Smoothing By Local Fitting: Various Issues Stimulated by a Paper of Fan and Marron

William S. Cleveland and Clive Loader

Abstract

Using a recent paper of Fan and Marron as a backdrop, a number of statistical and computational issues of the local fitting approach to smoothing are discussed. The topics are the following: (1) a benchmarking study of Fan and Marron; (2) the relative performance of kernel smoothers vs. local polynomial fitting of degree one and higher; (3) integrated vs. isolated solutions to computational problems; (4) striving for parsimony in local regression modeling; (5) what is the approximation and what is the base method when computational methods are developed; (6) the relevance of the current asymptotic theory for local fitting.

Fan and Marron [11] published a paper on computational methods for the local fitting approach to smoothing for the special case of one predictor. They investigated two local fitting methods: kernel smoothing [23, 36, 25, 1] and locally linear fitting [23, 33, 2, 4, 16, 9]. They focus on the problem of computing the fit, that is, the values of the regression curve. They investigate two methods, binning and updating. Binning, of course has long history in local fitting [28], and the updating method is similar to one presented by Seifert et al. [29]. In addition, Fan and Marron placed information and code in the statlib `jcgs` archive.

An important part of the Fan and Marron paper is a benchmarking study: Monte Carlo comparison of their new computational methods with the LOWESS implementation of locally weighted regression, software written in the 1970s that is available in a number of statistical packages. The authors write

> It seems clear, however, that the methods described in this article are very competitive (in terms of computational speed) with LOWESS, which has been considered a useful smoother because it is viewed as being very fast to compute.

And Seifert et al. [29] report

> The attractive computational efficiency of updating algorithms has also been confirmed by Fan and Marron (1993) in a comparison with existing fast algorithms.

Here, we comment on the Fan and Marron paper and the archive. Then, with the Fan and Marron paper and other work in smoothing as a backdrop, we discuss a number of issues of statistical and computational methods for local fitting.

Note that while we use the term "smoothing" to describe the methodology, which implies a nonparametric approach, there are plenty of parameters lurking everywhere. Local fitting in practice often involves explicit parameters such as the variances of Gaussian errors or the coefficients of conditionally-parametric fits or parametric terms in additive models, and always involves implicit parameters such as degrees of freedom for fits [2, 5, 6].

1 Summary

Sections 2 to 8 contain detailed comments on the Fan and Marron paper and detailed discussions of local fitting. The following is a summary.

1.1 Summary of Comments on the Paper

Section 2: The Fan and Marron paper has major errors in the description of the basic properties of locally weighted regression [2] and the description of the details of the computational method that is implemented in the widely distributed code LOWESS. In isolation, these misunderstandings would be minor, just reporting errors, but in the authors' benchmarking study the errors result in a comparison of LOWESS with the wrong versions of their methods.

Section 3: Unfortunately, the authors goal of reliably benchmarking LOWESS with the binning and updating methods is not achieved. The reason is that the two approaches are carried out under very different computational settings. First, certain settings could have been made commensurate, but were not. This completely distorts the speed comparisons with LOWESS, by factors of over 5 in all cases, and as much as 50 for small sample sizes. But more problematically, there are different types of computational parameters for the different methods that have a substantial impact on speed; the authors made no attempt to control for these parameters in their Monte Carlo experiment.

Section 4: Even more fundamentally, LOWESS was a poor choice for benchmarking. In recent years, far more sophisticated computational methods for local fitting have been developed [4]. These methods, the LOESS system, are available in widely distributed software [5]. A LOESS fit is characterized by data structures that can be stored in a much more compact manner than the original data set. This type of representation is well suited to multiple evaluations of a fit at different points in the design space, which is typically needed when a set of data is thoroughly analyzed in practice by local fitting methods [3].

1.2 Summary of Discussions of Local Fitting

Section 5: Suppose we want develop a computer implementation of a statistical method such as local fitting that requires a collection of computing methods. Post hoc integration of individual computing methods that result from isolated attacks on the computing issues is typically far less fruitful and less likely to succeed than an approach that attacks the problems in an integrated way at the outset.

Section 6: Many smoothing methods have advanced by developing a statistical method, investigating its properties by theory and simulation, and then using computational methods that approximate the original method in the sense that even with perfect arithmetic the computational and statistical methods would not agree. It is certainly useful to investigate how well the computational method approximates the base method and to infer, carefully, properties of the computational method from the base. But in the end, the computational method is the actual method and the original statistical method is the approximation; hence the properties of the computational method are paramount.

