Noname manuscript No.
(will be inserted by the editor)

Challenging the curse of dimensionality in multivariate local linear regression

James Taylor · Jochen Einbeck

Received: date / Accepted: date

Abstract Local polynomial fitting for univariate data has been widely studied and discussed, but up until now the multivariate equivalent has often been deemed impractical, due to the so-called *curse of dimensionality.* Here, rather than discounting it completely, we use density as a threshold to determine where over a data range reliable multivariate smoothing is possible, whilst accepting that in large areas it is not. The merits of a density threshold derived from the asymptotic influence function are shown using both real and simulated data sets. Further, the challenging issue of multivariate bandwidth selection, which is known to be affected detrimentally by sparse data which inevitably arise in higher dimensions, is considered. In an effort to alleviate this problem, two adaptations to generalized cross-validation are implemented, and a simulation study is presented to support the proposed method. It is also discussed how the density threshold and the adapted generalized cross-validation technique introduced herein work neatly together.

Keywords Multivariate smoothing · density estimation · bandwidth selection · influence function

J. Taylor
Durham University
Department of Mathematical Sciences
Durham, UK
E-mail: james.taylor1@durham.ac.uk

J. Einbeck
Durham University
Department of Mathematical Sciences
Durham, UK
E-mail: jochen.einbeck@durham.ac.uk

1 Introduction

Univariate nonparametric regression is widely used to fit a curve to a dataset for which a parametric method is not suitable. Multivariate nonparametric regression methods are not so prevalent, although several methods do exist such as the additive models of Hastie and Tibshirani (1990) and thin plate splines, introduced by Duchon (1977). Here we study the multivariate case of local linear regression, the origins of the univariate equivalent of which can be traced back to the late nineteenth century. The multivariate technique has often been deemed impractical due to the problems encountered in regions of sparse data, which become practically an unavoidable part of data in higher dimensions. This issue is often referred to as the *curse of dimensionality*. However, multivariate local regression has been implemented successfully in Cleveland and Devlin (1988) for 2 and 3–dimensional data and in Fowlkes (1986) for data of even higher dimensions. In this paper we introduce techniques which make the curse of dimensionality avoidable and so regression feasible for any reasonable dimension.

Given d-dimensional covariates $\mathbf{X}_i = (X_{i1}, ..., X_{id})^T$ with density $f(\cdot)$ and scalar response values Y_i where $i = 1, ..., n$, the task is to estimate the mean function $m(.) = E(Y|X = .)$ at a vector $\mathbf{x}$. Assumed is that

$$Y_i = m(\mathbf{X}_i) + \epsilon_i \tag{1}$$

where ϵ_i are random variables with zero mean and variance σ^2. We concentrate on local linear regression which uses a kernel–weighted version of least squares, in order to fit hyperplanes of the form $\beta_0 + \boldsymbol{\beta}_1^T \mathbf{x}$ *locally*, i.e., at each target point $\mathbf{x} \in \mathbb{R}^d$. Both the scalar β_0 and the vector $\boldsymbol{\beta}_1$ depend on $\mathbf{x}$, but we suppress this dependence for notational ease. To find the regression estimate, $\hat{m}(\mathbf{x})$, one minimizes with respect to $\boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_1^T)^T = (\beta_0, \beta_{11}, ..., \beta_{1d})^T$;

$$\sum_{i=1}^{n} \left\{ Y_i - \beta_0 - \sum_{j=1}^{d} \beta_{1j}(X_{ij} - x_j) \right\}^2 K_{\mathbf{H}}(\mathbf{X}_i - \mathbf{x}). \tag{2}$$

The estimator of the mean function $\hat{m}(\mathbf{x})$ is $\hat{\beta}_0$. Here K is a multivariate kernel function with $\int K(\mathbf{u})d\mathbf{u} = 1$ and $K_{\mathbf{H}}(\mathbf{x}) = |\mathbf{H}|^{-1/2} K(\mathbf{H}^{-1/2}\mathbf{x})$. The $d \times d$ *bandwidth matrix* $\mathbf{H}$ is crucial in determining the amount and direction of smoothing since it is this that defines the size and shape of the neighbourhood around $\mathbf{x}$, enclosing data points which are considered in the estimation at this point. For each $\mathbf{x}$, the contours of the weights $K_{\mathbf{H}}(\cdot - \mathbf{x})$ form ellipsoids centered at $\mathbf{x}$, with more weight usually given to those points closer to $\mathbf{x}$.

We choose to use a diagonal matrix, $\mathbf{H} = \text{diag}(h_1^2, ..., h_d^2)$, since computationally this is significantly easier than a full matrix. For K, we primarily use a product of Gaussian kernels since in practice we found this the least temperamental kernel function in regions where data is sparse.

Minimization (2) is a weighted least squares problem. The solution to this is

$$\hat{\beta}_0 = \hat{m}(\mathbf{x}) = \mathbf{e}_1{}^T (\mathbf{X}_\mathbf{x}^T \mathbf{W}_\mathbf{x} \mathbf{X}_\mathbf{x})^{-1} \mathbf{X}_\mathbf{x}^T \mathbf{W}_\mathbf{x} \mathbf{Y} \tag{3}$$

where

$$\mathbf{X_x} = \begin{bmatrix} 1 & X_{11} - x_1 & ... & X_{1d} - x_d \\ 1 & X_{21} - x_1 & ... & X_{2d} - x_d \\ \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} - x_1 & ... & X_{nd} - x_d \end{bmatrix} \tag{4}$$

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix} \tag{5}$$

$$\mathbf{W_x} = \text{diag}\left\{K_{\mathbf{H}}(\mathbf{X}_1 - \mathbf{x}), ..., K_{\mathbf{H}}(\mathbf{X}_n - \mathbf{x})\right\} \tag{6}$$

and $\mathbf{e}_1$ is a vector with 1 as its first entry and 0 in the other d entries.

Local polynomial regression in general, and local linear regression in particular, has many advantages which makes it of interest to find a solution to the problem of the curse of dimensionality. Firstly, the idea has great intuitive appeal, as it is easily visualized and understood which data points are contributing to the estimation at a point. Work by Cleveland and Devlin (1988) and Hastie and Loader (1993a) suggests that multivariate local polynomial regression compares favourably with other smoothers in terms of computational speed. Furthermore, kernels are attractive from a theoretical point of view, since they allow straightforward asymptotic analysis. It has been found that the technique exhibits excellent theoretical properties. Local polynomials were shown to achieve optimal rates of convergence in Stone (1980). In the univariate case, Fan (1993) showed that local linear regression achieves 100% minimax efficiency. The asymptotic bias and variance are known to have the same order of magnitude at the boundary as in the interior of the data, which is a particularly encouraging property for higher–dimensional data sets (Ruppert and Wand, 1994). For *finite* samples in high dimensions, in our experience, the quality of the regression is in fact likely to be poorer at the boundary than in the interior, with the variance in the boundary region being higher for local linear than for local constant estimation. This behaviour is also noted in Ruppert and Wand (1994). However, the quality of estimation of the local linear estimator, in terms of mean squared error, is usually still superior to local constant in these regions, due to the significantly reduced bias. Other advantages, as detailed in Hastie and Loader (1993b), include that it adapts easily to different data design and also has the interesting side-effect of implicitly providing the gradient of $\hat{m}$ at $\mathbf{x}$ through the same least squares calculation. Indeed, this is given by $\hat{\boldsymbol{\beta}}_1$.

Scott (1992) describes the curse of dimensionality as '*the apparent paradox of neighbourhoods in higher dimensions — if the neighbourhoods are 'local', then they are almost surely 'empty', whereas if a neighbourhood is not 'empty', then it is not 'local'.*' If there is not sufficient data in a neighbourhood, then the variance of the fit is too high, or with some kernel functions, such as the Epanechnikov kernel, the calculations may break down completely.

In order to simplify the problem, one may consider using dimension reduction or variable selection techniques in a pre–processing step. Examples

of such techniques include principal component analysis, with the selection of 'principal variables' (Cumming and Wooff, 2007) as an interesting variant, the LASSO, originally of Tibshirani (1996) and since implemented in a variety of forms, and the Dantzig method (Candes and Tao, 2007), which is specifically designed for situations with very large $d > n$.

