Boosting Kernel Density Estimates: a Bias Reduction Technique?

Marco Di Marzio

Dipartimento di Metodi Quantitativi e Teoria Economica,

Università di Chieti-Pescara,

Viale Pindaro 42,

65127 Pescara, Italy

dimarzio@dmqte.unich.it

Charles C. Taylor

Department of Statistics,

University of Leeds,

Leeds LS2 9JT, UK

c.c.taylor@leeds.ac.uk

SUMMARY

This paper proposes an algorithm for boosting kernel density estimates. We show that boosting is closely linked to a previously proposed method of bias reduction and indicate how it should enjoy similar properties. Numerical examples and simulations are used to illustrate the findings, and we also suggest further areas of research.

Some key words: Bias-reduction; Cross-validation; Kernel regression; Simulation; Smoothing.

The subject of boosting has recently received a great deal of attention from statisticians; see, for example, Friedman et al. (2000). First proposed by Schapire (1990) and subsequently developed by Freund (1995), Freund & Schapire (1996) and Schapire & Singer (1999), boosting was investigated as a means of improving the performance of a 'weak learner'. In the context of classification, the 'learner' was typically a decision tree algorithm. It was weak in the sense that, given sufficient data, it would be guaranteed to produce an error rate which was better than random guessing. In its original setting of machine learning, boosting works by repeatedly using a weak learner, such as a tree 'stump', to classify re-weighted data iteratively. Given n observations, the first weighting distribution is uniform, i.e. $w_1(i) = 1/n, i = 1, \ldots, n$, whilst the mth distribution $\{w_m(i), i = 1, \ldots, n\}$, with $m \in \{2, \ldots, M\}$, is determined on the basis of the classification rule, $\delta_{m-1}(x_i)$ say, resulting from the $(m-1)$th call. The final sequence of decision rules, $\delta_m(x), m = 1, \ldots, M$, is condensed into a single prediction rule which should have superior performance. The weighting distribution is designed to associate more importance to misclassified data through a loss function. Consequently, as the number of iterations increases, the 'hard to classify' observations receive an increasing weight. Moreover, in the case of two classes, a simply 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. 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 characterised by some correlation between the training and test error of the classifiers derived from $\sum_{m=1}^{M} \delta_m(x), M = 1, 2, \ldots$.

Instead of decision trees, Di Marzio & Taylor (2003) propose an algorithm in which a kernel density classifier is boosted by suitably reweighting the data. Simulations indicate that the error rate from this classifier is often lower than the best obtainable from the standard, non-boosted,

kernel density classifier, and a theoretical explanation is given to show how boosting is able to reduce the bias in this problem.

Rather than the traditional goal of boosting classifiers, in this paper we consider the goal of density estimation, and investigate how kernel density estimates can be boosted effectively. In §2 we briefly review the standard results for fixed-bandwidth kernel density estimation. We then propose an algorithm for boosting kernel density estimates, and establish a link between boosting and a previously proposed bias-reduction method. A brief simulation study illustrating some of the results, as well as the limitations, is given in §3, and the paper concludes with a brief discussion of some open questions.

2 Boosting Kernel Density Estimates

2·1 *Standard theory*

Throughout this paper we will suppose the data to be univariate. 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

$$\hat{f}(x; h) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{2·1}$$

(Wand & Jones, 1995, Ch. 2), where h is called the bandwidth or smoothing parameter and the function $K : \mathbb{R} \to \mathbb{R}$, called a kth-order kernel, satisfies the following conditions: $\int K(x)\, dx = 1$ and $\int x^j K(x)\, dx \neq 0, \infty$ only for $j \geq k$.

Suppose that the following standard assumptions hold (Wand & Jones, 1995, pp. 19–20):

(i) f'' is continuous and monotone in $(-\infty, -L) \cup (L, \infty)$, $L \in \mathbb{R}$; $\int (f'')^2 < \infty$;

(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, we obtain the bias as $\mu_2(K)\,h^2 f''(x)/2 + o\left(h^2\right)$ which is of order $O(h^2)$ and var $\{\hat{f}(x)\} = R(K)f(x)/(nh) + o\{(nh)^{-1}\}$ which is of order $O\left\{(nh)^{-1}\right\}$, where, for a real valued function t, $R(t) = \int t(x)^2\,dx$ and $\mu_k(t) = \int x^k t(x)\,dx$. Hence the asymptotic integrated mean squared error is AMISE$\{\hat{f}(\cdot)\} = \frac{h^4}{4}\mu_2(K)^2\,R(f'') + R(K)\,(nh)^{-1}$.