Section 7: Asymptotic theory has the potential to shed light on statistical methods, which are always used in practice with finite samples. The relevance of the asymptotic theory depends on the realism of the framework and, given a realistic framework, how well asymptotic distributions approximate finite sample ones. Asymptotic theory for smoothing has not been particularly helpful as a guide for choosing among different procedures because the framework for many studies has not been particularly realistic. We argue that optimality results for choices of bandwidth, polynomial degree, and the form of the weight function for local fitting do not provide helpful guides for what to use in practice because they optimize for highly artificial settings that do not corresponding to most practical settings.

Section 8: Until quite recently, kernel smoothers were the focus of most theoretical work for smoothing [1]. However, simple intuition and experience in practice together with more recent theoretical results show kernel methods do not work [16]. We briefly review the reasons for the failure of kernel methods, and issue a challenge involving real datasets for anyone who doubts the claims.

2 Errors

Fan and Marron write

> (page 54) A major weakness of the LOWESS smoother is that it only allows the uniform kernel .... .

This is incorrect. LOWESS finds nearest-neighbor neighborhoods, weights points in each neighborhood by the tricube kernel, and fits linear polynomials using these nonuniform weights.

Fan and Marron also write

> (page 43) For a grid of x_j values, updating ideas can be used to rapidly compute summations .... The same idea is key to LOWESS ....

This is incorrect. The LOWESS computational method does not use such summation updating.

2.1 The LOWESS Computational Method

As Fan and Marron correctly point out, the uniform kernel has very undesirable properties. This problem has been acknowledged for decades in a host of domains such as signal processing and spectrum estimation [35]. This boxcar function has a Fourier transform with side lobes that are of appreciable size, so behavior well away from a frequency of zero can leak into the result. This accounts for the "substantial roughness" noted by Fan and Marron that is plainly visible on graphs of fits using the boxcar. The universal solution, which is used by LOWESS, is a kernel that smoothly descends to zero; in practice it makes sense to do this over a finite interval to increase computation speed. The effect of this smooth kernel is quite obvious on plots of LOWESS curves; the telltale boxcar roughness is not present.

The LOWESS implementation computes the fit only at the data points x_i, uses only locally linear fitting, and is restricted to one predictor. As we shall see, recent implementations of locally weighted regression remove these restrictions.

LOWESS uses several computational methods to increase speed. The first is a nice idea of Paul Tukey. Suppose the x_i are ordered from smallest to largest. The fit at x_1 is computed. Then a step is taken to $x_1 + \delta$. If there are x_i between x_1 and $x_1 + \delta$, the fit is computed at the largest such x_i and the fit at values between x_1 and this largest value are computed by linear interpolation. This stepping out by δ is carried out repeatedly; each step goes out from the point of the last full computation of local fitting.

The idea of the method is that it is often unnecessary to fully compute the local fitting method at a dense set of x values; because the fit is smooth, it is sufficient to do a full computation at a sparse set and do fast interpolation at other points. Of course, the interpolation needs to be sound and it should be commensurate with the local fitting method. Tukey used linear interpolation because LOWESS used locally linear fitting. Locally weighted regression using locally linear fitting preserves linear functions; so does the implementation.

A second computational method is an updating scheme for finding nearest neighbors, not, we emphasize, to be confused with the summation updating for boxcars that is incorrectly attributed to LOWESS by Fan and Marron.

3 Benchmarking

The computational settings for the comparisons by the authors of their methods and LOWESS, shown in their Figure 8, are incommensurate. The result is unreliable information about the relative speeds of the two approaches. Some of the incommensurate aspects are easily rectified. Other aspects, fundamentally incommensurate parameters of the different methods, are not.

The authors run LOWESS from within S and compare it with a direct C implementation of the their binning and updating methods. S is a high level interpreted language with substantial overhead for data management and input/output, while C is a relatively low level compiled language. The S environment provides for easy implementation of new statistical models and methods; it is in no way intended to compete with the computational efficiency of compiled languages such as C and Fortran. To rectify this problem we ran their benchmarking test set from within S on a Silicon Graphics Indigo and timed both the S overhead and the local fitting. Our total times are nearly proportional to theirs so it is reasonable to infer the percentage of time their runs spent in S overhead from the percents on our runs. For the six values of n — 25, 100, 250, 1000, 2500, and 10000 — the Fan-Marron LOWESS times are increased by the following factors as a result of the S overhead: 3.28 1.56 1.20 1.09 1.06 1.04.

