−.0224	−.0518	.0041
500	.0058	−.0135	−.0291	.0009	.0873	−.0232	−.0469	−.0020


M1 Boosting: Bias vs m, h

M1 Boosting: SD vs m, h

Fig. 3. *Effect of number of boosting iterations and the smoothing parameter on bias and variance of estimation of x_0.* ***Left****: Bias;* ***Right****: Standard deviation for $m = 1, \dots, 13$ as a function of h. The points, which are shown on both panels, are the optimal (over h) root mean squared error values for each choice of m*


M3 Boosting: RMSE vs m,h

M4 Boosting: RMSE vs m,h

Fig. 4. *Effect of number of boosting iterations and the smoothing parameter on the root mean squared error (RMSE) of estimation of x_0. Points show minima for each M.* ***Left****: Model M3;* ***Right****: Model M4 as a function of h*

Concerning models M2 and M3, in the previous section we observed the substantial unbiasedness of CK for $n = 50$ because of the perfect or approximate symmetry exhibited near x_0. As a consequence, we expect our boosting will not help. Anyway, for this latter reason M2 and M3 constitute a good benchmark in order to measure the overfitting of our two-step algorithms.

Overall, comparing Tables 1 and 2, we can see that boosting does have a bias-reduction property which, in some cases, mimics those of the higher-order kernel methods. On the whole, if boosting CK works well, then the use of a common bandwidth seems preferable. Finally, an impressive feature of BoostKDC is that it appears robust to non-regular shapes of the populations.

5.3. *More boosting iterations*

In this subsection we explore the performance of boosting when more than two iterations are carried out. According to the boosting principles we expect an initial progressive bias reduction and a modest variance inflation. We expect that after a number of steps both variance and bias will start to increase and from around there we will observe overfitting.

Our objective is to explore the way in which the optimal choice of smoothing parameter varies as the number of iterations increases, and to investigate how many boosting iterations are effective. As noted, boosting cannot work for problem M2 since the distributions are symmetric and so there is no bias. In this equal variance Normal setting a linear discriminant $\hat{x}_0 = (\bar{x_1}+\bar{x_2})/2$ is optimal and this could be approximately achieved by using very large h. So we present results for M1, M3 and M4. For each distribution we simulate 500 samples with $n_1 = n_2 = 50$ and for common smoothing parameters ($h_1 = h_2$) in an appropriate range we calculate $\hat{x}_0$ for $m = 1, 2, \ldots$. Based on these 500 numbers we then estimate the bias and variance which would be achieved for each combination of m and h. Results for data model M1 are shown in Fig. 3, in which we show the bias-variance trade-off. We can see that the bias continues to reduce by boosting for 4 iterations or more, and then a larger value of h can be used to reduce the variance. In terms of RMSE, there is little improvement beyond 7 iterations, after which these values are almost entirely dominated by the variance component. The RMSE results for models M3 and M4 are shown in Fig. 4 and the behaviour is somewhat similar in each: as the number of boosting iterations increases the optimal choice of smoothing parameter also increases. Whereas for model M1 the RMSE corresponding to this optimal choice of h continued to decrease slowly up to $m = 13$ iterations, for models M3 and M4 the optimal choices of m were $m = 2$ and $m = 5$, respectively.

6. Conclusions

The goal of this paper was to consider some theoretical aspects and solutions in kernel density discrimination. Concerning the algorithm BoostKDC, we have demonstrated the utility of boosting in kernel density classification both theoretically and for finite samples. However, obtaining explicit formulae for the variance has proved elusive, and it is not theoretically clear the way in which the bias reduction works for more than two steps.

In many situations, the intersection point x_0 will not be unique, and, since the estimation at such x_0 is critical, adaptive smoothing parameters are likely to perform much better than global smoothing parameters. In particular, if $f_1''(x_0)f_2''(x_0) > 0$ then a much larger smoothing parameter is required; see equation (13). In practical applications, it would be necessary to obtain a rule to enable an appropriate data-based choice of smoothing parameter h and a regularization technique (appropriate choice of M) should also be a matter for concern. In general, it appears that the larger the choice of M, the larger is the optimal smoothing parameter.

