Computational Statistics and Data Analysis 62 (2013) 122–135

Contents lists available at SciVerse ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Sparse regularized local regression

Diego Vidaurre [a,*], Concha Bielza [b], Pedro Larrañaga [b]

[a] Oxford Centre for Human Brain Activity, Warneford Hospital, Department of Psychiatry, University of Oxford, UK

[b] Computational Intelligence Group, Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, Spain

ARTICLE INFO

Article history:
Received 26 January 2012
Received in revised form 17 December 2012
Accepted 15 January 2013
Available online 21 January 2013

Keywords:
Bandwidth selection
Kernel smoothing
Local linear regression
Multiple regression
Non-parametric regression
Variance reduction
Sparsity
Sparse estimation

ABSTRACT

The intention is to provide a Bayesian formulation of regularized local linear regression, combined with techniques for optimal bandwidth selection. This approach arises from the idea that only those covariates that are found to be relevant for the regression function should be considered by the kernel function used to define the neighborhood of the point of interest. However, the regression function itself depends on the kernel function. A maximum posterior joint estimation of the regression parameters is given. Also, an alternative algorithm based on sampling techniques is developed for finding both the regression parameter distribution and the predictive distribution.

1. Introduction

Consider p independent covariates $\{X_1, \ldots, X_p\}$ and a response variable Y. Let $\boldsymbol{X}$ and $\boldsymbol{y} = (y_1, \ldots, y_N)^t$ be, respectively, an $N \times p$ data matrix and a continuous-valued vector, so that each row $\boldsymbol{x}_i$ is i.i.d. related to a continuous response y_i by means of some unknown (nonlinear) function $m(\cdot)$:

$$y_i = m(\boldsymbol{x}_i) + e_i,$$

where $m(\cdot)$ is assumed to be sparse and have continuous second-order derivatives and e_i is the irreducible error term, with $\mathrm{E}[e_i|\boldsymbol{x}_i] = 0$. Therefore, $\mathrm{E}[y_i|\boldsymbol{x}_i] = m(\boldsymbol{x}_i)$. We denote the elements and the columns of $\boldsymbol{X}$, respectively, as x_{ij} and $\boldsymbol{X}_{\cdot j}$.

The objective is to estimate the response at a point of interest $\boldsymbol{x} = (x_1, \ldots, x_p)^t$ using a sparsity assumption: only a subset of the covariates is indeed relevant for the estimation. We denote as $\boldsymbol{X}^*$ the data matrix $\boldsymbol{X}$ centered at $\boldsymbol{x}$ and augmented with a first column of ones, so that $x^*_{i0} = 1$ for all $i = 1, \ldots, N$. In our approach, the homoscedasticity assumption is not strictly necessary, so we can generically define the variance of e_i as $\mathrm{Var}[e_i|\boldsymbol{x}_i] = s^2(\boldsymbol{x}_i) = \sigma_i^2$. However, as will be apparent below, we make the homoscedasticity assumption $\sigma_i^2 = \sigma^2, \forall i$, for computational reasons.

Multivariate local regression (Cleveland and Devlin, 1988; Loader, 1999) estimates a multivariate regression function valid for some neighborhood of $\boldsymbol{x}$. This function is often linear, corresponding to a first-order Taylor approximation of $m(\cdot)$ at $\boldsymbol{x}$, and it can be defined on the original covariates or on some set of basis functions defined on the original covariates. We consider for simplicity the first case, although the generalization is straightforward. Local regression is appealing from both theoretical and practical sides. On the one hand, it is known to enjoy 100% minimax efficiency for some choice of bandwidth

* Corresponding author. Tel.: +44 0 1865 289300.
E-mail addresses: diego.vidaurre@ohba.ox.ac.uk (D. Vidaurre), mcbielza@fi.upm.es (C. Bielza), pedro.larranaga@fi.upm.es (P. Larrañaga).

0167-9473/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2013.01.008

and kernel (Fan, 1993; Ruppert and Wand, 1994). On the other hand, it is computationally fast, easy to implement, flexible, and robust to data design (Hastie and Loader, 1993).

The neighborhood is defined by a kernel function, which assigns weights $\boldsymbol{w} = (w_1, \ldots, w_N)^t$ to the data points in the data set on the grounds of their distance to $\boldsymbol{x}$. The kernel function has a bandwidth parameter, which strongly influences the estimation. High bandwidths increase the bias and decrease the variance of the estimate; low bandwidths do the opposite. The simplest approach is to use a single-value bandwidth for all regressors, and the most general setting is to use a full-matrix bandwidth, which provides flexible smoothing on all orientations. While the first usually leads to a severely biased estimate, the second can imply the estimation of a large number of parameters. A convenient compromise is a bandwidth vector or diagonal bandwidth, denoted as $\boldsymbol{h} = (h_1, \ldots, h_p)^t$, which permits adaptive smoothing at each coordinate direction. The kernel function is defined as

$$w_i^2 = K_{\boldsymbol{h}}(\boldsymbol{x}_i - \boldsymbol{x}) = \prod_{j=1}^{p} \frac{1}{h_j} K\left(\frac{x_{ij} - x_j}{h_j}\right) = \prod_{j=1}^{p} \frac{1}{h_j} K\left(\frac{x_{ij}^*}{h_j}\right), \quad i = 1, \ldots, N, \tag{1}$$

where $K(\cdot)$ is a univariate, symmetric, and non-negative function with a compact support, such that $\int K(t)dt = 1$. In this paper, we use the well-known Gaussian kernel $K(t) = (2/\pi)^{1/2} \exp(-t^2/2)$.