In their call to LOWESS, Fan and Marron use three robustness iterations; roughly, k robustness iterations increase the run time by a factor of approximately $k + 1$. Thus, for the six values of n, the Fan-Marron LOWESS times are increased by a factor of approximately 4.

The authors compare LOWESS with a version of their methods with a uniform kernel. This means that LOWESS is doing many kernel evaluations while their method is not. However, we can adjust the results by comparing the run times produced by the Fan-Marron program with the uniform kernel and the run times produced by the triweight kernel, which are slower; the triweight kernel and the tricube kernel used in LOWESS are both compactly supported and have two continuous derivatives. For the six values of n, the triweight function increases the run times for the Fan and Marron update method by the following factors: 4.44 4.26 3.69 2.50 1.86 1.37. For the binning method the factors are the following: 1.14 1.14 1.13 1.60 1.54 1.33.

Together, these three sources (S overhead, robustness, and kernel choice) affect the speed comparisons by large factors; adjusting for the incompatibility speeds up LOWESS relative to the Fan-Marron implementations by factors of at least 5 in all cases, and as much as $3.28 \times 4 \times 4.44 > 50$, the factor for the case of updating with $n = 25$.

In addition, there is a problem with the Fan and Marron three-estimate comparisons. For LOWESS, their code simply makes successive calls to the S `lowess()` function. But for their binning and updating methods, they repeat only part of the computation; the preprocessing stages, binning and sorting, are only done once (see also Table 1 of Fan and Marron). This problem can be rectified by simply ignoring the three-estimate comparisons, which add little marginal information.

Without full knowledge of the LOWESS computational method, another comparison that might appear to be incommensurate is the number of points, g, at which the computation is carried out.

For LOWESS this is not an important factor because the interpolation is very fast. The speeds for a LOWESS evaluation at 25 points, the smallest number of observations in the benchmarking study, and at 401 points, the number used by Fan and Marron for the evaluations of the binning and updating methods, would be nearly the same.

For LOWESS the critical computational parameter is δ. It controls the number of values at which the full LOWESS computation is carried out. For a large amount of data, increasing δ by a factor of d speeds up the computation by a factor of d. The authors have ignored the parameter δ and simply used the default value, which is 1% of the range of the x_i. One might argue that many users of LOWESS work within S with default settings, and hence this is the correct comparison to make. But this clearly contradicts the purpose of the Fan and Marron paper, which is a comparison of algorithms (and in particular speed-ups) for local fitting, and not a comparison of computing environments. This is evidenced by the following comment of Fan and Marron on their methods:

> In our opinion, however, for any software package to be called "modern," it should use algorithms of this type in a central way.

Similarly, the bin size is a critical parameter of the binning method.

4 Modern Computation and LOESS

LOWESS, developed in the 1970s, and the methods presented by Fan and Marron are very limited because they attack an exceedingly small part of the overall domain of computation that is needed to make local fitting useful for practice. During the last few years, new computational methods for local regression that attempt considerably more have been developed and implemented in the widely distributed LOESS software [4, 5]. The system implements locally linear and locally quadratic fitting; it implements conditionally parametric fitting; it allows smoothing as a function of one, two, or more predictors; and it computes quantities that are needed for making statistical inferences from local fits.

It would have been far better to use LOESS in the Fan and Marron benchmarking instead of LOWESS because the former has an improved solution to the specific task of the latter — computing fits.

The LOESS computations are carried out in two distinct stages. In the first stage, local polynomial fitting of degree one or two is carried out at a small number of points (typically much smaller than the sample size), determined by a k-d tree algorithm. The fitted values and partial derivatives of the locally fitted polynomials are stored in a data structure. In the second stage, piecewise cubic blending functions are used to construct the estimate across the cells of the k-d tree. That is, it is the second stage that provides the evaluation of the fit at any desired set of points. This approach works because having a fitted value and derivative at a point x of a smooth surface provides substantial information about the fit in a neighborhood of x. When the local fitting at the vertices is linear, the algorithm preserves linear functions; when the local fitting at the vertices is quadratic, the algorithm preserves quadratic functions provided cross derivatives are not set to zero.