Although such procedures may alleviate the curse of dimensionality greatly, there are certain limits to what they can achieve. Firstly, even after successful application of such a technique, one will remain with a subset of 'relevant' variables. The remaining local linear regression problem may still suffer from the curse of dimensionality, which begins to have an impact in dimensions as little as $d = 3$ or 4. Furthermore, most such variable selection or dimension reduction techniques will make some implicit linear modelling assumption, which may not be adequate from a nonparametric perspective. In order to deal with this problem properly, one would need to interweave the bandwidth selection and the variable selection processes, as suggested by Cleveland and Devlin (1988). Recently, an interesting approach in this direction was provided by Lafferty and Wasserman (2008), who introduced the *rodeo* (regularization of derivative expectation operator). This technique initially assigns a large bandwidth in every covariate direction. The bandwidths are then gradually decreased, and variables are deemed irrelevant if this does not lead to a substantial change in the estimated regression function. This concept of 'relevance' is not without controversy; for instance, Vidaurre, Bielza and Larrañaga (2011) argue that this definition is somewhat strange, since all variables which have a linear impact onto the response would be deemed irrelevant by construction. In an alternative approach, Vidaurre, Bielza and Larrañaga (2011) implemented a lasso locally to reduce the number of variables in local regression.

These hybrid bandwidth/variable selection methods were demonstrated to work reliably in certain situations, but they clearly carry some non–canonical features. In this work, we return to a more basic setup in which all estimation is carried out by 'standard' multivariate local linear regression in $d-$dimensional space. In the developments that follow, it is irrelevant whether the $d-$variate data set corresponds to the original data, or is the result of a dimension–reducing pre–processing step. In terms of the magnitude of d, we have quite large (say, up to two dozen), but not huge, numbers of variables in mind. We comment on the case of very large dimensions d in the Discussion.

Hastie, Tibshirani and Friedman (2001) and Cleveland and Devlin (1988) agree that the way to overcome the curse of dimensionality would be to increase the sample size n in order to capture complexities in the regression surface that might otherwise be lost through the necessary introduction of larger bandwidths. Of course, increasing n is often not a realistic option for a given data set, but, putting their statement in other words, there must be sufficient data around $\mathbf{x}$ for a reliable estimate to be made at that point. This is the attitude adopted in this paper, and in Section 2 we describe a solution which essentially identifies such "reliable" regions by dismissing all neighbourhoods which do not contain enough data. The actual smoothing step is then only performed over such regions in which estimation is considered reliable,

where the bias and variance of $\hat{m}$ can be kept reasonably low. This is achieved through a threshold imposed on a suitable estimate of the density f. In Section 3 a bandwidth matrix selection procedure is suggested which specifically tailors generalized cross-validation, first developed by Craven and Wahba (1979), for use with multivariate data. A simulation study is included to demonstrate the value of this technique, before finishing the paper with a Discussion in Section 4.

2 A density threshold

The basic idea is to identify regions which are suitable for local regression estimation by looking at the density, f. Since this is unknown, it needs to be estimated. At a multivariate point $\mathbf{x} \in \mathbb{R}^d$, the kernel density estimate, $\hat{f}(\mathbf{x})$, is

$$\hat{f}(\mathbf{x}) = n^{-1} \sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{X}_i - \mathbf{x}) \tag{7}$$

where $K_{\mathbf{H}}$ is a multivariate kernel function, as defined earlier, and again a bandwidth matrix $\mathbf{H}$ is needed. For reasons that will become clear later, we use the same $\mathbf{H}$ in calculating $\hat{f}$ as in the regression step. A threshold T is sought such that, if at point $\mathbf{x}$ we have $\hat{f}(\mathbf{x}) \geq T$, then an estimate using local linear regression can be considered somewhat reliable, and otherwise, care should be taken and an alternative method sought. Intuitively, T should depend on n and $\mathbf{H}$, as decreasing either of them will reduce the number of data points which are locally available at $\mathbf{x}$, requiring in turn a larger threshold to allow reliable estimation.

There seems to exist a tempting shortcut solution to the problem. One could argue that, for sufficient local estimation of a hyperplane with $p = d+1$ parameters, one needs effectively p pieces of information in the neighbourhood of $\mathbf{x}$. In other words, the observations in the vicinity of $\mathbf{x}$ need to contribute p times the information which would be provided by a data point situated *exactly* at $\mathbf{x}$. Using (7), this means that an initial candidate threshold, say T_0, would take the form

$$T_0 = \frac{(d+1)K(\mathbf{0})}{n|\mathbf{H}|^{1/2}}, \tag{8}$$

which contains the quantities n and $|\mathbf{H}|$ in the denominator, as expected. Later in this section we will arrive, through a more rigorous justification than the above, at a threshold of the same shape as (8), but with the constant $d+1$ replaced by a more adequate one, which is of the same magnitude as $d+1$ only for small values of d.

We present a data set to illustrate the motives. This data set contains 9 variables on 14000 chamois, a species of goat-antelope, shot between the months of October and December between the years 1973-2009, in the Trentino area of the Italian Alps. The response is body mass and the 8 covariates are climate variables, age and elevation. A subset of size 12000 is used as training

data while a further 2000 observations act as test data. For each point in the test set the body mass is estimated using (3) and compared with the observed response values. Since d is relatively large, automatic bandwidth selection is computationally too intensive and so here the $h_j, j = 1, \ldots, 8$ are chosen as one fifteenth of the data range in each direction. The difficulties of bandwidth selection will be discussed at length in Section 3. In the first graph in Fig. 1 the difference between Y_i and $\hat{m}(\mathbf{X}_i)$ is plotted against $\hat{f}(\mathbf{X}_i)$, calculated using (7), for each of the 2000 test points. The effects of the curse of dimensionality are clearly visible in the way that large errors in estimating the regression surface exist on the left hand side of the graph which examines the lower $\hat{f}(\mathbf{X}_i)$ and so sparser regions of the data set. This is the area which would ideally be cut off. For this data set, with $p = 9$ parameters, one would obtain $T_0 = 9K(\mathbf{0})/n|\mathbf{H}|^{1/2} = 1.13 \times 10^{-7}$. If this was employed as threshold T, then 599 of the test points would be considered to have a large enough $\hat{f}(\mathbf{X}_i)$ for regression to be reliable. The second plot in Fig. 1 again plots $\hat{m}(\mathbf{X}_i) - Y_i$ against $\hat{f}(\mathbf{X}_i)$, but only for these denser 599 points. However, this threshold is considered inadequate since errors of magnitude as large as 200 are observed at points where regression would be considered feasible, when the response range is approximately 40. For this reason we develop T differently, using the concept of influence.

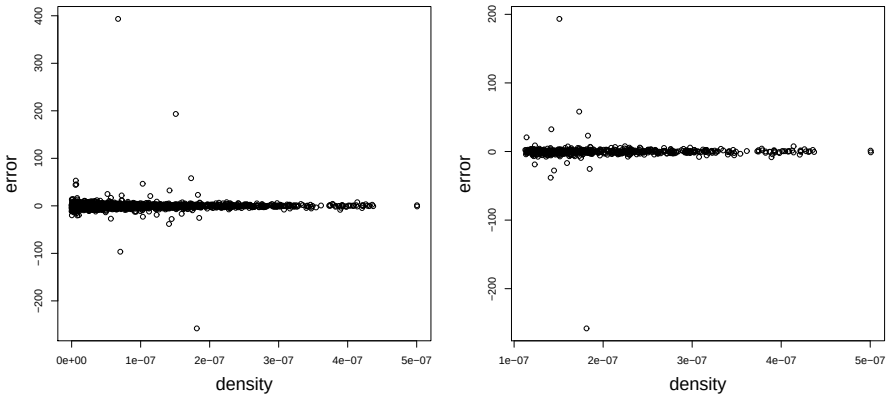


Fig. 1 The graph on the left shows $\hat{m}(\mathbf{X}_i) - Y_i$ v. $\hat{f}(\mathbf{X}_i)$ for all 2000 test points of the chamois data, and the graph on the right shows this for the 599 test points at which $\hat{f}(\mathbf{X}_i)$ is greatest

The influence, $\text{infl}(\mathbf{X}_i)$, describes the contribution of observation $\mathbf{X}_i$ to the estimation at $\mathbf{x} = \mathbf{X}_i$. It is given by the diagonal element of the ith row of the

smoother matrix $\mathbf{S}$, where $(\hat{m}(\mathbf{X}_1), \ldots, \hat{m}(\mathbf{X}_n))^T = \mathbf{SY}$, i.e.

$$\begin{aligned}\text{infl}(\mathbf{X}_i) &= \mathbf{e}_1^T(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\tilde{\mathbf{e}}_i = \mathbf{e}_1^T(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\begin{pmatrix} K_{\mathbf{H}}(\mathbf{0}) \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ &= |\mathbf{H}|^{-1/2}\mathbf{e}_1^T(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{e}_1 K(\mathbf{0}), \end{aligned} \tag{9}$$

where $\mathbf{X} = \mathbf{X}_{\{\mathbf{x}=\mathbf{X}_i\}}$, $\mathbf{W} = \mathbf{W}_{\{\mathbf{x}=\mathbf{X}_i\}}$, $\tilde{\mathbf{e}}_i$ is a vector of length n with 1 in the ith position, and $(K_{\mathbf{H}}(\mathbf{0}), 0, ..., 0)^T$ is a vector of length $d+1$.