Then, the estimated local linear regression function $g(\cdot)$ is defined by a vector of local regression coefficients $\hat{\boldsymbol{\beta}}(\boldsymbol{x}) = (\hat{\beta}_1(\boldsymbol{x}), \ldots, \hat{\beta}_p(\boldsymbol{x}))^t$ and an intercept term $\hat{\beta}_0(\boldsymbol{x})$. For simplicity of notation, in the following we denote, unless it is necessary to do otherwise, $\hat{\beta}_0(\boldsymbol{x})$ as $\hat{\beta}_0$ and $\hat{\boldsymbol{\beta}}(\boldsymbol{x})$ as $\hat{\boldsymbol{\beta}}$. Then, we have

$$\hat{y}_i = g(\boldsymbol{x}_i) = \hat{\beta}_0 + \boldsymbol{x}_i^t \hat{\boldsymbol{\beta}}.$$

Since the data is centered at $\boldsymbol{x}$, we have $\hat{y} = g(\boldsymbol{x}) = \hat{\beta}_0$. In the following, we denote $(\hat{\beta}_0, \hat{\boldsymbol{\beta}})$ as $\hat{\boldsymbol{\beta}}^*$. We can linearly estimate $\hat{\boldsymbol{\beta}}^*$ as

$$\hat{\boldsymbol{\beta}}^* = \left(\boldsymbol{X}^{*t} \boldsymbol{W} \boldsymbol{X}^*\right)^{-1} \boldsymbol{X}^{*t} \boldsymbol{W} \boldsymbol{y},$$

where $\boldsymbol{W} = \mathrm{diag}(\boldsymbol{w}^2)$. We can interpret $\hat{\boldsymbol{\beta}}$ as an estimate of the gradient $(\partial m(\boldsymbol{x})/\partial X_1, \ldots, \partial m(\boldsymbol{x})/\partial X_p)^t$. For estimates of second derivatives, we would need at least a second-order fit.

Therefore, the cornerstone of (linear) local regression is the estimation of a suitable bandwidth. We focus on the multivariate case, where a direct estimation is not straightforward. This estimation implies unknown functionals which themselves depend on the bandwidth. Typically, the bandwidth is set by direct (plugin) computation (Wand and Jones, 1994; Yang and Tschernig, 1999), selected by cross-validation (Sain et al., 1994; Hall et al., 2007), or found within some type of suboptimal search (Lafferty and Wasserman, 2008). It is known that a suitable plugin estimate of $\boldsymbol{h}$ can improve the cross-validated estimate. In this paper, we work on the basis of a plugin diagonal bandwidth estimate, which, as mentioned above, is a reasonable tradeoff between a scalar bandwidth and a full-matrix bandwidth.

We state that the kernel estimate in the local regression framework should account for the importance of each variable. In other words, if a variable is absolutely irrelevant for the regression function, or noisy, it should not participate in the weights calculation (Vidaurre et al., 2012). Since the best rate of convergence in non-parametric regression is $N^{-4/(4+p)}$ (Györfi et al., 2002), to exploit the sparse nature of $m(\cdot)$ is extremely convenient.

The approach taken by Lafferty and Wasserman (2008), so-called *regularization of derivative expectation operator*, or *rodeo*, also uses a diagonal bandwidth, and is of special interest to us because they consider sparsity in $m(\boldsymbol{x})$. Specifically, they use the estimated gradient of the regression function with respect to the bandwidth, $\partial m(\boldsymbol{x})/\partial \boldsymbol{h}$, to conduct a greedy search, considering that a high value of $\partial m(\boldsymbol{x})/\partial h_j$ is indicative of the relevance of variable X_j. In other words, this gradient tells how the regression function varies with infinitesimal changes of the bandwidth. If it varies little, then the variable is considered to be irrelevant and will be assigned a relatively large bandwidth. Favoring computational speed and applicability in high-dimensional settings, this approach does not generalize for arbitrary nonlinearities. This method assumes a known value of σ^2. If σ^2 is unknown, it has to be separately estimated.

