On Local Polynomial Smoothers and Their Competitors

Peter Hall¹ and Berwin A. Turlach¹,²

SUMMARY. Local polynomial smoothers recently received much attention in the literature, owing to their optimality properties (Fan, 1993). However, Seifert and Gasser (1996a; 1996b) showed that in finite samples these smoothers may suffer problems arising from data sparseness. To overcome this problem they suggest a modification based on ridge regression ideas. In this paper we shall describe another approach, based on interpolation techniques proposed by Hall and Turlach (1997b). This method is easy to implement, and simulation results show that it significantly improves the finite-sample performance of local polynomial smoothers while not interfering with their asymptotic properties. We shall also discuss how interpolation methods (and other schemes) can be used to improve the performance of other well-known smoothers. Using appropriate modifications the performance of these smoothers can be improved to the extent of being arbitrarily close to that of local polynomial smoothers.

KEY WORDS AND PHRASES. Bandwidth, convolution, interpolation, pseudo data, kernel methods, local linear smoothing, local polynomial smoothing, ridge parameter.


¹Centre for Mathematics and its Applications, Australian National University, Canberra, ACT 0200, Australia.
²Cooperative Research Centre for Advanced Computational Systems, Australian National University, Canberra, ACT 0200, Australia.
1 INTRODUCTION

Local polynomial smoothers have a long history in actuarial sciences (see e.g. Woolhouse, 1870; Spencer, 1904; Henderson, 1916) and other applied sciences (see e.g. Macaulay, 1931). They are the basis for widely-used software such as LO(W)ESS (see e.g. Cleveland and Devlin, 1988; Cleveland and Grosse, 1991; Cleveland, 1979, 1993). Although studied earlier in the statistical literature (see e.g. Stone, 1977; Tsybakov, 1986), local polynomial methods recently gained increased attention and popularity in the smoothing community because of their outstanding optimality properties (see e.g. Fan, 1993; Fan et al., 1997) and other attractive features (see e.g. Hastie and Loader, 1993).

However, local polynomial methods are susceptible to sparse design which can lead to erratic behaviour in practice. This problem is illustrated in Figure 1.1 and is reflected in the theoretical observation that the unconditional variance of a local linear estimator, computed using a compactly supported kernel, is infinite. The $X$-observations in Figure 1.1 are drawn from a uniform distribution on $[0,1]$ (with additional observations generated outside this interval to avoid boundary effects) and $Y$-data are obtained from the regression function

$$g(x) = 2 - 5x + \exp\left\{-400(x - 0.5)^2\right\}, \quad x \in [0,1]$$

(1.1)

by adding errors that follow a $N(0,0.5)$ distribution. Note the erratic excursions, which are due to data sparseness and are an artifact of the local linear smoother used. Of course, when analysing a simple dataset, looking at pictures like Figure 1.1 reveals data sparseness problems. However, with more complicated datasets where many local polynomial fits are calculated, one is not always able to examine all plots, and one would like to have a safeguard against artefacts of the local polynomial smoother such as those demonstrated in Figure 1.1.

The problem of data sparsity and the non-existence of moments of the local polynomial estimator are discussed in detail by Seifert and Gasser (1996a; 1996b), who suggest incorporating a ridge parameter into the definition of the local polynomial smoothing estimator, similar in principle to the approach adopted by, for example, Fan (1993). Their ridge technique is relatively sophisticated, employing empirical selection of the ridge as well as minor adjustments to the smoothing parameter. Seifert and Gasser also suggest overcoming the design sparseness problem by making more major adjustments to the amount of smoothing, in particular by increasing the bandwidth in regions of sparse design so that it allows sufficiently many design points to be included. This technique is in the spirit of bandwidth
choice methods based on nearest neighbour smoothing, such as are found in LO(W)ESS.

Here we shall describe an alternative approach proposed by Hall and Turlach (1997b), which is based on adding extra design points in regions where the original design was too sparse. The ordinates corresponding to new design points are calculated by linear interpolation of the neighbouring ordinates. Thus, the original data are augmented by adjoining new “pseudo data” in regions where the original data were distributed too sparsely.