For the second, or evaluation, stage of LOESS, it is only the evaluation data structure that is required; neither the original data nor any other information is needed. In other words, the fit is parsimoniously characterized. Evaluation is $O(g)$ where g is the number of evaluation points and is exceedingly fast. Note that is does not depend on n the sample size. Such a structure is suitable for evaluation of the surface at any set of points in the design space — single points (for prediction), a grid (for plotting the surface), the original data (for diagnostics such as residual plots), or any other collection of points. And the two-step computational approach can serve as a basis for local fitting procedures in more complicated settings. For example, in [22], the model is applied directly to local likelihood density estimation.

There are other computational approaches to smoothing that result in parsimonious data structures. For example, Silverman [30] estimates a bivariate density on a coarse grid of points, and interpolates across cells. And many spline based approaches (e.g., [12], [26]) give rise to piecewise polynomial approximations. Another example is the logspline density estimation methods [19] and the associated `statlib` archive where functions for plotting, computing quantiles, and resampling operate on a fitted object and do not require the original data.

In contrast, the binning and updating methods of Fan and Marron have very limited potential. For binning, the speed gains come from the equal spacing of the data after binning, from using a fixed bandwidth, and from allowing evaluation of the fit only at the bin points. In multiple dimensions, binning would require a tortuously large number of bins. The updating method does not extend at all to multiple dimensions. Neither method would be useful for the local likelihood density estimation just mentioned[22], since the expensive part of that procedure is the optimization, and not accumulation of sums. For a similar reason they are not likely to affect computation time for local regression with degree greater than one, particularly in multiple dimensions; the solution of the normal equations becomes a much more significant part of the computation.

5 Integrated vs. Isolated Solutions

Many computing environments today provide sophisticated solutions to computational methods and user interfaces. The most sophisticated do so by taking a comprehensive approach. Instead of developing individual uncoordinated solutions to isolated tasks, and then trying to bundle them together, a whole system is developed that provides for integration from the outset. A good example is a graphical user interface such as that on the Macintosh and on PCs.

We can expect far more satisfactory software solutions for statistical methods when such a comprehensive approach is taken. If we optimize any one task in isolation, we do not take account of other tasks with which the first must communicate. The others can be substantially degraded by an optimal solution to the first. Furthermore, attacking tasks in isolation can result in different solutions to variants of the same task which after post hoc integration lead to complex systems that are difficult to use and difficult to maintain.

One striking example of the tremendous success that can result from a systems approach is the work of Hoppe et al. [18] who have developed a collection of mathematical, statistical, and computational methods for reconstruction of three-dimensional objects from laser scans.

Today, we should expect comprehensive approaches to local fitting that integrate many features: smoothing as a function of one or more predictors, allowing for categorical variables, additive fits, conditionally parametric fits, local polynomial fitting up to at least degree two, robust fitting, efficient evaluation of surfaces at one point or at tens of thousands of points, rapid computation of quantities needed for diagnostic checking and for statistical inference, and parsimonious data structures that characterize fitted models.

LOWESS and the Fan and Marron methods are incomplete solutions that attack the isolated task of evaluating a local fit for one predictor. Moreover, the Tukey algorithm in LOWESS and the Fan and Marron updating algorithm have no clear generalizations to two or more predictors. The computational methods of LOESS at least attempt an integrated, comprehensive system, although a number of its pieces are very much in need of further development. Possible changes to the methods include local refinement of the k-d tree, possibly combining with local bandwidth choice. More generally, one could abandon the k-d trees and consider alternative structures, such as triangulation [22]. For three or more predictors, the surface can sometimes be discontinuous, and for two or more predictors there can sometimes be flat spots since certain cross derivatives are set to zero; neither of these problems affects the fit appreciably but does result in somewhat unesthetic pictures when they occur and surfaces are visualized.

6 Computational Methods and Approximation

Many smoothing methods have advanced by developing a statistical method, investigating its properties by theory and simulation, and then using computational methods that approximate the original method. There are two cases. In the first, were it not for round-off error, the computational method would agree exactly with the original method. In the second case, the fitting method is altered, and even with perfect arithmetic, the computational method would not agree with the statistical. The updating method of Fan and Marron falls in the first category and the binning method falls in the second.