It seems a sensible approach to dismiss local regression at observations for which $\text{infl}(\mathbf{X}_i)$ is very large. In the search for a criterion which identifies what a "very large influence" means in this context, we use Theorem 2.3 in Loader (1999), which states that the inequality

$$\text{infl}(\mathbf{X}_i) \leq 1 \tag{10}$$

holds at all observation points $\mathbf{X}_i$.

In order to relate the influence function to the density, we develop an asymptotic version of (9). At $\mathbf{x} \in \mathbb{R}^d$, let f be continuously differentiable and $f(\mathbf{x}) > 0$. Assuming $n^{-1}|\mathbf{H}|^{-1/2} \longrightarrow 0$ as $n \longrightarrow \infty$, one can show that

$$\text{infl}(\mathbf{x}) \approx \frac{\rho\, K(\mathbf{0})}{nf(\mathbf{x})|\mathbf{H}|^{1/2}} + o_p(n^{-1}|\mathbf{H}|^{-1/2}), \tag{11}$$

where

$$\rho = \left[\int K(\mathbf{u})d\mathbf{u} - \int \mathbf{u}^T K(\mathbf{u})d\mathbf{u} \left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)^{-1} \int \mathbf{u}K(\mathbf{u})d\mathbf{u}\right]^{-1}. \tag{12}$$

See Appendix A for the derivation of this result. Although the original definition of influence, (9), only applies at the observed values $\mathbf{X}_i$, the asymptotic influence function given by (11) can be computed at every $\mathbf{x}$. It can be seen as the influence which would be expected under idealized (asymptotic) conditions for a (hypothetical) data point situated at $\mathbf{x}$. Similarly, the inequality (10) applies only to the observed valued $\mathbf{X}_i$. However, due to the implicit averaging process happening in the computation of the asymptotic influence function, any $\mathbf{x}$ which is situated in between or close to data points $\mathbf{X}_i$ is still likely to possess the property $\text{infl}(\mathbf{x}) \leq 1$. In other words, in populated regions of the predictor space, the asymptotic influence will be less than 1, while it will exceed 1 in very sparse or remote regions. Using this rationale, it makes sense to define T by bounding the asymptotic influence by 1;

$$\frac{\rho K(\mathbf{0})}{nf(\mathbf{x})|\mathbf{H}|^{1/2}} \leq 1$$

so

$$f(\mathbf{x}) \geq \frac{\rho K(\mathbf{0})}{n|\mathbf{H}|^{1/2}}$$

and, hence,

$$T = \frac{\rho K(\mathbf{0})}{n|\mathbf{H}|^{1/2}}, \tag{13}$$

which is of the same form as (8) but with $d+1$ replaced by ρ.

Let us firstly note that the bandwidth matrix, $\mathbf{H}$, featuring in this *density* threshold stems from an expression involving the influence of the *regression*, which explains our earlier statement that the bandwidth matrix used for the density estimation should be the same as that used in the actual regression step.

Of great importance are the limits used in the integrals in ρ. If one estimates at an interior point, then these integral limits would range from $-\infty$ to ∞. For a boundary point, the lower integral limit would need to be altered according to the distance to the boundary (for instance, assuming a diagonal bandwidth matrix and $f(\mathbf{x}) > 0$, then if $\mathbf{x}$ is half a bandwidth h_j away from the boundary of the support of f in each coordinate direction, then the lower limit of each integral would be -0.5; for a rigorous definition of boundary points see Ruppert and Wand (1994)). This is of crucial importance for us since the boundary region, where data become sparse, is just the region in which we are interested. Hence, in order to represent the true influence as accurately as possible in the area of interest, we replace the lower integral limit by a small negative value, say a, which reflects the distance between the boundary of f and the area for which the criterion is optimized (notice that the integrals in (12) are $d-$variate, but we always use the same a for each coordinate direction). A choice of $a = 0$ would optimize the threshold for use at the edge of the data range, while a value of $a = -\infty$ would be best for use in the interior. For us, a value in between is optimal, to assess reliably the region where there is doubt over the validity of local linear regression as a suitable regression technique.

In fact, ρ varies quite strongly with a, as is shown in the top left plot in Fig. 2 in the case $d = 3$. This plot suggests that a value of a between -0.5 and -1 is approximately the point where ρ stabilises as a moves away from 0, which makes this a logical range to choose a from.

Our primary method of determining a suitable value of a has been to work backwards and look directly at the data by examining the error of estimated points. To illustrate this strategy, we generate two training data sets by subjecting the functions

$$m_3(x_1, x_2, x_3) = -12\cos(x_1) + 5\sin(5x_2) + 10\log(x_3) + 17$$

and

$$m_5(x_1, x_2, x_3, x_4, x_5) = m_3(x_1, x_2, x_3) + \cos(3x_4) + 7\tan(x_5)$$

to independent Gaussian noise, whereby $x_1, \ldots, x_5$ were generated from appropriately centered $t-$distributions with 2 df. These training data sets, each of size $n = 300$, were used to fit the respective local linear models. Test data sets, also of size $n = 300$, were generated in the same manner from m_3 and m_5. Fig. 2 (right) examines the MSEs for these test data (top: $d = 3$; bottom:

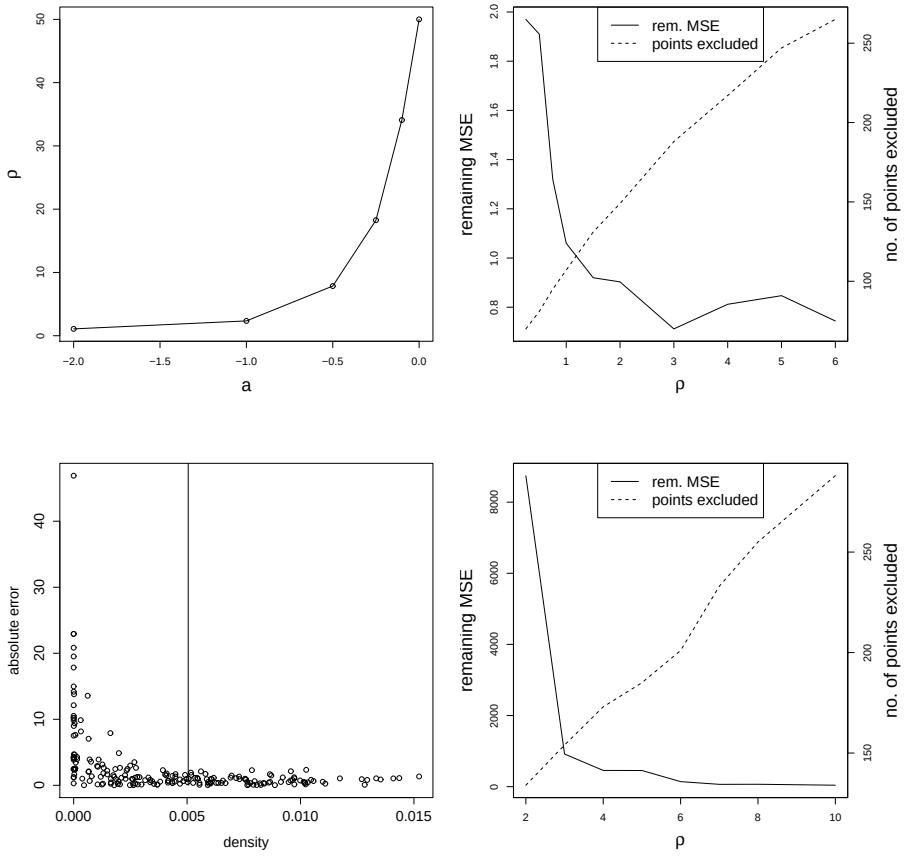


Fig. 2 Top left: ρ v. a; Bottom left: $|m(\mathbf{X}_i) - \hat{m}(\mathbf{X}_i)|$ v. $\hat{f}(\mathbf{X}_i)$ for the trivariate simulation; Right: MSE for points accepted by T, and number of points excluded, as a function of ρ, for the trivariate simulation (top) and the five-dimensional simulation (bottom)

$d = 5$), but only including those points accepted by threshold (13). The two plots show initially quite a steep decline in the MSE, but then flatten at a certain value of ρ. One also observes from these figures that the number of *excluded* points increases quite linearly with ρ. Once all badly fitting points are excluded, the exclusion of further points does not continue to improve the fit. As one wishes to exclude as few data as possible, it is important to find a value of ρ situated shortly after the steep descent in the MSE(ρ) function. Hence, an adequate integral limit a should correspond approximately to these values of ρ (which depend on d). In both plots, at these values of ρ, approximately one third of the test points are considered adequate, which seems like a reasonable proportion.