For the sake of simplicity, we restrict ourselves to fixed bandwidths in this paper. But it should be stressed that this interpolation idea is not inherently associated with a fixed bandwidth rule. Indeed, the interpolation step should be seen as a step prior to bandwidth selection, with the amount of interpolation depending on the smallest bandwidth to be considered. As local (and global) bandwidth selectors (see e.g. Brockmann et al., 1993; Ruppert et al., 1995) depend on, among other matters, estimates of the curva-

Figure 1.1: Illustration of the problem that local polynomial methods may have with data sparseness. A local linear smoother with a Gaussian kernel and bandwidth of $h = 0.02$, which is asymptotically optimal in the sense of minimising MISE, is shown. Note the region of data sparseness around $x_0 \approx 0.4$. Sample size is $n = 50$. 


ture of the regression curve, they also suffer from density sparseness. Modifications using a prior interpolation step should be feasible. While a simple nearest-neighbour bandwidth rule alleviates data sparseness problems and may well be appropriate for a wide range of applications, it does not take into account the curvature of the regression curve, and is in general not optimal as indicated by work of Jennen-Steinmetz and Gasser (1988).

By way of contrast, the interpolation approach may be employed with any bandwidth. It is particularly easy to implement, involving a simple and explicit rule for determining where new design points should be added. In small samples the method performs very well, while in large samples, it is designed to achieve the large-sample optimal performance suggested by work of Fan (1993).

The use of interpolation methods in statistics dates back to work by Cauchy 160 years ago, when he proposed an iterative interpolation scheme as an alternative to least squares. Interpolation methods have been exploited to analyse noisy data in actuarial science for more than a century (see e.g. Steffensen, 1950). Clark (1977; 1980) discusses the use of interpolation rules for nonparametric regression (see also Chu and Marron, 1991). Clark’s work effectively addresses the case where an infinite number of pseudo design points are added uniformly between each adjacent pair of original design points, regardless of fluctuations in design density. It may be shown that Clark’s procedure has the effect of increasing variance by the factor $1.5$. The interpolation method discussed here uses much less interpolation and is not subject to this inflation.

We shall discuss how Clark’s procedure can be modified to avoid this inflation of variance. The main idea is to reduce the stochastic fluctuation that affects the estimator through the weights used to construct it. This stochastic fluctuation is the reason why proposals such as Gasser and Müller’s (1979) and Priestley and Chao’s (1972) convolution methods and Clark’s (1977) interpolation approach have larger variance than local linear smoothing, even though they suffer less from data sparseness problems. This observation motivates a wide class of estimators that enjoy good numerical and theoretical performance. The class includes estimators proposed by Jennen-Steinmetz and Gasser (1988), Mack and Müller (1989) and Jones et al. (1994). We shall describe this class as discussed by Hall and Turlach (1997a), and refer to that paper for technical details.

General convolution and interpolation methods allow high-order estimators to be defined explicitly, without matrix inversion, and their large-sample performance is describable directly through their moments. This contrasts with high-order local polynomial methods, which despite having identical first-order asymptotic performance, have positive probabil-
ity of not being well-defined, owing to singularity problems with the matrix used to define them. General convolution and interpolation methods also permit derivatives to be estimated by straight differentiation of the basic curve estimator, the main prerequisites being sufficient smoothness of the kernel and an appropriately-chosen bandwidth. They lead directly to simple techniques for overcoming the problem of stochastic design in applications of wavelet methods. Details of the latter are available in Hall and Turlach (1997c).

In Section 2 we shall introduce the regression model and the (kernel-type) estimators that we consider in this paper. Section 3 describes how these estimators have to be modified to achieve the performance claimed above. We shall not dwell on the properties of these estimators but refer the reader to the extensive literature that exists. Local polynomial methods are treated in Wand and Jones (1995) and Fan and Gijbel (1996). Kernel methods in general are discussed by Müller (1988), Härdle (1990), Scott (1992), Simonoff (1996) and Bowman and Azzalini (1997). Other nonparametric smoothing techniques can be found in Eubank (1988), Thompson and Tapia (1990), Wahba (1990) and Green and Silverman (1994). Section 4 outlines the conclusion of our analysis.

2 Model and Estimators

We assume that we observe \( n \) independent and identically distributed data pairs \((X'_i, Y'_i)\), with the property that \( g(x) \equiv E(Y'_i|X'_i = x) \) is a smooth function. It is assumed that the design points \( X'_i \) come from a continuous distribution with density \( f \), and that they lie within a compact interval \( I = [a, b] \). Suppose for the sake of simplicity that the errors \( \epsilon'_i \equiv Y'_i - E(Y'_i|X'_i) \) have constant variance \( \sigma^2 \). Let \( X_1 \leq \ldots \leq X_n \) denote the order statistics of the sample of values of \( X' \), write \( Y_i \) for the value of \( Y' \) paired with \( X_i \).