In the first case, the focus of approximation studies is the accuracy of computation. Fan and Marron address one accuracy issue in their paper — the accuracy of the running sums. But they ignore another important issue. Their formula for the fit, equation (2.3), is the notoriously unstable pocket calculator formula. The resulting instability, bad enough for degree one fitting, becomes particularly acute when one goes beyond locally linear fitting, as is necessary in many applications. The paper of Seifert et al. [29] does address this problem by restarting the updating algorithm at frequent intervals. Of course, the restarting adversely affects the speed of the algorithm.

In the second case, there are two levels of accuracy: round-off error and the deviation of the computational method from the original with perfect arithmetic. It is certainly useful to investigate how well the computational method approximates the original method with perfect arithmetic and to infer, carefully, properties of the computational method from the base. But in the end, the computational method is the actual method and the original statistical method is the approximation Thus the properties of the fit from the computational method are paramount.

Seen in this light, binning has undesirable properties. For example, it is not commensurate with local fitting of degree higher than zero; binning with locally linear fitting does not preserve a line and binning with locally quadratic fitting does not preserve a quadratic.

7 Asymptotics for Smoothing

Asymptotic theory for smoothing has in most cases not been helpful as a guide for choosing among different procedures, since the framework that governs most asymptotic work is not very realistic. Methods have been put forward with a label of "optimal" that provide optimality in no sense that is meaningful for practice.

The most common asymptotics for smoothers are derived as the sample size $n \to \infty$ with the design density converging in a sufficiently strong sense to a density $f(x)$, and a bandwidth $h \to 0$ with (at least) $nh \to \infty$. Then, approximations to the bias, variance, and covariance are derived. Under some regularity conditions, first order asymptotics for a local linear regression are

$$\begin{aligned} E\hat{m}(x) - m(x) &= \frac{h^2}{2} m''(x) \int v^2 K(v) dv + o(h^2) \\ \mathrm{var}\hat{m}(x) &= \frac{\sigma^2}{nhf(x)} \int K(v)^2 dv + o((nh)^{-1}). \end{aligned}$$

Müller [24], appears to be the first to record these.

Unfortunately these asymptotics are very crude, and one must be very careful not to conclude too much. For example, the bias expansion relies on the quadratic term in a Taylor series expansion, and can be poor over the large smoothing windows often used in practice. The variance expansion relies on the design being close to uniform over the smoothing window. The situation becomes even more problematic for fitting higher order polynomials, since the bias expansion depends on higher order derivatives and typical bandwidths are much larger.

These limitations of the asymptotics have been noted before. For example, Rosenblatt [27] in discussion of optimal kernels states:

> The arguments usually given have been of an asymptotic character and it is a mistake to take them too literally from a finite sample point of view.

Stoker [32] goes even further and challenges the standard approach to asymptotics, suggesting instead that a bandwidth fixed as $n \to \infty$ gives a more realistic assessment of the performance of estimates.

Despite these occasional words of wisdom, the use of asymptotics to compare estimates and derive optimal bandwidths and kernels has been reckless. Due to space considerations we can discuss only one example here — bandwidth selection; some further examples of the misuse of asymptotics are discussed in [16].

The most commonly used bandwidth specifications are fixed bandwidths and nearest neighbor bandwidths. These are both simple and easily implemented, and result in linear estimators, which is useful since it substantially simplifies inference. But there is a compelling practical reason for using nearest neighbor rather than fixed. In many applications, fixed bandwidths lead to dramatic differences in the variances of fits as we move from regions dense with data to regions sparse with data. The problem occurs most commonly in going from the interior of the design space to the boundary. The problem occurs even for smoothing as a function of one equally-spaced predictor. And in the extreme case, the fixed-bandwidth estimator is not even defined because there are not data within a neighborhood; running the `Best` program provided in the Fan and Marron code in statlib, on their own data and bandwidths, produces 42101 warnings about the bandwidth being too small.

Use of nearest neighbor based bandwidths fixes this simple problem, as work by Fan and Gijbels [10] and numerous others shows. More generally, nearest-neighbor methods tend to produce less dramatic swings in variance than a fixed bandwidth. While this is a compelling reason for favoring nearest neighbor over fixed bandwidth as a default method, it is not an optimality result. We certainly do not mean to rule out applications in which constant bandwidths might be useful. But we do think that nearest neighbor methods, because of this simple variance consideration, do provide more satisfactory solutions in most cases.