The bottom left plot shows the absolute error, $|m(\mathbf{X}_i) - \hat{m}(\mathbf{X}_i)|$, against $\hat{f}(\mathbf{X}_i)$ for the trivariate simulation. The vertical line in this plot is T with $a = -0.85$, and this is approximately where the threshold should cut in order to exclude the large errors associated with low density. Similar analyses were carried out for a variety of real and simulated data sets of varying dimension, and the value of $a = -0.85$ performed consistently well in these analyses, regardless of d. A value of $a = -0.85$ gives $\rho = 3.12$ for trivariate, and $\rho = 6.1$ for five-dimensional, data. Indeed, in the plots on the right hand side the curves seem to flatten at approximately these values of ρ, suggesting that any further increase in T would be pointless.

There is no theoretical argument that would tell us exactly where the threshold should cut. The most important aim is that extreme estimates, or points at which estimation breaks down computationally, are ruled out by the threshold. These analyses suggest that by making $a = -0.85$, (13) is capable of achieving this. This value corresponds to a point situated $0.85h_j$ inside the boundary, which is quite intuitive since this is approximately the region where one would assume data sparsity to become a problem. An attractive feature of threshold (13) is its interpretability, since (13) is neat in the sense that it takes the form of a multiple of the density of one point. The threshold is effectively imposing a required equivalent number of data points at $\mathbf{x}$. Applying this threshold to the chamois data gives $T = 1.93 \times 10^{-7}$, which means only 273 of the test points are considered to have large enough $\hat{f}(\mathbf{X}_i)$ for regression, via (3), to be reliable. As Fig. 3 illustrates, this is a considerable improvement compared to the residual pattern obtained in Fig. 1, with all unreasonably large errors now being eliminated.

Table 1 Comparing the number of parameters in the regression, p, with the corresponding value of ρ for different dimensions

Dimension	p	ρ	Dimension	p	ρ
1	2	1.50	9	10	20.41
2	3	2.19	10	11	27.82
3	4	3.12	11	12	36.94
4	5	4.46	12	13	51.13
5	6	6.10	13	14	68.72
6	7	8.35	14	15	88.72
7	8	11.22	15	16	110.49
8	9	15.34	16	17	147.30

Table 1 gives values of ρ for $d \leq 16$ as well as the number of parameters required, p, for each dimension. These values are data–independent; so the table can be used for general reference. The values for p and ρ are similar in lower dimensions, but it is for $d > 7$ that they differ more significantly, and, as shown with the chamois data, p is too small. This data demonstrates the merits of (13), a threshold which increases substantially in higher dimensions.

It should be noted that these values are all calculated using the lower integral limit $a = -0.85$. Our rationale for the selection of this value of a,

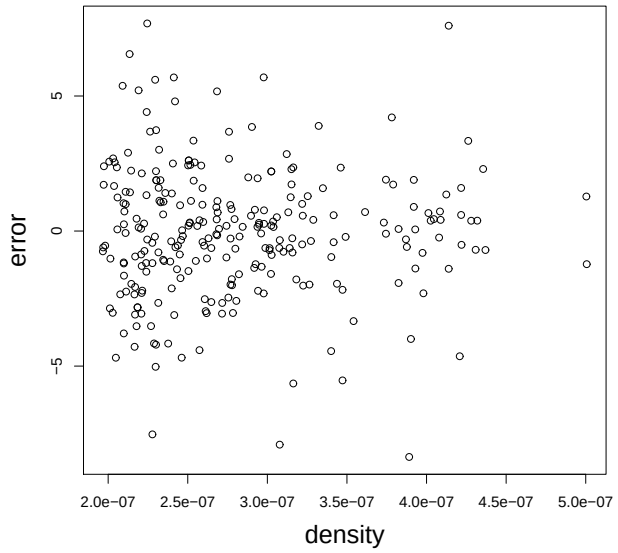
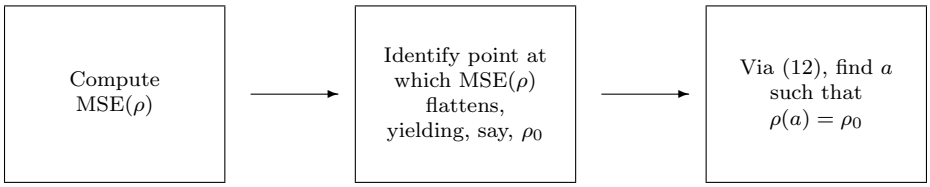


Fig. 3 $\hat{m}(\mathbf{X}_i) - Y_i$ v. $\hat{f}(\mathbf{X}_i)$ for the 273 test points accepted by (13), with $a = -0.85$, in the chamois data

as outlined before, is summarized in diagrammatic form in Fig. 4. However, we discourage the user from attempting to adjust a for each given data set. Firstly, the process is cumbersome, and the MSE will only be available for a simulated data set. Secondly, and more importantly, the suggested value of a is based on a careful mixture of theoretical arguments, heuristic considerations, and experimental results from many trials, which would be difficult to tune even further. Hence, our recommendation is clearly to bypass this step, and work directly with the values of ρ provided in Table 1.

Fig. 4 Schematic diagram illustrating our rationale for the choice of the lower integral limit a



3 AGCV

3.1 Adaptations to GCV

Bandwidth selection is also influenced detrimentally by the curse of dimensionality, so that appropriate measures need to be taken also at this stage. The first question arising is which family of bandwidth selection techniques should be used at all. While asymptotic bandwidth selection criteria have been found to work well in the univariate case, the assumption of bandwidths tending to 0 seems inappropriate in the multivariate context, where the bandwidths needed are relatively large. This was less of an issue in the previous section, where asymptotics were solely used to find an approximation of the influence function, but it is obviously an issue here as the goal is now bandwidth selection itself. Therefore, we focus on generalized cross-validation (GCV), developed by Craven and Wahba (1979), due to its relative computational ease and the fact that, unlike some competing methods, it does not rely on asymptotics. The criterion takes the form

$$GCV(\mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{Y_i - \hat{m}_{\mathbf{H}}(\mathbf{X}_i)}{1 - \frac{\text{trace}(\mathbf{S})}{n}}\right\}^2. \tag{14}$$

GCV struggles greatly to cope with high dimensional data, even when a diagonal bandwidth matrix is used, which we will assume throughout this section. GCV suggests the bandwidth matrix $\mathbf{H} = \text{diag}(h_1^2, \ldots, h_d^2)$ which minimizes (14), and it is the actual minimization process which causes computational problems.

When using R to carry out the minimization (R Development Core Team, 2010), computation of this minimum frequently breaks down entirely and an error message is returned. When this is not the case, often this process is very sensitive to the starting point of the minimization algorithm, and different optimal parameters are suggested depending on the starting point. Even if these problems are overcome and a reasonable looking selection is made, often the chosen bandwidth matrix performs poorly, and extreme values will be suggested frequently for the h_j, much larger than even the data range.

To alleviate these problems we propose two steps which remove the influence of data points in less dense areas, which otherwise may have a disproportionate impact on the procedure. Both steps are important in ensuring the technique is as robust as possible to the issues surrounding high–dimensional data. Firstly, we propose using the median, ψ, of the diagonal elements of $\mathbf{S}$, in the place of $\text{trace}(\mathbf{S})/n$. In practice, this prevents extremely large values of h_j being chosen. The effect of this adaptation alone is shown graphically in Fig. 5. Both plots show the GCV surface for a bivariate data set, simulated from a t-distribution with 1.3 degrees of freedom in order to create some very sparse areas of data. The first plot shows the unaltered GCV decreasing as the h_j increase, explaining why in this case the GCV minimization process chooses extremely high h_j. The second plot shows how using ψ gives the opposite result, with a clear minimum. As is visible, the minimum here occurs in a region

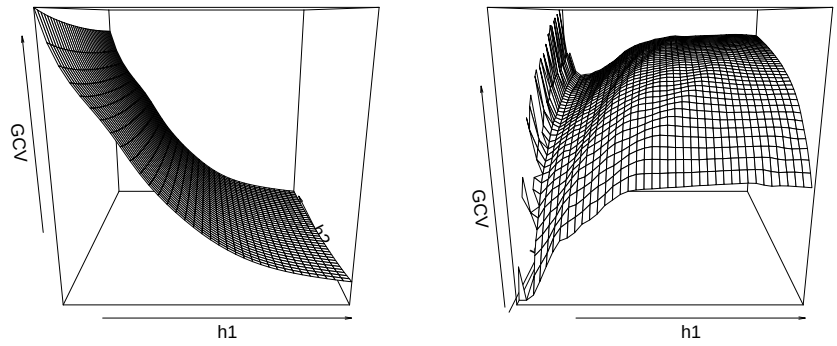


Fig. 5 The effect of replacing trace($\mathbf{S}$)/n by ψ on the GCV surface of a simulated bivariate data set. The first plot shows the GCV calculated using trace($\mathbf{S}$)/n and the second plot using ψ

of computational instability, but importantly, this minimum, captured by the proposed median–based version of GCV, is located in a sensible region of the parameter space.