2·2 *Choice of boosting scheme*

Evidently, designing a boosted kernel density estimation algorithm involves two main choices; namely the weighting strategy, i.e. the way to 'give importance' to poorly estimated data, and 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, because of its nonparametric nature, kernel density estimation lends itself to several solutions. Two obvious criteria are to adapt the bandwidths locally or to adapt the mass of the kernels locally. These correspond to undersmoothing and increasing the probability mass of kernels, respectively, for poorly estimated data. In our experiments, varying the mass of the kernel seems the most promising solution. In this case, the traditional kernel estimator, which gives all observations the same mass, corresponds to the weak learner for $m = 1$.

2·3 *The first boosting step ($M = 2$)*

Fundamental to the boosting paradigm is the re-weighting of data based on a 'goodness-of-fit' measure or loss function. As originally conceived, this measure was derived from classifiers, or their relative likelihood. In the case of density estimation, we can obtain such a measure by comparing $\hat{f}(x_i)$ with the leave-one-out estimate (Silverman, 1986, p. 49),

$$\hat{f}^{(i)}(x_i) = \frac{1}{(n-1)h}\sum_{j\neq i} K\left(\frac{x_i - x_j}{h}\right). \tag{2·2}$$

Our boosting algorithm then re-weights the data using a log-odds ratio, i.e. $\log\left\{\hat{f}(x_i)/\hat{f}^{(i)}(x_i)\right\}$. The specific algorithm uses weighted kernel density estimates given by

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

in which the weights are updated at each step, and the final output is the product of all the density estimates, normalised so that the integrand is unity; see Algorithm 1 in §2·4 for a full description.

Considering the first boosting step, $m = 2$, we now compute the weights used to obtain $\hat{f}_2(x)$. Since

$$\hat{f}^{(i)}(x_i) = \frac{n}{n-1}\left(\hat{f}(x_i) - \frac{K(0)}{nh}\right)$$

we then have

$$w_2(i) = w_1(i) + \log\left(\frac{\hat{f}_1(x_i)}{\hat{f}_1^{(i)}(x_i)}\right) \simeq \frac{1}{n} + \frac{K(0)}{nh\hat{f}_1(x_i)} + \log\left(\frac{n-1}{n}\right) \simeq \frac{K(0)}{nh\hat{f}_1(x_i)},$$

since $\log((n-1)/n) \simeq -1/n$, and so $w_2(i)$ is approximately proportional to $\hat{f}_1(x_i)^{-1}$. Hence, for $M = 2$ we have the final estimate given by $\hat{f}(x) = c\hat{f}_1(x)\hat{f}_2(x)$, with c a normalising constant.

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

$$\widehat{f}\,(x) = \widehat{f}_b\,(x) \times \frac{1}{n}\sum_{i=1}^{n} \widehat{f}_b^{-1}\,(x_i)\,\frac{1}{h}K\left(\frac{x-x_i}{h}\right), \tag{2·3}$$

where $\widehat{f}_b$ is estimator (2·1) with the bandwidth b. Equation (2·3) is simply the product of an initial estimate, and a re-weighted kernel estimate, with the weights depending on the inverse of the first estimate. This is of the same form as our boosted kernel density estimate for $M = 2$. The idea behind equation (2·3) is that the leading bias in $\widehat{f}_b\,(x)$ should cancel with the leading bias in $\widehat{f}(x_i)$. In its simplest form, $b = h$. Some simulation results were given in Jones & Signorini (1997) and a recent semiparametric modification of this method was proposed by Jones et al. (1999).

Note also that the $O(h^4)$-biased kernel regression estimator of Jones et al. (1995) enjoys an alternative boosting interpretation as follows. First, consider that it has the same structure as (2·3), being the product of an initial fit and a fit of the ratios between the data and their smoothed values given by the first estimate. Evidently, a logarithmic version of this would constitute a typical 'fitting of residuals' method. In particular, it takes the form of a greedy forward stagewise technique that spans an additive model over 2, or more generally, M basis functions by minimising an L_2 loss. However, this latter procedure is exactly the definition of the boosting algorithm for regression problems discussed by Friedman (2001) and Friedman et al. (2000). Thus, the bias reduction technique introduced by Jones et al. (1995) is closely linked to boosting, although conceived from a very different perspective.