2.1 Local Linear Smoothers

A local linear regression estimator \( \hat{g}_t \) of \( g \) may be defined as follows. Let \( K \) denote a kernel function and \( h \) a bandwidth, and put

\[
s_k = s_k(x) \equiv \sum_{i=1}^{n} K\{(x - X_i)/h\}(x - X_i)^k
\]
for \( k = 1 \) and \( 2 \), and \( w_i(x) \equiv K\{(x - X_i)/h\} \{s_2 - (x - X_i)s_1\} \). Then
\[
\hat{g}_1(x) = \left\{ \sum_{i=1}^{n} w_i(x) Y_i \right\} / \left\{ \sum_{i=1}^{n} w_i(x) \right\}.
\] (2.1)

The denominator on the right-hand side vanishes if there do not exist two or more values \( X_i \) within the the support \( S_{x,h} \) of the function \( K\{(x - \cdot)/h\} \). More generally, the estimator \( \hat{g}_1 \) can be very unstable if there are not sufficiently many data values in \( S_{x,h} \).

Define \( \kappa_1 = \frac{1}{2} \int u^2 K(u) \, du \) and \( \kappa_2 = \int K(u)^2 \, du \). A main result from the theory on local polynomial smoothers is that the asymptotic bias and variance of \( \hat{g}_1(x) \) are given by
\[
\text{Asymptotic bias } \{ \hat{g}_1(x) \} = h^2 \kappa_1 g''(x)
\] (2.2)
\[
\text{Asymptotic variance } \{ \hat{g}_1(x) \} = \{n \, h \, f(x)\}^{-1} \kappa_2 \sigma^2.
\] (2.3)

### 2.2 (Clark’s) Interpolation-Type Estimator

Clark (1977; 1980) proposed estimating the unknown regression function by convolving a kernel with the continuous \( Y \)-process obtained by linear interpolation between the observed \( Y \)-values, i.e.
\[
\hat{g}_2(x) = \int Y(u) K_h(x - u) \, du,
\] (2.4)
where \( K_h(\cdot) = K(\cdot/h)/h \) and for \( u \in (X_i, X_{i+1}] \)
\[
Y(u) = \frac{u - X_i}{X_{i+1} - X_i} Y_{i+1} + \frac{X_{i+1} - u}{X_{i+1} - X_i} Y_i.
\] (2.5)

It has been conjectured (see e.g. Chu and Marron, 1991) that the actual bias and variance of this estimator are given by
\[
E\{\hat{g}_2(x)\} - g(x) = h^2 \kappa_1 g''(x) + o(h^2),
\] (2.6)
\[
\text{Var}\{\hat{g}_2(x)\} = \frac{3}{2} \{n \, h \, f(x)\}^{-1} \kappa_2 \sigma^2 + o \left\{ (n \, h)^{-1} \right\}.
\] (2.7)

See Hall and Turlach (1997a) for a proof.

Note that the actual bias of \( \hat{g}_2 \) given in (2.6) equals the asymptotic bias of the local linear estimator \( \hat{g}_1 \) given in (2.2), whereas the actual variance (2.7) of \( \hat{g}_2 \) is inflated by a factor of 1.5 compared to the asymptotic variance (2.3) of \( \hat{g}_1 \). In (2.5) we can write, for
where \( w_j(u), -\infty < j < \infty \) are weights depending only on the design points \( X_i \). The inflation of the variance by the factor 1.5 is due to the variability of the weights \( w_j(u) \) used in (2.5). In Section 3.2 we shall discuss how the weights may be modified to reduce this inflation factor.

### 2.3 Convolution Type Estimator

Let \( \{a_i, b_i\}, 1 \leq i \leq n \) denote a sequence of functions of the design points \( X_1, \ldots, X_n \), such that \( a_1 \leq \ldots \leq a_n, b_1 \leq \ldots \leq b_n \), and \( a_i \leq b_i \) for each \( i \). For a constant \( c > 0 \), define

\[
\hat{g}_3(x) = c \sum_{i=1}^{n} Y_i \int_{a_i}^{b_i} K_h(x - u) \, du, \tag{2.8}
\]

\[
\hat{g}_4(x) = c \sum_{i=1}^{n} Y_i (b_i - a_i) K_h(x - X_i), \tag{2.9}
\]

where the latter can be interpreted as an approximation of the former. The normalisation constant \( c \) is completely determined by the weights, and is such that \( \mathbb{E}[\hat{g}_j(x)] = g(x) + o(1) \) for \( j = 3, 4 \). Most choices of \( (a_i, b_i) \) proposed in the literature lead to \( c = 1 \). In Section 3.3 we shall discuss choices of \( (a_i, b_i) \) for which \( c \neq 1 \).

