A comparative study of nonparametric derivative estimators

John Newell[1] and Jochen Einbeck[2]

[1] Department of Mathematics, National University of Ireland, Galway, Ireland
[2] Department of Math. Sciences, Durham University, Durham DH1 3LE, UK

Abstract: Nonparametric derivative estimation has never attracted much attention as one gets the derivative estimates as "by-products" from a local polynomial or spline fit. However, these estimates often suffer from boundary effects and are very sensitive to outliers. Apart from this, the local polynomial estimators suffer from a systematic downward bias, as we will demonstrate. This article is intended to re-establish research interest in derivative estimation, and to guide the user who needs to work with one of the available packages.

Keywords: Derivatives; Splines; Kernels

1 Motivation

Nonparametric estimation of derivatives is important in a variety of disciplines. Specifically, when considering a regression problem of type $y_i = m(x_i) + e_i$, one is often not interested in $m(\cdot)$ itself, but rather in the relative change dm/dx of m when increasing or decreasing x by a small value dx. An important special case is when x represents time, in which the 1st derivative of m has the interpretation of a speed, and the 2nd derivative of an acceleration, which is of interest in the analysis of growth curves. However, the importancy of estimating derivatives goes far beyond the end in itself. Often one relies on asymptotic approximations in order to obtain bias and variance estimates, confidence intervals, optimal bandwidths, etc., and these expressions usually involve derivatives of $m(\cdot)$, which are normally unknown and have to be estimated. A further field of application for derivative estimators are change point problems. For instance, when analyzing blood lactate data of elite athletes, one is interested in the workload at which the lactate level suddenly rises, which can be detected by finding the maximum of the 2nd derivative (Newell et al., 2005).

2 On nonparametric derivative estimation

There are two main approaches to nonparametric derivative estimation. Consider firstly local polynomials of degree p. The estimator of the j^{th} derivative $m^{(j)}(x)$ $(0 < j \leq p)$ at point x is given by $\hat{m}^{(j)}(x) = j!\hat{\beta}_j(x)$

according to Taylor's theorem, where $\hat{\beta}_j(x)$ is obtained by minimizing

$$\sum_{i=1}^{n} K\left(\frac{x_i - x}{h}\right)\left(y_i - \sum_{j=0}^{p} \beta_j(x)(x_i - x)^j\right)^2$$

in terms of the vector $(\beta_0(x), \ldots, \beta_p(x))$. Thereby K is a kernel function and h the bandwidth controlling the degree of smoothing. Secondly, in spline smoothing, the usual way of estimating derivatives is to take the derivatives of the spline estimate. In other words, if $\hat{m}(x)$ is an estimate of $m(x)$, one considers $\frac{d^j}{dx^j}\hat{m}(x)$ as an estimator of $m^{(j)}(x)$. Several authors have pursued this idea, using splines with (Heckman & Ramsay, 2000) or without penalization.
As these ideas are quite simple, several papers published in the mid-nineties, particularly originating from the local polynomial smoothing community, gave the impression that the entire issue of nonparametric derivative estimation is solved, and as a result the research activity about this topic stalled to some extent. This is unfortunate, as most problems are treated rather cursorily in the literature and many open questions remain. For instance, Ramsay (1998) noted that 'typically one sees derivatives go wild at the extremes, and the higher the derivative, the wilder the behavior', and that further problems arise when it comes to smoothing parameter (bandwidth) selection, where CV and GCV can be 'poor guides'. In the sequel, we discuss some of these issues in the framework of a comparative study.

3 Comparison of available routines

For illustration, we consider a data set generated by contaminating the function $m(x) = x + 2\exp(-16x^2)$, $x \in [-2, 2]$, with very small Gaussian noise ($\sigma = 0.1$). A moderate outlier at the left boundary with coordinates $(-1.97, -1.75)$ and a further outlier at $(0.95, 0)$ were added by hand, giving a total sample size $n = 60$.

3.1 Local polynomial methods

We start with considering the functions `locfit` (contained in the homonymous package) and `locpoly` in package **KernSmooth**. We use the usual default setting $p = j + 1$ as theoretically motivated by Fan & Gijbels, 1996, p. 77ff. The bandwidths are chosen such that the curves pass roughly equally well through the central part of the curve (we used the result of **locfit**'s `gcvplot` for the 2nd derivative, but undersmoothed for the first). Both functions produce a considerable bias there, which cannot be cured by modifying the bandwidth as otherwise the outlier and boundary effects get even worse. In fact, there is a systematic problem with this kind of