2·4 *Further boosting steps*

Although Jones et al. (1995) realised that their estimator could be iterated, and so $m > 2$, they doubted its efficacy. However, at least for $M = 2$, our boosted kernel estimator should inherit their bias-reduction properties.

Our pseudocode for boosting a kernel density estimate is given in Algorithm 1.

Algorithm 1

Step 1. Given $\{x_i, i = 1, \ldots, n\}$, *initialise* $w_1(i) = 1/n, \quad i = 1, \ldots, n$.

Step 2. Select h.

Step 3. For $m = 1, \ldots, M$, *obtain a weighted kernel estimate,*

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

and then update the weights according to

$$w_{m+1}(i) = w_m(i) + \log\left(\frac{\hat{f}_m(x_i)}{\hat{f}_m^{(i)}(x_i)}\right).$$

Step 4. Provide as output

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

renormalised to integrate to unity.

2·5 *Boosting justification for bias reduction in density estimation*

Suppose that 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 $n \to \infty$. For simplicity we consider a 'population' version rather than a sample version in which our weak learner, for $h > 0$, is given by

$$\hat{f}_{(m)}(x) = \int \frac{1}{h} w_m(y)\, K\left(\frac{x-y}{h}\right) f(y)\, dy,$$

where $w_1(y)$ is taken to be 1. Without essential loss of generality we will take our kernel function K to be Gaussian. The first approximation in the Taylor series, valid for $h < 1$ provided that the derivatives of $f(x)$ are properly behaved, is then $\hat{f}_{(1)}(x) = f(x) + h^2 f''(x)/2$, and so we observe the usual bias of order $O(h^2)$. Now letting $w_2(x) = \hat{f}_{(1)}(x)^{-1}$ the boosted estimator at the second step is

$$\begin{aligned} \hat{f}_{(2)}(x) &= \int K(z) \left\{ f(x+zh) + h^2 f''(x+zh)/2 + O(h^4) \right\}^{-1} f(x+zh)\, dz \\ &= 1 - \frac{h^2 f''(x)}{2f(x)} + O(h^4). \end{aligned}$$

This gives an overall estimator at the second step as

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

so we can see a bias reduction.

3·1 *Iterative re-weighting*

Some insight into the way that the weights change can be obtained by examining one dataset in some detail. We take 250 observations from a standard Normal distribution. The way that the weights change is dependent on the smoothing parameter, and Fig. 1 shows the weights of the first 4 boosting steps for $h = 0.336$, which is the optimal value for the usual kernel density estimate, and $h = 1.456$, which was found to be the optimal value for these data for 3 boosting iterations ($M = 4$). For the smaller values of h boosting has a more drastic effect on the weights, and for the larger h the variation in the weights is more smooth.

[Figure 1 about here]

The effect on the boosted density estimate is shown for these data in Fig. 2. It can be seen that, as a consequence of the re-weighting of the data, boosting has a greater effect for large smoothing parameters. We also note that, if the smoothing parameters are correctly chosen, then the boosted kernel density estimates appear to have smaller integrated squared error than the usual kernel estimate.

[Figure 2 about here]

3·2 *The choice of smoothing parameter and the number of boosting steps*

In the original formulation of boosting, weak learners were used. For example, tree stumps are used in the boosting of decision trees. To expect boosting to work in a kernel density estimation framework, we need to 'weaken' the learner.

If a flexible base learner is employed, we would expect smaller values of M to be optimal. 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, and the other steps can only overfit misclassified data without varying the estimated boundary, thereby degrading the general performance. In order to avoid this, a low-variance base learner is suggested, so that, in effect each stage makes a small, low variance step, sequentially chipping away at the bias.

Obviously kernel density estimation is a flexible base learner, whatever its formulation. Then, in a first approximation we can adopt the criterion suggested by Ridgeway (2000) by significantly oversmoothing, using a bandwidth somewhat larger than the optimal value as obtained from usual methods. This suggestion seems to be supported by the results shown in Fig. 2 in which increasing values of h are needed when more boosting is performed.