Set \( X_0 = a \) and \( X_{n+1} = b \), being the boundaries of the interval \( I \). Then, choosing \( c = 1, a_i = X_i, b_i = a_{i+1} \) and \( Y(u) \equiv Y_i \) on \( (X_i, X_{i+1}] \) in (2.4) would lead to the estimator given in (2.8). The estimator \( \hat{g}_3 \) in (2.8) obtained by taking \( c = 1, b_i = a_{i+1} \) and \( a_i = \frac{1}{2}(X_i + X_{i-1}) \) was proposed by Gasser and Müller (1979). For the fixed design case, \( \hat{g}_4 \) given in (2.9) was proposed by Priestley and Chao (1972) with \( c = 1, b_i = a_{i+1} \) and \( a_i = X_i \). In the random design case both these estimators have actual bias and variance as given by (2.6) and (2.7). The inflation of the variance is again due to the variability of the weights used in averaging the \( Y_i \)'s in (2.8) and (2.9). In Section 3.3 we shall discuss how these weights may be modified to reduce this inflation factor.
3 Modifications of the Estimators

3.1 Local Linear Smoothers and Interpolation

To overcome the problem that the actual bias and variance of the local linear estimator do not exist, and data sparseness problems in finite samples, Hall and Turlach (1997b) suggest the following procedure.

Assume that the kernel $K$ is supported on the interval $[-1, 1]$, put $X_0 = a$ and $X_{n+1} = b$, and write $S_i = X_{i+1} - X_i$ for the $i$'th spacing, where $0 \leq i \leq n$. Let $J_h$ denote the set of indices $i$ such that $a + h \leq X_i \leq X_{i+1} \leq b - h$. Fix a real number $r$, and write $m_i$ for the integer part of $rS_i/(2h)$ (if $i \in J_h$) or for the integer part of $rS_i/h$ (otherwise). If $m_i$ equals 1 or more, add $m_i$ equally-spaced pseudo design points to the interval $(X_i, X_{i+1})$, thereby dividing it into $m_i + 1$ equal portions of length not exceeding $2h/r$ (if $i \in J_h$) or $h/r$ (otherwise). If $m_i = 0$, we do not add any pseudo design points. (The rule described in Hall and Turlach (1997b) is slightly different from the one described here, but it was pointed out to us by Edwin Choi and Burkhardt Seifert that the algorithm given in Hall and Turlach (1997b) may not give the best results in extreme situations.)

This rule ensures that whenever $x \in I$, the number of design or pseudo design points in the interval $(x - h, x + h)$ is at least equal to the integer part of $r$; and, arguably more importantly, that none of the spacings between those points exceeds $2h/r$. This is critical to ensuring that the denominator of the estimator $\hat{g}_1$ does not stray too close to zero.

To impute $Y$-observations at the generated $X$-values, put $Y_0 \equiv Y_1$ and $Y_{n+1} \equiv Y_n$. If $X^* \in (X_i, X_{i+1})$ denotes a pseudo design point generated by the rule above, define the corresponding value of $Y^*$ by linear interpolation between the pairs $(X_i, Y_i)$ and $(X_{i+1}, Y_{i+1})$, for $0 \leq i \leq n$.

Figure 3.1 illustrates the effect of applying these rules to the data shown in Figure 1.1. The original data pairs are shown as dots, the pseudo data pairs for $r = 3$ as crosses and the original design points as a “rug” along the horizontal axis. Curve estimates $\hat{g}_1$ are depicted for several different values of $r$ given in the legend. Note particularly that there are several obvious gaps in the set of design points, and that these have been filled by pseudo design points. Hall and Turlach (1997b) note that choosing $r$ as small as possible gives the best finite sample performance for this modification of the local polynomial estimator.