However, the binning method presented by Fan and Marron requires a fixed bandwidth to achieve fast computation times, and this is justified by claims of nearest neighbors being inefficient. The most common basis for such claims originates from asymptotic mean squared error considerations. The bandwidth minimizing the *pointwise* mean-squared error asymptotically is

$$h_{\mathrm{opt}}(x) \sim \frac{C}{(nf(x))^{1/5}} \tag{1}$$

where f is the density of the x_i, and $C = C(x)$ depends on the mean function of y given x and on the kernel but not on the design density [31]. Similarly, a bandwidth based on k nearest neighbors, with $k \to \infty$ but $k/n \to 0$ behaves like

$$h_{\mathrm{nn}}(x) \sim \frac{k}{2nf(x)}. \tag{2}$$

Since h_{opt} has the dependence $f(x)^{-1/5}$, it appears to be much closer to a fixed bandwidth than the nearest neighbor; in particular, h_{nn} is much too large in regions of small density.

But these asymptotics are far too crude to draw any meaningful conclusions. Suppose, for example, that $n = 100$, the x_i are drawn from a standard normal distribution. Suppose $k = 0.75n = 75$. Denote by Φ and ϕ the standard normal distribution and density functions. Then the bandwidth $h_{\mathrm{nn}}(0)$ will be close to the solution of $\Phi(h) - \Phi(-h) = 0.75$; that is, $h_{\mathrm{nn}}(0) \approx 1.15$. Similarly, $h_{\mathrm{nn}}(3) \approx 3.67$, so the ratio is

$$\frac{h_{\mathrm{nn}}(3)}{h_{\mathrm{nn}}(0)} \approx 3.19. \tag{3}$$

By the asymptotics (2) this ratio would be $\phi(0)/\phi(3) = 90.02$. The asymptotic approximation (2) provides no basis on which to meaningfully assess a nearest neighbor bandwidth. The approximation (3) can be viewed as 'fixed bandwidth' asymptotics in the spirit of [32]; here, we would keep k/n fixed to formulate a limit theorem.

Of course, it is very easy to construct examples with nonhomogeneous smoothness where either a fixed or nearest neighbor bandwidth (or any smoother) will fail. For this reason, it is important that smoothers be used in conjunction with proper diagnostic techniques to determine the adequacy of the fit. The development of methods of parametric regression has had a long history of diagnostic methods for the common parametric statistical models fitted to regression data — methods checking for surplus and lack of fit and methods for checking the distributional assumptions of the models [7]. It is much to the credit of researchers in parametric regression that these methods have become part of the mainstream of statistical practice.

Unfortunately this has not been the case in smoothing. Instead, much more attention has been given to 'automatically' selecting a bandwidth and assessing how close the selected bandwidth is to an 'optimal' bandwidth [14], with little regard for the possibility that the 'optimal' bandwidth may be very inadequate. Moreover, the asymptotic justification for modern bandwidth selectors is very questionable. Methods such as that presented by [14] rely on separately estimating the bias and variance of the estimate. But it is very difficult to estimate the bias of a good estimate $\hat{m}(x)$; a 'good' estimate $\hat{b}(x)$ of the bias would require that $\hat{m}(x) - \hat{b}(x)$ is a better estimate of m.

The examples demonstrating that modern bandwidth selectors 'work' typically apply kernel estimators to cases with high curvature, where higher order methods should be used. (See the next section.) This can even be traced through to the asymptotic justifications. For example, with a second order ($r = 2$) kernel, Härdle, Hall, and Marron [14] require the mean function $m(x)$ to have at least four derivatives for their Theorem 1 to apply, and at least seven derivatives for their $n^{-1/2}$ asymptotics to apply. But regardless of how good the bandwidth selector is, the best rate of convergence for a kernel estimate is $n^{-2/5}$, which is suboptimal under the assumption of more than two derivatives [34]. If one unwinds all the kernel carpentry involved in these bandwidth selectors, the end result is that something approximating a local quadratic (or higher order) fit is being used for the sole purpose of selecting the bandwidth of inefficient kernel methods.

The asymptotic bias and variance expansions have been used even further to give rise to procedures that for many practical purposes are disastrous. One example of such a procedure is the wild bootstrap procedure for constructing simultaneous confidence bands [15], a procedure that was justified as being asymptotically correct. But a simulation study in [21] shows the procedure is based on very poor variance estimates, resulting in bad finite sample properties.