3·3 *Experiments with simulated data*

[Figure 3 about here]

In this simple example we investigate how the choice of M affects the optimal choice of h, and how the average integrated squared error changes with these two choices. We use the average over all simulations of the integrated squared error, MISE, as a criterion. Fig. 3 shows MISE(h) for various values of M and for symmetric, skewed, bimodal and thick-tailed distributions. We can see that larger smoothing parameters are required for boosting to be beneficial; this corresponds to the 'weak learner' concept in which boosting was originally proposed. In the symmetric case of a Normal distribution improvement due to boosting is diminishing, with most of the benefit being obtained at the second iteration. The optimal value is $M = 4$ with corresponding optimal smoothing parameter $h = 1.6$. This represents a reduction in MISE to 43% of the optimal value with $h = 0.52$ for the usual estimator, with $M = 1$. In the skewed example, χ^2_{10}, and the thick-tailed distribution, boosting is only beneficial for one step, $M = 2$, after which the MISE

starts to increase. In the bimodal example boosting is beneficial for two steps, $M = 3$. That this boosting algorithm is less effective for skewed and thick-tailed distributions is perhaps caused by the increased presence of 'isolated' observations. In sparse regions $\hat{f}(x_i) \simeq K(0)/(nh)$ and the quantity $\log\left\{\hat{f}(x_i)/\hat{f}^{(i)}(x_i)\right\}$ will be large. Unless the smoothing parameter h is increased, the weights of such observations will lead to overestimates of $\hat{f}(x)$ in sparsely populated tails. In this case, the approximation to the estimator of Jones et al. (1995), will be very poor since $K(0)/\{nh\hat{f}(x_i)\}$ will be close to unity.

4 DISCUSSION

Although the connection to the variable-width multiplicative kernel estimator of Jones et al. (1995) is evident, it in not completely clear how Algorithm 1 works for more than two steps. Also, as for every boosting application, a regularisation technique should be matter of concern. As seen, there are several parameters that could be optimised to obtain a convergent-like pattern of MISE across the steps. A further methodological point is to establish 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 our techniques allows straightforward extensions to the multidimensional case. Indeed, because of its bias reduction character, iterating Algorithm 1 with large smoothing parameters could suggest a powerful new tool for addressing curse of dimensionality effects.

Acknowledgement

We would like to thank the referee, the associate editor, John Kent and Chris Glasbey for helpful suggestions that led to significant improvements in this paper.

DI MARZIO, M. & TAYLOR, C.C. (2003). Kernel Density Classification and Boosting. Submitted for publication.

FREUND, Y. (1995). Boosting a weak learning algorithm by majority. *Inform. Comp.* **121**, 256–85.

FREUND, Y. & SCHAPIRE, R. (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, Ed. L. Saitta, pp. 148–56. San Francisco: Morgan Kauffman.

FRIEDMAN, J.H. (2001). Greedy function approximation: A gradient boosting machine. *Ann. Statist.* **29**, 1189–232.

FRIEDMAN, J.H., HASTIE, T. & TIBSHIRANI, R. (2000). Additive logistic regression: a statistical view of boosting (with Discussion). *Ann. Statist.* **28**, 337–407.

JONES, M.C., LINTON, O. & NIELSEN, J. (1995). A simple bias reduction method for density estimation. *Biometrika* **82**, 327–38.

JONES, M.C. & SIGNORINI, D.F. (1997). A comparison of higher-order bias kernel density estimators. *J. Am. Stat. Assoc.* **92**, 1063–73.

JONES, M.C., SIGNORINI, D.F. & HJORT, N.L. (1999). On multiplicative bias correction in kernel density estimation. *Sankya* A, **61**, 422–30.

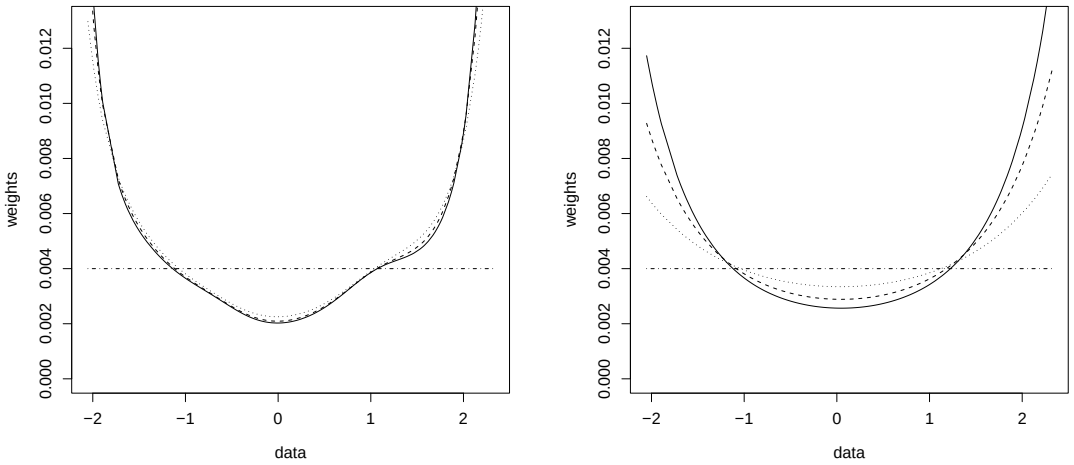
RIDGEWAY, G. (2000). Discussion of 'Additive logistic regression: a statistical view.' *Ann. Statist.* **28**, 393–400.