The second adaptation we propose is removing isolated points completely from the process. An isolated point, in this context, is one at which no point other than itself contributes to its local regression estimate. Often an isolated point will impose a computational constraint on the minimization process. In the numerator of GCV, and within the diagonal elements of $\mathbf{S}$, is $\hat{m}_{\mathbf{H}}(\mathbf{X}_i)$, which is very sensitive to the bandwidths h_j. It is computationally impossible to compute $\hat{m}_{\mathbf{H}}(\mathbf{X}_i)$ at an isolated $\mathbf{X}_i$ if the h_j are not sufficiently large to make the point *not isolated*. This means that the solution space of the minimization problem is restricted to those bandwidths which are large enough to avoid computational error. In effect, the isolated points are enforcing minimum bandwidths (spanning the distance to their nearest neighbours), which are in fact far higher than the optimal h_j for the majority of the data. Therefore, we eliminate those r points at which $\hat{f}(\mathbf{X}_i)$ is smallest, by allocating them a weight $w(\mathbf{X}_i) = 0$, and $w(\mathbf{X}_i) = 1$ otherwise. The value r should be large enough to eliminate at least all isolated points, which will be explained in more detail below.

Applying these two adaptations to GCV, we formulate *adapted generalized cross-validation* (AGCV) which is defined as follows;

$$AGCV(\mathbf{H}) = n^{-1} \sum_{i=1}^{n} \left\{ \frac{Y_i - \hat{m}_{\mathbf{H}}(\mathbf{X}_i)}{1 - \psi_w} \right\}^2 w(\mathbf{X}_i) \tag{15}$$

where ψ_w is the median of the diagonal elements of the smoother matrix, $\mathbf{S}$, after excluding the elements contributed by the $\mathbf{X}_i$ for which $w(\mathbf{X}_i) = 0$.

To demonstrate the effect of these measures, we present a simple example. Consider a simulated five-dimensional dataset, of size $n = 300$, simulated through a t-distribution with 1 degree of freedom. The response values are generated according to the model $m(\mathbf{X}_i) = m_5(X_{i1}, X_{i2}, X_{i3}, X_{i4}, X_{i5})$ and noise $\epsilon_i \sim N(0, 1)$. An altered GCV containing the median, but without the isolated points removed, is minimized with h_j values of $(21.1, 3.45, 11.1, 0.8, 50.9)$, and here the selection of h_1, h_3 and h_5 in particular, is adversely affected by the restrictions caused by the points in less dense areas. If the 100 data points at which the density is smallest are removed from the procedure, equivalent to taking $r = 100$ in AGCV, then the AGCV criterion can be minimized at $(h_1, \dots h_5) = (2.5, 4.5, 2.4, 0.4, 1.6)$, which are parameters of a more reasonable magnitude, given the range of the majority of the data.

Removing points is both a matter of removing any computational constraint imposed by points in sparser regions, and also fine-tuning by focusing on the denser region in which we are interested. Any points excluded from AGCV should be outside the region of acceptability defined by (13). In this way AGCV is tailored towards finding optimal h_j for the areas accepted by T. Choosing r is effectively choosing a *pilot region* in which local polynomial regression is considered feasible. As a rule of thumb, we recommend setting r as the number of points at which the density is equal to the density estimate for just one data point, $n^{-1}|\mathbf{H}|^{-1/2}K(\mathbf{0})$. This density estimate should be calculated using the Epanechnikov kernel $K(u) = \frac{3}{4}(1 - u^2) \cdot 1_{\{-1 \le u \le 1\}}$ in (7), since this kernel, due to its truncated nature, identifies isolated points more clearly, compared with a Gaussian kernel. The bandwidth parameters to be used in (7) can be obtained through standard routines such as Scott's rule (Scott, 1992). We used the *npudensbw* function in the *np* package by Hayfield and Racine (2008) in R, which employs least-squares cross validation using the method of Li and Racine (2003).

It is possible to choose r higher than this rule–of–thumb suggests, which would lead to h_j values optimal for a denser part of the data range. A choice of r which includes *exactly* the points accepted by T would be ideal since this would then provide the best regression estimates at those points. However, since T is dependent on the h_j an optimal r cannot be chosen, and the simple rule of thumb, specified above, acts as an effective method of selecting r.

3.2 AGCV as a measure of error

GCV, as introduced by Craven and Wahba (1979), is the average squared error corrected by a factor.

$$GCV(\mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{Y_i - \hat{m}_{\mathbf{H}}(\mathbf{X}_i)}{1 - \frac{\text{trace}(\mathbf{S})}{n}}\right\}^2 = ASR(\mathbf{H})\left(1 - \frac{\text{trace}(\mathbf{S})}{n}\right)^{-2}, \quad (16)$$

where

$$ASR(\mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n}\left\{Y_i - \hat{m}_{\mathbf{H}}(\mathbf{X}_i)\right\}^2. \quad (17)$$

This is shown in Craven and Wahba (1979) as being effective in finding an estimate of the smoothing parameter that minimizes the mean squared error. Now

$$AGCV(\mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{Y_i - \hat{m}_{\mathbf{H}}(\mathbf{X}_i)}{1-\psi_w}\right\}^2 w(\mathbf{X}_i) = AWSR(\mathbf{H})(1-\psi_w)^{-2} \quad (18)$$

with the *average of weighted squared residuals*,

$$AWSR(\mathbf{H}) = \frac{1}{n}\sum_{i=1}^{n}\left\{Y_i - \hat{m}_{\mathbf{H}}(\mathbf{X}_i)\right\}^2 w(\mathbf{X}_i). \quad (19)$$

So AGCV is the average of weighted squared residuals, corrected by a factor. The factors used in (16) and (18) both calculate an average over the diagonal of $\mathbf{S}$ and subtract it from 1. The factor used in the AGCV is simply more robust. The other difference between the GCV and the AGCV is that the AGCV approximates the average *weighted* squared residual rather than the *unweighted*. Again, this is used to make the procedure more robust. In this way, AGCV can be justified as a legitimate proxy for the mean squared error, since it works in the same way as GCV, but in a more robust manner.

We finally note that, in principle, weight functions other than the strict zero/one–valued weights could be used, in which case ψ_w would be the weighted median of the diagonal elements of $\mathbf{S}$; although for our purposes there seems to be little benefit in doing so.

3.3 Simulation study

A rigorous simulation was carried out to measure the performance of AGCV against other bandwidth selection tools for multivariate data. Two trivariate data sets were generated.

- P— 3-dimensional covariates simulated through a t-distribution with 5 degrees of freedom. The response values were generated according to the model $m(\mathbf{X}_i) = m_3(X_{i1}, X_{i2}, X_{i3})$ and $\epsilon_i \sim N(0,1), i = 1, ..., 250$.
- Q— 3-dimensional covariates simulated through a t-distribution with 1.5 degrees of freedom. The response values were generated according to the model $m(\mathbf{X}_i) = m_3(X_{i1}, X_{i2}, X_{i3})$ and $\epsilon_i \sim N(0,3), i = 1, ..., 250$.

The difference between the two data sets is that Q contains much sparser areas of data.