As another example where asymptotics are misused, Gasser and Engel [13] compare the asymptotic integrated mean squared errors of fixed bandwidth Nadaraya-Watson and Gasser-Müller estimates, concluding the Gasser-Müller estimate is 'minimax optimal'. But there are many procedures, including local linear regression, that are not even considered in [13], and the optimality conclusions are unjustified. In fact, results of Fan [8, 9] show the local linear regression has superior asymptotic minimax properties.

8 Why Kernel Smoothers Do Not Work

The problems we discuss in this section are well known, and have been addressed in the kernel literature through the use of higher order kernels, boundary kernels, integral kernels and the like; as has been demonstrated in [9, 16], such solutions are at best just crude approximations to local regression. Hence we concentrate here on any locally constant fitting, and do not distinguish between fixed, nearest neighbor and other bandwidth variation schemes.

In practice, the trade-off between variance and bias almost never makes it worthwhile to use kernel smoothers. They cannot even accommodate a linear effect at the boundaries of the design space of the factors; boundary kernels can reduce the bias but they do not exactly follow linear effects for finite samples. For curves and surfaces with peaks and troughs kernel smoothers perform wretchedly; in fact, in these cases even locally linear fitting can do poorly and locally quadratic fitting is often needed. In these cases there is a common occurrence: the largest bandwidth with tolerable bias is so much bigger for locally quadratic fitting than for locally linear fitting that the quadratic method results in a far smoother fit. This is readily seen when certain graphical diagnostic methods are used; see for example Figure 3.10 of [3].

The point is also clearly demonstrated in Figure 5 of Fan and Marron, which shows kernel estimates of a smooth regression function at bandwidths $h/3$, h and $3h$, where h minimizes the asymptotic MISE. The only smooth curve is that resulting from bandwidth $3h$, which has substantially trimmed both peaks and tail troughs. A higher order fit (at least locally quadratic) is crucial even in relatively simple examples like this.

The same is true of nonparametric density estimation. A kernel estimate trims peaks of the density and substantially inflates tails, unless a very small bandwidth is used, leading to a noisy estimate. The extension of local polynomial methods to density estimation has been studied with examples showing the advantage of local quadratic fitting, especially in multiple dimensions [20, 17].

Under sufficient smoothness assumptions, local cubic or higher order methods have better asymptotic properties than local linear or quadratic, but are less likely to be useful in practice due to variance considerations. This is particularly true in boundary regions, which can dominate especially in multivariate cases.

8.1 A Data Challenge

If anyone is in doubt about the claim that kernel smoothers do not work we issue a challenge. Detailed descriptions of fits to a number of data sets using locally linear fitting and locally quadratic fitting are given in [3]. Try to fit the data using kernel smoothers. We believe that for most of the data sets the results will be a disaster; in a few cases it might be possible to get a kernel smoother that produces an answer that is not entirely embarrassing although certainly not an answer that is better than that of locally linear or locally quadratic fitting.

The data sets are available on netlib and statlib by email, ftp, and mosaic. Particularly challenging will be the examples introduced in Chapter 4 of [3], such as the ethanol data (page 188) and galaxy data (page 228). Afficionados of kernel smoothing will probably conclude that bivariate fourth order boundary kernels are needed. Try one and see what happens.

Acknowledgements

We are grateful to John Chambers, Trevor Hastie, and Colin Mallows for helpful discussions of the issues raised in these comments.

References

[1] C. K. Chu and J. S. Marron. Choosing a kernel regression estimator (with discussion). *Statistical Science*, 6:404–436, 1991.

[2] W. S. Cleveland. Robust locally-weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836, 1979.

[3] W. S. Cleveland. *Visualizing Data.* Hobart Press, Summit, New Jersey (books@hobart.com), 1993.

[4] W. S. Cleveland and E. Grosse. Computational methods for local regression. *Statistics and Computing*, 1:47–62, 1991.

[5] W. S Cleveland, E. Grosse, and W. M. Shyu. Local regression models. In J. M. Chambers and T. Hastie, editors, *Statistical Models in S*, pages 309–376. Wadsworth, Pacific Grove, California, 1991.