The rules for imputing pseudo data described here are by no means the only ones possible. Especially in the multivariate case, a myriad of possibilities exist; see Hall et al. (1996).
3.2 Clark’s Estimator Revisited

Hall and Turlach (1997a) propose replacing the simple (linear) averaging in (2.5) by a \( \nu \)-running means fit to the \( Y_i, \nu \geq 1 \). That is, define \( Y(u) \) for \( u \in (X_i, X_{i+1}] \) by

\[
Y(u) = \frac{1}{2\nu} \sum_{j=-\nu+1}^{\nu} Y_{i+j}.
\]

They show that with this modification the actual bias and variance of the estimator are given by

\[
E\{\hat{g}_2(x)\} - g(x) = h^2 \kappa_1 g''(x) + o(h^2),
\]

Figure 3.1: Estimates of the function \( g \), defined at (1.1). Typical estimates are depicted, successively shifted vertically by 2 units, their type indicated by legends on the graphs. All estimators were constructed using the Epanechnikov kernel and bandwidth \( h = 0.0457 \), which is asymptotically optimal in the sense of minimising MISE. The original data set (dots) was the same as in Figure 1.1. Crosses indicate the added pseudo data.
\[ \text{Var}\{\hat{g}_2(x)\} = d \{nh f(x)\}^{-1} \kappa_2 \sigma^2 + o\left\{(nh)^{-1}\right\}, \quad (3.3) \]

with \( d = 1 + (2\nu)^{-1} \). Note that by choosing \( \nu \) appropriately, \( d \) can be arbitrarily close to one, i.e. the first-order formula for the variance of the modified estimator can be arbitrarily close to the asymptotic variance of the local linear estimator given in (2.3).

### 3.3 Convolution Type Estimator Revisited

As mentioned above, the inflation of variance of the estimators given in (2.8) and (2.9) is due to the variability of the weights used in the definition. To alleviate this problem we may consider the following alternative definitions of \( a_i \) and \( b_i \), where \( \nu \) is an integer:

\[
(a_i, b_i) = \left( \nu^{-1} \sum_{j=1}^{\nu} X_{i-\nu+j}, \nu^{-1} \sum_{j=1}^{\nu} X_{i+j-1} \right), \quad \nu > 1; \quad \text{or} \quad (3.4)
\]

\[
(a_i, b_i) = (X_{i-\nu+1}, X_{i+1}), \quad \nu \geq 1. \quad (3.5)
\]

(Of course, we have to take care how to define \( a_i \) and \( b_i \) close to the boundary. There it is also necessary to use boundary kernels which can be done in any of several ways. Details are given in Hall and Turlach (1997a).)

If \((a_i, b_i)\) are chosen according to (3.4) or (3.5) then they satisfy

\[
E(b_i - a_i) \sim \{cn f(x)\}^{-1}
\]

\[
E(b_i - a_i)^2 \sim d\{cn f(x)\}^{-2}, \quad (3.6)
\]

with \((c, d) = \left(\frac{1}{\nu - 1}, \frac{2\nu^2 - \nu - 1}{\nu(\nu - 1)}\right)\) or \((c, d) = \left(\frac{1}{2\nu - 1}, \frac{2\nu}{2\nu - 1}\right)\), respectively. For both, \( d = 1 + O(\nu^{-1}) \) as \( \nu \) increases. The first result in (3.6) ensures that \( a_i \) and \( b_i \) are the same distance apart, on average, as adjacent order statistics in the sample \( \{X'_0, \ldots, X'_n\} \). That distance is of size \( \{nf(x)\}^{-1} \), when the order statistics are near \( x \), and the constant \( c \) in the first part of (3.6) serves merely to define the average distance as a specific constant multiple of that quantity. The second result in (3.6) reflects the Cauchy–Schwarz inequality, which declares that \( E(b_i - a_i)^2 / \{E(b_i - a_i)\}^2 \) exceeds 1.

In the case where \((a_i, b_i)\) is chosen according to (3.5), the estimator in (2.9) has been considered by Jones et al. (1994). When \((a_i, b_i)\) is given by (3.4), with \( \nu = 2 \), the estimator in (2.8) is identical to one discussed by Gasser and Müller (1979), Müller (1988) and Chu and Marron (1991). Other special combinations of (2.8) or (2.9), with (3.4) or (3.5), were
Figure 3.2(a): Mean integrated squared error as a function of bandwidth $h$, on a logarithmic scale, for estimators of $g$, defined at (1.1), with different sample sizes $n$. Panel (a) shows estimator (2.8) with $a_i$ and $b_i$ determined by (3.4) and different choices of $\nu$. Panel (b) shows, for the same choices of $a_i$ and $b_i$, the performance of the estimator given at (2.9). The behaviour of the estimator defined by (2.4) and (3.1) is shown in panel (c). Each curve is based on 5000 simulations, with uniformly distributed design points and $\sigma^2 = 0.5$. Vertical lines indicate asymptotically optimal bandwidths.

discussed by Jennen-Steinmetz and Gasser (1988) and Mack and Müller (1989).