Each of these data sets was simulated 100 times and then the optimal smoothing parameters were calculated using four different methods; AGCV, GCV, least squares cross-validation (the default method in the *np* package) and GCV for thin plate splines. For the methods dependent on a starting point, this was chosen carefully to give each method the best chance of finding the optimal h_j. The MSE was then calculated using each set of smoothing parameters. The MSE was calculated both including all 250 points and for

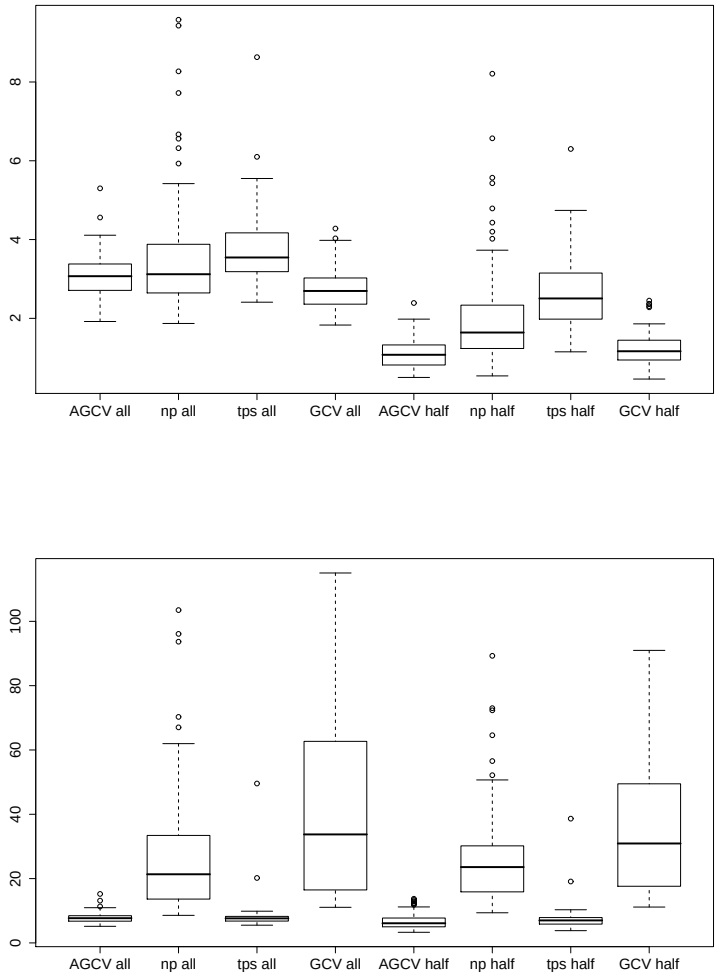


Fig. 6 The top plot is simulation P, and the bottom is Q. Each box plot represents the 100 MSEs for the simulation for different bandwidth selection techniques. *all* represents the MSE of all n points, and *half* represents the MSE for the densest 50 percent

just the densest 50 percent of each data set, since AGCV is tailored towards use in the denser areas of the data. One cannot use the threshold (13) here to compare the procedures since different h_j are selected for each method. Since the threshold T depends on the h_j, the density would exceed T at different points for each method, and so no fair comparison could be made. The density was measured using kernel density estimation tools in the *np* package. As shown in Fig. 6, AGCV consistently outperforms the other techniques with

a smaller median MSE. With the less sparse data, P, shown in the top plot, the AGCV and GCV perform best, with the AGCV performing better for the densest 50 percent. The *np* and thin plate spline methods have larger MSEs as well as larger interquartile ranges. With the sparser data, Q, shown in the bottom plot, the AGCV and thin plate splines are the only techniques whose MSEs could be considered of a reasonable size given the magnitude of the response values. Out of these, AGCV is marginally better with a smaller median, which again improves when only including the densest 50 percent of the data. The GCV and the *np* least squares cross-validation both perform extremely poorly on this sparser data.

3.4 Notes on computational issues

- Throughout this study we have used the *optim* function on R, which uses the Nelder-Mead algorithm, detailed in Nelder and Mead (1965), to carry out the minimization of the (A)GCV functions.
- AGCV is still sensitive to the starting point specified for *optim*, and a more successful minimization is more likely if this point is chosen with care. From experience it is observed that a starting point smaller than the actual minimum is often more successful, and it is sometimes helpful to perform the minimization more than once, using the result of the previous minimization as the new starting point. However, due to the nature of *optim* the selection of the overall minimum cannot be guaranteed. This is not a problem with AGCV itself, rather a problem of the minimizing technique selecting one of many minima, but not necessarily the smallest, as desired.
- Although AGCV is fast in comparison to GCV (using R), it is still time-consuming for $d > 6$, and a potential solution to this is to search for a constant $h = h_1 = ... = h_d$, after standardizing the covariates.

4 Discussion

We have proposed two relatively simple measures which enable local linear smoothing with high–dimensional data. The problem of "local neighbourhoods being not local", as usually reported in this setting, is circumvented by focusing on dense regions of the predictor space where reliable estimation, with relatively small bandwidths, is achievable. It was demonstrated how such a feasible region is identified through a simple criterion based on the asymptotic influence function. A multivariate version of GCV, which uses a pilot region to select a suitable diagonal bandwidth matrix, was also introduced.

The adjustments made to GCV here are made specifically in response to problems encountered on R. In spite of this, it fits perfectly with the general solution to the curse of dimensionality expressed in this work, of excluding the areas of low density from consideration. The points that are ignored in

AGCV are sufficiently isolated that they would never be accepted by T. In this way, the h_j selected by AGCV are more suited to the points accepted by the threshold, by not having to take into account other points excluded by it. This conclusion is supported by the strong performance of AGCV in the densest 50 percent of the data in the simulation study. Our overall attitude of ignoring data in sparse regions seems successful in creating a reliable smoothing strategy elsewhere.

A variable bandwidth matrix $\mathbf{H}(\mathbf{x})$, similar to that described for kernel density estimation by Sain (2002), could be beneficial for multivariate kernel regression too, but this would be computationally costly. AGCV can be seen as the first step towards a variable bandwidth matrix, in the sense that it selects bandwidths h_j suitable for a proportion of the data, determined by r.

It is important to reflect on the relevance of the techniques presented here to data sets of very high dimension. We have tested our methods successfully with data of dimension up to $d = 16$, and we did not identify an immediate obstacle which would prevent us from going further than that. However, as can be seen from Table 1, the parameter ρ increases very strongly for higher dimensions. While this increase has been demonstrated to be appropriate, often n will not be large enough to satisfy this threshold. In other words, for very large d, we are likely to encounter situations where practically the whole data set will be eliminated by the threshold, unless n is large enough to counterbalance the effect of ρ. Data sets which have such properties still do exist, and are most likely to be generated by numerical simulation from complex computer models.

One timely question would be whether the proposed techniques could deal with situations where $d > n$, as encountered for genomic data in computational biology, or even with functional data (Ramsay and Silverman, 1997), which can be considered to be of quasi-infinitely dimensional character. Though one may feel encouraged by results such as in Ferraty and Vieu (2006), who demonstrate how to adapt multivariate local *constant* regression to functional data, we hit an obstacle when taking the step to local *linear* regression: by construction, at least $n = d + 1$ data points are locally (and, hence, globally) needed to fit a hyperplane involving $d + 1$ parameters. Hence, for data sets of such large dimension d, some form of variable selection or dimension reduction, as mentioned in the introduction, is necessary, before the methods proposed in this paper can be applied. In the case of functional data, an attractive pre–processing tool which identifies the design points with the "greatest predictive influence", was suggested by Ferraty, Hall and Vieu (2010).

An issue that we did not discuss in this paper is the *shape* of the space $\mathcal{S}_T = \{\mathbf{x}|\hat{f}(\mathbf{x}) \geq T\}$. Generally, this space does not need to be either convex or connected, but we found it usually to be of a reasonably well–behaved shape (i.e., not consisting of lots of scattered pieces, etc.) in practice, provided that a sensible bandwidth is chosen for the initial density estimator. The space $\mathcal{S}_T$ will be compact by construction. This is an important property as it enables access to boundary measures for $\mathcal{S}_T$ (for instance, for $d = 3$, the surface area).

The reader is referred to a recent paper by Armendáriz, Cuevas, and Fraiman (2009) for recent advances in this respect.

Acknowledgements Many thanks to Marco Apollonio, of the Dept. of Zoology and Evolutionary Genetics, University of Sassari, and Tom Mason, of the School of Biological and Biomedical Sciences, University of Durham, for permitting us to use the chamois data. The authors are also grateful to two anonymous referees for their helpful comments, particularly for pointing us to interesting literature which enabled us to motivate our methods from a wider perspective than in the original version.