[6] W. S Cleveland, E. Grosse, and W. M. Shyu. Local regression models. In Chambers and Hastie, editors, *Statistical Models in S.* Wadsworth, Pacific Grove, California, 1991.

[7] C. Daniel and F. Wood. *Fitting Equations to Data.* Wiley, New York, 1971.

[8] J. Fan. Design-adaptive nonparametric regression. *Journal of the American Statistical Association*, 87:998–1004, 1992.

[9] J. Fan. Local linear regression smoothers and their minimax efficiency. *The Annals of Statistics*, 21:196–216, 1993.

[10] J. Fan and I. Gijbels. Censored regression: Local linear approximations and their applications. *J. Amer. Statist. Assn.*, 89:560–570, 1994.

[11] J. Fan and J. S. Marron. Fast implementations of nonparametric curve estimators. *Journal of Computational and Statistical Graphics*, 3:35–56, 1994.

[12] J. H. Friedman. Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19:1–141, 1991.

[13] T. Gasser and J. Engel. The choice of weights in kernel regression estimation. *Biometrika*, 77:377–381, 1990.

[14] W. Härdle, P. Hall, and J. S. Marron. Regression smoothing parameters that are not far from their optimum. *J. Amer. Statist. Assn.*, 87:227–233, 1992.

[15] W. Härdle and J. S. Marron. Bootstrap simultaneous error bars for nonparametric regression. *Ann. Statist.*, 19:778–796, 1991.

[16] T. Hastie and C. Loader. Local regression: automatic kernel carpentry (with discussion). *Statistical Science*, 8:120–143, 1993.

[17] N. L. Hjort and M. C. Jones. Locally parametric nonparametric density estimation. Technical report, Institute of Mathematics, University of Oslo, 1994.

[18] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *Computer Graphics*, 26:71–78, 1992.

[19] C. Kooperberg and C. J. Stone. Logspline density estimation for censored data. *J. Comp. Graph. Statist.*, 1:301–328, 1992.

[20] C. Loader. Local likelihood density estimation. Technical report, AT&T Bell Laboratories, 1994.

[21] C. R. Loader. Nonparametric regression, confidence bands and bias correction. In *Computing Science and Statistics: Proceedings of the 25th Symposium on the Interface*, pages 131–136. Interface Foundation of North America, 1993.

[22] C. R. Loader. Computing nonparametric function estimates. In *Computing Science and Statistics: Proceedings of the 26th Symposium on the Interface*, page To appear. Interface Foundation of North America, 1994.

[23] F. R. Macauley. *The Smoothing of Time Series.* National Bureau of Economic Research, New York, 1931.

[24] H. G. Müller. Weighted local regression and kernel methods for nonparametric curve fitting. *J. Amer. Statist. Assn.*, 82:231–238, 1987.

[25] E. A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9:141–142, 1964.

[26] F. O'Sullivan. Discretized laplacian smoothing by fourier methods. *J. Amer. Statist. Assn.*, 86:634–642, 1991.

[27] M. Rosenblatt. Curve estimates. *Ann. Statist.*, 42:1815–1842, 1971.

[28] D. W. Scott. *Multivariate Density Estimation.* Wiley, New York, 1992.

[29] B. Seifert, M. Brockman, J. Engel, and T. Gasser. Fast algorithms for nonparametric curve estimation. *Journal of Computational and Statistical Graphics*, 3:192–213, 1994.

[30] B. Silverman. Density estimation for univariate and bivariate data. In V. Barnett, editor, *Interpreting Multivariate Data*, pages 37–53. John Wiley and Sons, 1980.

[31] B. W. Silverman. Some aspects of the spline smoothing approach to nonparametric regression curve fitting (with discussion). *J. R. Statist. Soc. Ser. B*, 47:1–52, 1985.

[32] T. M. Stoker. Smoothing bias in density derivative estimation. *Journal of the American Statistical Association*, 88:855–871, 1993.

[33] C. J. Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5:595–620, 1977.

[34] C. J. Stone. Optimal rates of convergence for nonparametric estimators. *The Annals of Statistics*, 8:1348–1360, 1980.

[35] J. W. Tukey. In D. Brillinger, editor, *The Collected Works of John W. Tukey*, pages 309–376. Chapman and Hall, New York, 1994.

[36] G. S. Watson. Smooth regression analysis. *Sankhya A*, 26:359–372, 1964.