SCHAPIRE, R.E. (1990). The strength of weak learnability. *Mach. Learning* **5**, 313–321.

SCHAPIRE, R.E. & SINGER, Y. (1999). Improved boosting algorithms using confidence-rated prediction. *Mach. Learning* **37**, 297–336.

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

WAND, M.P. & JONES, M.C. (1995). *Kernel Smoothing*. London: Chapman and Hall.



(a) (b)

Figure 1: Standard Normal sample. Weights for each observation plotted for the first three iterations for smoothing parameter (a) $h = 0.336$ and (b) $h = 1.456$. Dot-dash line: $M = 1$; dotted line: $M = 2$; dashed line: $M = 3$; continuous line: $M = 4$.

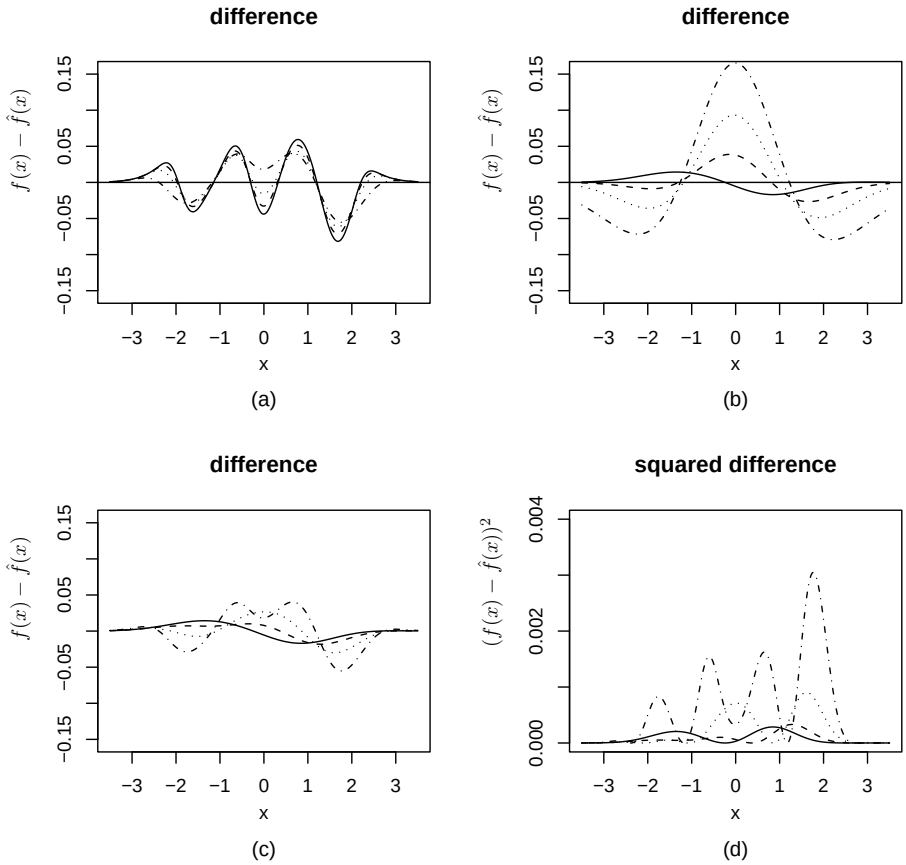


Figure 2: Standard normal sample. (a)-(b) Difference between true density and density estimates ($f - \hat{f}$) plotted for the first three iterations for smoothing parameter (a) $h = 0.336$ and (b) $h = 1.456$. Dot-dash line: $M = 1$; dotted line: $M = 2$; dashed line: $M = 3$; continuous line: $M = 4$. (c)-(d) Difference between true density and density estimates plotted for the first three iterations for the optimal smoothing parameters $h = 0.336$ ($M = 1$), $h = 0.728$ ($M = 2$), $h = 1.120$ ($M = 3$) and $h = 1.456$ ($M = 4$). (c) $f - \hat{f}$; (d) $(f - \hat{f})^2$.