A Derivation of (11)

Take expression (9)

$$\text{infl}(\mathbf{X}_i) = |\mathbf{H}|^{-1/2}\mathbf{e}_1^T(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{e}_1 K(\mathbf{0})$$

where $\mathbf{X} = \mathbf{X}_{\{\mathbf{x}=\mathbf{X}_i\}}$ and $\mathbf{W} = \mathbf{W}_{\{\mathbf{x}=\mathbf{X}_i\}}$. Now

$$\mathbf{X}^T\mathbf{W}\mathbf{X} = \begin{bmatrix} \sum_{i=1}^n K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x}) & \sum_{i=1}^n K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})(\mathbf{X}_i-\mathbf{x})^T \\ \sum_{i=1}^n K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})(\mathbf{X}_i-\mathbf{x}) & \sum_{i=1}^n K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})(\mathbf{X}_i-\mathbf{x})(\mathbf{X}_i-\mathbf{x})^T \end{bmatrix}$$

Approximating each of these entries;

$$\sum_{i=1}^n K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x}) = E\left(\sum_{i=1}^n K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})\right) + O_p\left(\sqrt{\text{Var}\left(\sum_{i=1}^n K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})\right)}\right). \quad (20)$$

Since the $\mathbf{X}_i$ are i.i.d.

$$E\left(\sum_{i=1}^n K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})\right) = n\int K_{\mathbf{H}}(\mathbf{t}-\mathbf{x})f(\mathbf{t})d\mathbf{t} = n\int |\mathbf{H}|^{-1/2}K(\mathbf{H}^{-1/2}(\mathbf{t}-\mathbf{x}))f(\mathbf{t})d\mathbf{t}$$

Then using the substitution $\mathbf{u} = (u_1,...,u_d)^T = \mathbf{H}^{-1/2}(\mathbf{t}-\mathbf{x})$ and Taylor's theorem, one gets

$$\begin{aligned} n\int |\mathbf{H}|^{-1/2}K(\mathbf{u})f(\mathbf{x}+\mathbf{H}^{1/2}\mathbf{u})|\mathbf{H}|^{1/2}d\mathbf{u} &= n\int K(\mathbf{u})f(\mathbf{x}+\mathbf{H}^{1/2}\mathbf{u})d\mathbf{u} \\ &= n\left(f(\mathbf{x})\int K(\mathbf{u})d\mathbf{u} + o(1)\right). \end{aligned}$$

For the variance term,

$$\begin{aligned} \text{Var}\left(\sum_{i=1}^n K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})\right) &= n\left[E\left((K_{\mathbf{H}}(\mathbf{X}_1-\mathbf{x}))^2\right) - \left(E\left(K_{\mathbf{H}}(\mathbf{X}_1-\mathbf{x})\right)\right)^2\right] \\ &= n\left[\int |\mathbf{H}|^{-1}K^2(\mathbf{H}^{-1/2}(\mathbf{t}-\mathbf{x}))f(\mathbf{t})d\mathbf{t} - \left(\int |\mathbf{H}|^{-1/2}K(\mathbf{H}^{-1/2}(\mathbf{t}-\mathbf{x}))f(\mathbf{t})d\mathbf{t}\right)^2\right] \end{aligned}$$

Then using the same substitution as above

$$\begin{aligned}\mathrm{Var}\left(\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})\right) =& n|\mathbf{H}|^{-1/2} f(\mathbf{x})\left(\int K^2(\mathbf{u})d\mathbf{u}+o(1)\right)\\ &- n\left(f(\mathbf{x})\int K(\mathbf{u})d\mathbf{u}+o(1)\right)^2\\ =& n|\mathbf{H}|^{-1/2}\left[f(\mathbf{x})\int K^2(\mathbf{u})d\mathbf{u}+o(1)\right]\\ =& o(n^2)\end{aligned}$$

Using (20)

$$\begin{aligned}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x}) &= n\left(f(\mathbf{x})\int K(\mathbf{u})d\mathbf{u}+o(1)\right)+O_p\left(\sqrt{o(n^2)}\right)\\ &= n\left(f(\mathbf{x})\int K(\mathbf{u})d\mathbf{u}+o_p(1)\right). \qquad (21)\end{aligned}$$

Similarly

$$\begin{aligned}&E\left(\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})(\mathbf{X}_i-\mathbf{x})\right)\\ &= n\mathbf{H}^{1/2}\int \mathbf{u}K(\mathbf{u})f(\mathbf{x}+\mathbf{H}^{1/2}\mathbf{u})d\mathbf{u}\\ &= n\mathbf{H}^{1/2}\left[f(\mathbf{x})\int \mathbf{u}K(\mathbf{u})d\mathbf{u}+\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})\mathbf{H}^{1/2}(\nabla f(\mathbf{x})+o(\mathbf{1}))d\mathbf{u}\right]\\ &= n\mathbf{H}^{1/2}f(\mathbf{x})\int \mathbf{u}K(\mathbf{u})d\mathbf{u}+n\mathbf{H}^{1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)\mathbf{H}^{1/2}\nabla f(\mathbf{x})\,(1+o(1))\end{aligned}$$

and

$$\begin{aligned}&\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})(\mathbf{X}_i-\mathbf{x})\\ &= n\mathbf{H}^{1/2}f(\mathbf{x})\int \mathbf{u}K(\mathbf{u})d\mathbf{u}+n\mathbf{H}^{1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)\mathbf{H}^{1/2}\nabla f(\mathbf{x})\,(1+o_p(1)). \qquad (22)\end{aligned}$$

Similarly

$$\begin{aligned}&\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})(\mathbf{X}_i-\mathbf{x})^T\\ &= nf(\mathbf{x})\int \mathbf{u}^T K(\mathbf{u})d\mathbf{u}\mathbf{H}^{1/2}+n\nabla f(\mathbf{x})^T\mathbf{H}^{1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)\mathbf{H}^{1/2}\,(1+o_p(1)). \qquad (23)\end{aligned}$$

Finally,

$$\begin{aligned}&E\left(\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{X}_i-\mathbf{x})(\mathbf{X}_i-\mathbf{x})(\mathbf{X}_i-\mathbf{x})^T\right)\\ &= n\int |\mathbf{H}|^{-1/2}K(\mathbf{H}^{-1/2}(\mathbf{t}-\mathbf{x}))(\mathbf{t}-\mathbf{x})(\mathbf{t}-\mathbf{x})^T f(\mathbf{t})d\mathbf{t}\\ &= n\int |\mathbf{H}|^{-1/2}K(\mathbf{u})\mathbf{H}^{1/2}\mathbf{u}(\mathbf{H}^{1/2}\mathbf{u})^T f(\mathbf{x}+\mathbf{H}^{1/2}\mathbf{u})|\mathbf{H}|^{1/2}d\mathbf{u}\\ &= n\mathbf{H}^{1/2}\left[\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)f(\mathbf{x})+o(\mathbf{1})\right]\mathbf{H}^{1/2}\end{aligned}$$

and

$$\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{X}_i - \mathbf{x})(\mathbf{X}_i - \mathbf{x})(\mathbf{X}_i - \mathbf{x})^T$$
$$= n\mathbf{H}^{1/2}\left[\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right) f(\mathbf{x}) + o_p(\mathbf{1})\right]\mathbf{H}^{1/2}. \tag{24}$$

So $\mathbf{X}^T\mathbf{W}\mathbf{X}$ can be written as

$$\begin{bmatrix} (21) & (23) \\ (22) & (24) \end{bmatrix} \tag{25}$$

For (9) one needs the top left entry of the inverse of (25). For a general block matrix $\mathbf{B}$, such as this one, The Matrix Cookbook (Petersen & Pedersen, 2008) states that this is equivalent to, $(\mathbf{B}_{11} - \mathbf{B}_{12}\mathbf{B}_{22}^{-1}\mathbf{B}_{21})^{-1}$. For (25),

$$\begin{aligned}
&(\mathbf{B}_{11} - \mathbf{B}_{12}\mathbf{B}_{22}^{-1}\mathbf{B}_{21}) \\
&= n\left(f(\mathbf{x})\int K(\mathbf{u})d\mathbf{u} + o_p(1))\right) - \\
&\left(nf(\mathbf{x})\int \mathbf{u}^T K(\mathbf{u})d\mathbf{u}\mathbf{H}^{1/2} + n\left(\nabla f(\mathbf{x})^T\mathbf{H}^{1/2}\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\mathbf{H}^{1/2}\right)(1 + o_p(1))\right)\times \\
&\left(n\mathbf{H}^{1/2}\left(\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right) f(\mathbf{x}) + o_p(\mathbf{1})\right)\mathbf{H}^{1/2}\right)^{-1}\times \\
&\left(n\mathbf{H}^{1/2}f(\mathbf{x})\int \mathbf{u}K(\mathbf{u})d\mathbf{u} + n\mathbf{H}^{1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)\mathbf{H}^{1/2}\nabla f(\mathbf{x})\,(1 + o_p(1))\right) \\
&= n\left(f(\mathbf{x})\int K(\mathbf{u})d\mathbf{u} + o_p(1)\right) - n\left(f(\mathbf{x})\int \mathbf{u}^T K(\mathbf{u})d\mathbf{u}\mathbf{H}^{1/2} + o_p(\mathbf{1}^T\mathbf{H}^{1/2})\right)\times \\
&\left(n\left(\mathbf{H}^{1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right) f(\mathbf{x})\mathbf{H}^{1/2} + o_p(\mathbf{H})\right)\right)^{-1}\times \\
&n\left(\mathbf{H}^{1/2}f(\mathbf{x})\int \mathbf{u}K(\mathbf{u})d\mathbf{u} + o_p(\mathbf{H}^{1/2}\mathbf{1})\right)
\end{aligned} \tag{26}$$