A further issue which requires more investigation is the Learning Rate Parameter $1/T$ ($= 1/2$ in step 3(ii)) of our boosting algorithm. We have used $T = 2$, but a larger value makes the learning process slower, reducing the overfitting phenomenon. In fact, values of T a little larger than 2 often generate much more efficient estimates. A further method to ameliorate overfitting would be to use shrinkage (Bühlman and Yu 2003). A final methodological point is establishing if the use of boosting weights $\{w_{i,m}, i = 1, \ldots, n, m = 1, \ldots, M\}$ could be incorporated into the calculation of the bandwidth, so achieving a step-adaptive bandwidth.

The simple nature of BoostKDC allows a straightforward extension to the multidimensional case which is examined by Di Marzio and Taylor (2004b) who show the effectiveness of boosting in reducing the error rate on both simulated and real data.

Acknowledgment

We are grateful to two anonymous referees for detailed and helpful comments that led to significant improvements in this paper.

References

Abramson I.S. 1982. On bandwidth variation in kernel estimates—a square root law. Annals of Statistics 9: 127–132.

Bühlmann P. and Yu B. 2003. Boosting with L_2-Loss: Regression and classification. Journal of the American Statistical Association 98: 324–449.

Di Marzio M. and Taylor C.C. 2004a. Boosting Kernel density estimates: A bias reduction technique? Biometrika 91: 226–233.

Di Marzio M. and Taylor C.C. 2004b. On learning kernel density methods for multivariate data: Density estimation and classification. submitted for publication.

Freund Y. 1995. Boosting a weak learning algorithm by majority. Information and Computation 121: 256–285.

Freund Y. and Shapire R. 1996. Experiments with a new boosting algorithm. In: Saitta L. (Ed.), Machine Learning: Proceedings of the Thirteenth International Conference, Morgan Kauffman, San Francisco, pp. 148–156.

Friedman J.H. 1997. On bias, variance, 0/1-loss, and the curse of dimensionality. J. Data Mining and Knowledge Discovery 1: 55–77.

Friedman J.H. 2001. Greedy function approximation: A gradient boosting machine. Annals of Statistics 29: 1189–1232.

Friedman J.H., Hastie T., and Tibshirani R. 2000. Additive logistic regression: A statistical view of boosting (with discussion). Annals of Statistics 28: 337–407.

Habbema J.D.F., Hermans J., and van der Burgt A.T. 1974. Cases of doubt in allocation problems. Biometrika 61: 313–324.

Hall P., Hu T.-C., and Marron J.S. 1995. Improved variable window kernel estimates of probability densities. Annals of Statistics 23: 1–10.

Hall P. and Turlach B. 1999. Reducing bias in curve estimation by the use of weights. Computational Statistics and Data Analysis 30: 67–86.

Hall P. and Wand M.P. 1988. On nonparametric discrimination using density differences. Biometrika 75: 541–547.

Hand D.J. 1982. Kernel Discriminant Analysis. Research Studies Press, Chichester.

Hastie T., Tibshirani R., and Friedman J. 2001. The Elements of Statistical Learning, Springer, New York.

Jones M.C., Linton O., and Nielsen J.P. 1995. A simple bias reduction method for density estimation. Biometrika 82: 327–338.

Jones M.C. and Signorini D.F. 1997. A comparison of higher-order bias kernel density estimators. Journal of the American Statistical Association 92: 1063–1073.

Jones M.C., Signorini D.F., and Li H.N. 1999. On multiplicative bias correction in kernel density estimation. Sankya 61: 1–9.

Michie D., Spiegelhalter D.J., and Taylor C.C. 1994. Machine Learning, Neural and Statistical Classification, Ellis Horwood, Chichester.

Ridgeway G. 2000. Discussion of Additive logistic regression: A statistical view. Annals of Statistics 28: 393–400.

Shapire R.E. 1990. The strength of weak learnability. Machine Learning 5: 313–321.

Shapire R.E. and Singer Y. 1998. Improved boosting algorithms using confidence-rated prediction. In Proceeding of the Eleventh Annual Conference on Computational Learning Theory.

Silverman B.W. 1986. Density estimation for Statistics and Data Analysis. Chapman and Hall, London.

Terrell G.R. and Scott D.W. 1992. Variable kernel density estimation. Annals of Statistics 20: 1236–1265.

Wand M.P. and Jones M.C. 1995. Kernel smoothing, Chapman and Hall, London.

Wright D.E., Stander J., and Nicolaides K. 1997. Nonparametric density estimation and discrimination from images of shapes. Journal of the Royal Statistical Society C 46: 365–380.