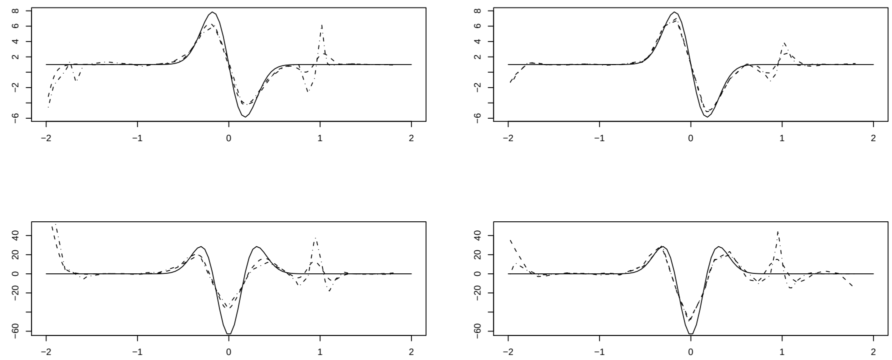

FIGURE 1. Tutorial on behavior of derivative estimators (top: 1st deriv., bottom: 2nd deriv.), left: `locpoly` (dashed), `locfit` (dashed-dotted); right: `smooth.Pspline` (dashed), `D1D2` (dashed-dotted).

estimators: Note that the asymptotic bias of the derivative estimate based on a quadratic fit with bandwidth h is given by

$$\text{Bias}(\hat{m}'(x)|x_1, \ldots, x_n) = c \cdot m'''(x)h^2 + o_P(h^2)$$

($c > 0$ being a constant depending on kernel moments), which can be deduced from Fan & Gijbels (1996), Th^m 3.1. This implies that, where $m'(\cdot)$ is concave, the bias is negative, and where $m'(\cdot)$ is convex, the bias is positive. Hence, concave parts of the derivative will be pulled down and convex parts will be pulled up. As the concave part will usually (but not necessarily in a mathematical sense) correspond to positive and the convex part to negative derivatives, we can speak of a *downward smoothing bias* similar as observed by Stoker (1993) for density derivative estimation. This bias, clearly visible in the left panel of Fig. 1, tends to increase with the derivative order j; one reason is that the necessary bandwidth h (appearing in the bias generally as a factor h^{p+1-j}) increases with j. The smoothing bias diminishes when setting $p = j + 2$ as suggested by Ruppert (1997), at the expense of increased outlier and boundary effects (not shown).

3.2 Spline based methods

We consider here for comparison the functions `smooth.Pspline` (R package **pspline**) and `D1D2` (**sfsmisc**), both using penalized smoothing splines. The latter is restricted to cubic splines, whereas we use for the former a quintic and septic spline for the 1st and 2nd derivative, respectively (Ramsay, 1998). The smoothing parameter is selected for `smooth.Pspline` using the built-in GCV routine, and for `D1D2` such that the fits pass equally well through the central part. Both fits are much less biased than the local polynomial estimators, and more stable at the left boundary.

4 Conclusion

In the conference, we extend this study to include comparisons of the R packages **lokern**, **lpridge** (local), and **SemiPar** (splines). Given the overall stability, functionality, and performance (assessed through a small simulation study), our favorites are rather among the spline based methods; in particular **pspline** and **SemiPar** work generally quite well and offer several interesting options (notably, **SemiPar** is, apart from **locfit**, the only package featuring confidence bands). However, there is a general lack of *robust* derivative estimators. Further, smoothing parameter selection tools are in all packages based on optimizing the estimate of the *regression function* and not of the derivative, which can lead to serious undersmoothing (Jarrow et al., 2004). Function `D1D2` at least addresses this problem by adding a "fudge" offset to the GCV-selected smothing parameter. As a brief guide, the capacities of the packages investigated are summarized below:

Package version	`function`	j_{max}	Smooth. Par.
locfit 1.5-3	`locfit`	2	GCV
KernSmooth 2.22-19	`locpoly`	no limit	—
lokern 1.0-4	`glkerns`	4*	plug-in**
lpridge 1.0-3	`lpridge`	9	—
pspline 1.0-10	`smooth.Pspline`	4***	CV/GCV
sfsmisc 0.95-9	`D1D2`	2	GCV
SemiPar 1.0-2	`spm`	7***	(RE)ML

*if bandwidth selected automatically, then $j_{max} = 2$. **a variant `lokerns` featuring a variable bandwidth is also implemented. ***no formal requirement, but from our experience it breaks down computationally for higher orders.

References

Fan, J., and Gijbels, I. (1996). *Local Polynomial Modelling and Its Applications*. London: Chapman & Hall.

Heckman, N.E., and Ramsay, J. (2000). Penalized regression with model-based penalties. *The Canadian Journal of Statistics*, **28**, 241-258.

Jarrow, R., Ruppert, D., and Yu, Y. (2004). Estimating the term structure of corporate debt with a semiparametric penalized spline model, *JASA*, **99**, 57-66.

Newell, J., Einbeck, J., Madden, N., and McMillan, K. (2005). Model free endurance markers based on the second derivative of blood lactate curves. In: Francis et al. (Eds), *Proceedings of the 20th IWSM*, 357-364, Sydney.

Ramsay, J. (1998). Derivative estimation. StatLib— S-News Thu, 12 March 1998, `www.math.yorku.ca/Who/Faculty/Monette/S-news/0556.html`

Ruppert, D. (1997). Empirical-bias bandwidths for local polynomial nonparametric regression and density estimation. *JASA*, **92**, 1049–1062.

Stoker, T.M. (1993). Smoothing bias in density derivative estimation. *JASA*, **88**, 855–863.