Within (26), defining a_n as a sequence $a_n = o_p(\mathbf{H})$, b_n as a sequence $b_n = o_p(\mathbf{1})$ and c_n as a sequence $c_n = O(\mathbf{H}^{-1})$ one uses the Kailath Variant from the Matrix Cookbook to re-express the inverse. The Kailath Variant states that $(\mathbf{A} + \mathbf{B}\mathbf{C})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} + \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1}$. Here, say $\mathbf{A} = \mathbf{H}^{1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right) f(\mathbf{x})\mathbf{H}^{1/2}$, $\mathbf{B} = a_n$ and $\mathbf{C} = \mathbf{I}$. Hence

$$\begin{aligned}
&\left(\mathbf{H}^{1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right) f(\mathbf{x})\mathbf{H}^{1/2} + o_p(\mathbf{H})\right)^{-1} \\
&= \left(\mathbf{H}^{1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right) f(\mathbf{x})\mathbf{H}^{1/2}\right)^{-1} - c_n a_n(\mathbf{I} + c_n a_n)^{-1}c_n \\
&= \left(\mathbf{H}^{1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right) f(\mathbf{x})\mathbf{H}^{1/2}\right)^{-1} - b_n c_n \\
&= \left(\mathbf{H}^{1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right) f(\mathbf{x})\mathbf{H}^{1/2}\right)^{-1} + o_p(\mathbf{H}^{-1}) \\
&= \mathbf{H}^{-1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)^{-1}(f(\mathbf{x}))^{-1}\mathbf{H}^{-1/2} + o_p(\mathbf{H}^{-1})
\end{aligned}$$

Replacing this in (26) one gets

$$
\begin{aligned}
&(\mathbf{B}_{11} - \mathbf{B}_{12}\mathbf{B}_{22}^{-1}\mathbf{B}_{21}) \\
&= n\left(f(\mathbf{x})\int K(\mathbf{u})d\mathbf{u} + o_p(1)\right) - n\left(f(\mathbf{x})\int \mathbf{u}^T K(\mathbf{u})d\mathbf{u}\mathbf{H}^{1/2} + o_p(\mathbf{1}^T\mathbf{H}^{1/2})\right)\times \\
&\frac{1}{n}\left(\mathbf{H}^{-1/2}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)^{-1}(f(\mathbf{x}))^{-1}\mathbf{H}^{-1/2} + o_p(\mathbf{H}^{-1})\right)\times \\
&n\left(\mathbf{H}^{1/2}f(\mathbf{x})\int \mathbf{u}K(\mathbf{u})d\mathbf{u} + o_p(\mathbf{H}^{1/2}\mathbf{1})\right) \\
&= n\left(f(\mathbf{x})\int K(\mathbf{u})d\mathbf{u} + o_p(1)\right) - \\
&\left(\int \mathbf{u}^T K(\mathbf{u})d\mathbf{u}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)^{-1}\mathbf{H}^{-1/2} + o_p(\mathbf{1}^T\mathbf{H}^{-1/2})\right)\times \\
&n\left(\mathbf{H}^{1/2}f(\mathbf{x})\int \mathbf{u}K(\mathbf{u})d\mathbf{u} + o_p(\mathbf{H}^{1/2}\mathbf{1})\right) \\
&= n\left[f(\mathbf{x})\left[\int K(\mathbf{u})d\mathbf{u} - \int \mathbf{u}^T K(\mathbf{u})d\mathbf{u}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)^{-1}\int \mathbf{u}K(\mathbf{u})d\mathbf{u}\right] + o_p(1)\right]
\end{aligned}
$$

Applying the inverse, one obtains an approximation for the top left entry of the inverse of (25)

$$
\begin{aligned}
&(\mathbf{B}_{11} - \mathbf{B}_{12}\mathbf{B}_{22}^{-1}\mathbf{B}_{21})^{-1} \\
&= (nf(\mathbf{x}))^{-1}\left[\int K(\mathbf{u})d\mathbf{u} - \int \mathbf{u}^T K(\mathbf{u})d\mathbf{u}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)^{-1}\int \mathbf{u}K(\mathbf{u})d\mathbf{u}\right]^{-1} + o_p(n^{-1})
\end{aligned}
$$

Substituting this in (9) gives the result

$$
\begin{aligned}
\mathrm{infl}(\mathbf{x}) =& \frac{K(\mathbf{0})}{nf(\mathbf{x})|\mathbf{H}|^{1/2}}\left[\int K(\mathbf{u})d\mathbf{u} - \int \mathbf{u}^T K(\mathbf{u})d\mathbf{u}\left(\int \mathbf{u}\mathbf{u}^T K(\mathbf{u})d\mathbf{u}\right)^{-1}\int \mathbf{u}K(\mathbf{u})d\mathbf{u}\right]^{-1} \\
&+ o_p(n^{-1}|\mathbf{H}|^{-1/2})
\end{aligned}
$$

The above calculations for the asymptotic approximation to $(\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}$ are more general compared to those in other sources, such as Ruppert and Wand (1994), since here the kernel moments are not assumed to vanish. This allows for non-symmetric kernels, as well as handling of boundary points.

References

1. Armendariz I, Cuevas A, Fraiman R (2009). Nonparametric estimation of boundary measures and related functionals: Asymptotic properties. *Adv. Appl. Prob.* **41**, 311–322.
2. Candes E, Tao T (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *Ann. Statist.* **35**, 2313–2351.
3. Cleveland WS, Devlin SJ (1988). Locally weighted regression: an approach to regression analysis by local fitting. *J. Amer. Statist. Assoc.* **83**, 596–610.
4. Craven P, Wahba G (1979). Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math.* **3**, 377–403.
5. Cumming JA, Wooff DA (2007). Dimension reduction via principal variables. *Computational Statistics and Data Analysis* **52**, 550-565.

6. Duchon J (1977). Splines minimizing rotation-invariant semi-norms in Sobolev spaces. *Constructive Theory of Functions of Several Variables,* W. Schempp and K. Zeller, eds., 85–100. Springer-Verlag, Berlin.
7. Fan J (1993). Local linear regression smoothers and their minimax efficiencies. *Ann. Statist.* **21**, 196–216.
8. Ferraty F, Hall P, Vieu P (2010). Most-predictive design points for functional data predictors. *Biometrika* **97**, 807–824.
9. Ferraty F, Vieu P (2006). *Nonparametric Functional Data Analysis.* Springer, New York.
10. Fowlkes EB (1986). Some diagnostics for binary logistic regression via smoothing (with discussion). *Proceedings of the Statistical Computing Section, American Statistical Association* **1**, 54–56.
11. Hastie T, Loader CR (1993a). Rejoinder to: "Local regression: Automatic kernel carpentry." *Statistical Science* **8**, 139–143.
12. Hastie T, Loader CR (1993b). Local regression: Automatic kernel carpentry. *Statistical Science* **8**, 120–129.
13. Hastie T, Tibshirani R (1990). *Generalized Additive Models.* Chapman and Hall, London.
14. Hastie T, Tibshirani R, Friedman J (2001). *The Elements of Statistical Learning.* Springer, New York.
15. Hayfield T, Racine JS (2008). Nonparametric Econometrics: The np package. *Journal of Statistical Software* **27**, 1–32.
16. Li Q, Racine JS (2003). Nonparametric estimation of distributions with categorical and continuous data. *Journal of Multivariate Analysis* **86**, 266–292.
17. Lafferty J, Wasserman L (2008). Rodeo: sparse, greedy nonparametric regression. *Ann. Statist.* **36**, 28–63.
18. Loader CR (1999). *Local Regression and Likelihood.* Springer, New York.
19. Nelder JA, Mead R (1965). A simplex method for function minimization. *Computer Journal* **7**, 308–313.
20. Petersen KB, Pedersen MS (2008). The Matrix Cookbook. *http://matrixcookbook.com/*.
21. R Development Core Team (2010). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
22. Ruppert D, Wand MP (1994). Multivariate locally weighted least squares regression. *Ann. Statist.* **22**, 1346–1370.
23. Sain SR (2002). Multivariate locally adaptive density estimation. *Computational Statistics and Data Analysis* **39**, 165–186.
24. Scott DW (1992). *Multivariate Density Estimation.* Wiley, New York.
25. Stone CJ (1980). Optimal rates of convergence for nonparametric estimators. *Ann. Statist.* **8**, 1348–1360.
26. Tibshirani R (1996). Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B* **58**, 267–288.
27. Vidaurre D, Bielza C, Larrañaga P (2011). Lazy lasso for local regression. *Computational Statistics*, DOI 10.1007/s00180-011-0274-0.