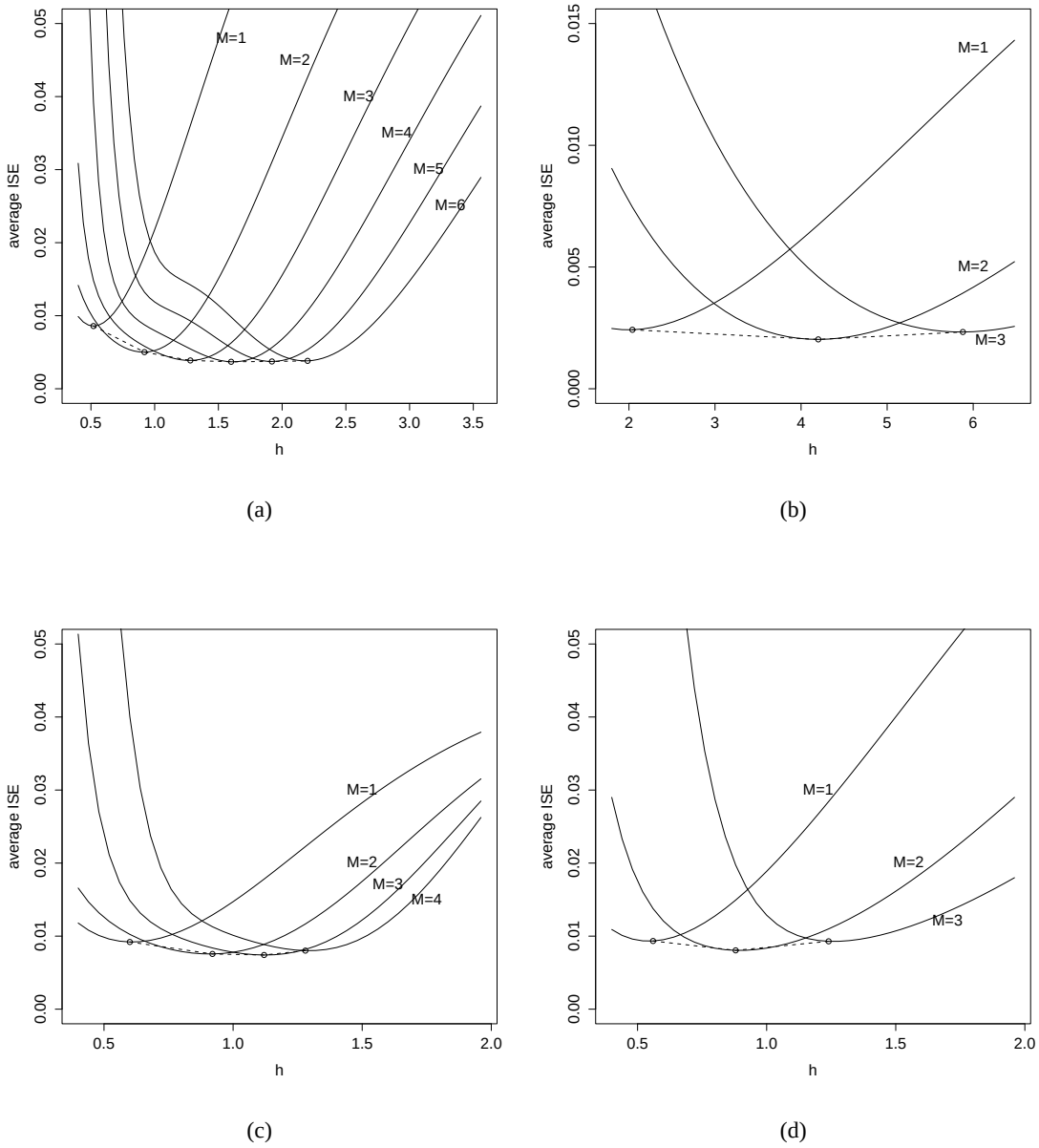


Figure 3: For 500 samples of size $n = 50$ the average integrated squared error is shown as a function of the smoothing parameter h for various values of the boosting iteration M. The dashed line joins the points corresponding to the optimal smoothing parameters for each boosting iteration. Underlying distributions: (a) $N(0, 1)$; (b) χ^2_{10}; (c) equal mixture of normals $N(\pm 2.5, 1)$; (d) t_4.