In general, if $(a_i, b_i)$ is chosen according to (3.4) or (3.5), and the corresponding $c$ given after (3.6) is used in (2.8) or (2.9), then the actual bias and variance of the modified estimator are given by (3.2) or (3.3), respectively, with $d$ as given after (3.6). These results are discussed at greater length by Hall and Turlach (1997a). Closely related results were obtained independently by Herrmann (1996). Hall and Turlach (1997a) also report results from an extensive simulation study, parts of which we summarise below.

The simulations reported in Hall and Turlach (1997a) show that higher values of $\nu$ lead in the finite sample case to an increase of bias, and that this increase may more then compensate for the reduction in variance. Since (3.5) results typically in a larger
distance between $a_i$ and $b_i$, compared with (3.4), the results using (3.5) with a given $\nu$ are comparable to those obtained employing (3.4) with a slightly larger value for $\nu$.

Panel (a) of Figure 3.2 depicts mean integrated squared error curves for different sample sizes, for values of $\nu = 2(2)10$ and for the estimator $\hat{g}_3$ defined by (2.8). Panel (b) presents the results of using the same parameter settings for the estimator $\hat{g}_4$ defined by (2.9). Panel (c) depicts the estimator $\hat{g}_2$ defined by (2.4) and (3.1), with $\nu = 1(2)9$. All curves were obtained by evaluating, for each replication, the corresponding smooths at 400 equispaced points and calculating integrated squared errors using the trapezoidal rule. The curves represent the means of 5000 realizations of these integrated squared error curves. The grid of bandwidths consisted of 51 logarithmically equispaced bandwidths.

Obviously, for small sample sizes a larger value of $\nu$ results in poorer performance of the estimator, due to the increase in bias. For medium sized samples, usually $n = 100$, the estimators start to have comparable performance over the range of $\nu$’s that were used in the simulation study. Here, some intermediate values of $\nu$ may outperform the smaller
and larger choices. Beyond a sample size of $n = 500$ the higher values of $\nu$ outperform the smaller values.

To assess the efficiency of the modified estimators, Hall and Turlach (1997a) also calculated the minimal value of each integrated squared error curve. This was done by quadratic interpolation around the value of the $h$-grid which achieved smallest integrated squared error. The ratio of the asymptotically achievable mean integrated squared error to the mean of these minimal values is a measure of the efficiency of the estimator; see Seifert and Gasser (1996a). The results of these calculations are presented in Figure 3.3.

Again the adverse effects of increased bias are visible. For small sample sizes, smaller values of $\nu$ produce higher efficiency. Note that the solid line in Figure 3.3(a) corresponds to the “standard” Gasser–Müller estimator. The results in Seifert and Gasser (1996a) indicate that for this setting the Gasser–Müller estimator has for small samples a higher efficiency than an unmodified local linear smoother. Only with modifications such as ridge regression does local linear smoothing dominate the Gasser–Müller estimator.
Figure 3.3(a): Efficiency as a function of sample size, $n$, for estimators of $g$ in (1.1); at integrated squared error-optimal bandwidths and for uniformly distributed design points, with $\sigma^2 = 0.5$. Panel (a) shows estimator (2.8) with $a_i$ and $b_i$ determined by (3.4). Panel (b) shows, for the same choices of $a_i$ and $b_i$, the performance of the estimator given at (2.9). The behaviour of the estimator defined by (2.4) and (3.1) is depicted in panel (c).

Figure 3.3 also shows that for small values of $\nu$ the estimator reaches its asymptotic efficiency rather quickly, after which its performance levels off. By way of contrast, with higher values of $\nu$ the asymptotic efficiency is achieved only for relatively large sample sizes. The theoretical prediction, that asymptotic efficiency increases with $\nu$, and that an efficiency arbitrarily close to one can be achieved by choosing $\nu$ sufficiently large, is clearly reflected in these results.

4 Conclusions

We described how to overcome the problem of design sparsity for local polynomial methods using simple interpolation schemes. This approach achieves first–order mean–square
optimality in an unconditional sense. Likewise, the generalisations of the interpolation and convolution type estimators discussed in this paper make these methods competitive with local polynomial smoothers, and allow them to come arbitrarily close to the optimal performance of local polynomial smoothers. Overall, $\nu = 2$ produces good performance of convolution methods for $n < 100$, although for $n \geq 100$, performance can be improved by taking $\nu$ equal to the integer nearest to $\log_{10} n$. Empirical choice of $\nu$ is still an open question, but (unlike the case of bandwidth choice) the effort may not be justified.
References


87