In this paper, we take a adaptive regularized multivariate local regression approach by defining appropriate distributions over the parameters, combining it with an efficient bandwidth estimation method. We call it *sparse bandwidth selector* (*sbase* for short). The method includes the estimation of σ^2 and considers sparsity by analyzing the estimated bandwidths at each step. Several elements of the method are analogous to other approaches (which do not consider sparsity explicitly), as for example the work by Yang and Tschernig (1999). We expect that a suitable application of adaptive ridge regularization will further improve the bias–variance tradeoff of the estimation. We also propose an estimation of the regression coefficients though sampling methods, so that we obtain an estimate of the posterior distribution of the response.

The rest of the paper is organized as follows. Section 2 introduces the Bayesian hierarchical model. Section 3 describes the bandwidth selection procedure. Section 4 details how to obtain a maximum a posteriori (MAP) estimate of the response and the parameters. Section 5 introduces how to obtain a posterior distribution of the regression parameters and the response. Section 6 discusses the complexity of the algorithms. Section 7 provides some empirical examples of the performance of the proposed methods. Finally, we draw some conclusions in Section 8.

2. Hierarchical model

We first define the distribution of the response,

$$y|\boldsymbol{\beta}^*, \qquad \sigma^2 \sim \mathcal{N}\left(\beta_0, \sigma^2\right), \tag{2}$$

which does not imply homoscedasticity because it only concerns the point of interest $\boldsymbol{x}$. We show below, nevertheless, how the homoscedasticity assumption makes the algorithm more applicable in high dimensions.

Next, we define a non-isotropic Gaussian prior distribution over $\boldsymbol{\beta}$:

$$\boldsymbol{\beta}|\boldsymbol{\alpha}^2 \sim \mathcal{N}\left(0, \boldsymbol{\Sigma}\right), \tag{3}$$

with parameter $\boldsymbol{\Sigma} = \text{diag}(\boldsymbol{\alpha}^2)$ and $\alpha_j^2 > 0, j = 1, \ldots, p$. We choose a Gaussian prior as it is the conjugate of the Gaussian density, so that the problem is analytically tractable. From the frequentist perspective, this is equivalent to imposing an adaptive L_2-penalty on the regression coefficients, where the regularization parameters play the role of the diagonal matrix $\sigma^2\boldsymbol{\Sigma}^{-1}$. Then, $\boldsymbol{\alpha}^2$ adaptively tunes the regularization for each parameter β_j.

Considering non-informative improper prior densities over σ^2 and α_j^2, given respectively by $1/\sigma^2$ and $1/\alpha_j^2$, along with a non-informative prior over β_0, we complete the hierarchical representation of the model.

3. Bandwidth selection

In this section, we assume that σ^2 and $\boldsymbol{\alpha}^2$ are known quantities. We also assume the set Ω to contain the relevant variables for the estimation. We denote its cardinality as $|\Omega|$. The optimal estimate of $\boldsymbol{h}$ is given by the minimization of the weighted mean integrated squared error (MISE) statistic

$$\text{MISE}(\boldsymbol{h}) = E\left[\int (m(\mathbf{x}) - g(\mathbf{x}))^2 f_X(\mathbf{x}) z(\mathbf{x}) d\mathbf{x}\right],$$

where $f_X(\cdot)$ is the design density function and $z(\cdot)$ is some weighting function, provided to allow the design density $f_X(\cdot)$ to be not compactly supported. In practice, $z(\cdot)$ can be, for example, an indicator of the support of $m(\cdot)$ or an indicator of some neighborhood of the point of interest $\boldsymbol{x}$.

Let us define, for $j, j' \in \Omega$, the Hessian matrix $\boldsymbol{\Lambda}(\mathbf{x})$ with elements

$$\lambda_{jj'}(\mathbf{x}) = \frac{\partial^2 m(\mathbf{x})}{\partial X_j \, \partial X_{j'}}. \tag{4}$$

An asymptotic approximation of the MISE-optimal bandwidth $\hat{\boldsymbol{h}}$ is defined for example in Yang and Tschernig (1999). Particularizing for the Gaussian kernel, we have

$$\hat{\boldsymbol{h}} = \left(\frac{\|K\|_2^{2p}\psi(s)}{N\varphi(K)}\right)^{(1/p+4)} \exp\left(\frac{\phi(\boldsymbol{C}(m))}{2}\right),$$

where $\|K\|_2 = \int K^2(t)dt$ and $\varphi(K) = \int t^2 K(t)dt$. If the support is infinite, then $\|K\|_2 = 1/\sqrt{\pi}$ and $\varphi(K) = 1$.

Under the homoscedasticity assumption, the functional $\psi(s)$ is defined as

$$\psi(s) = \int s^2(\mathbf{x}) z(\mathbf{x}) d\mathbf{x} = \sigma^2 \int z(\mathbf{x}) d\mathbf{x}. \tag{5}$$

The elements of the non-negative definite matrix $\boldsymbol{C}(m) \in \mathbb{R}^{|\Omega|\times|\Omega|}$ are defined as

$$c_{jj'}(m) = \int \lambda_{jj}(\mathbf{x})\lambda_{j'j'}(\mathbf{x}) f_X(\mathbf{x}) z(\mathbf{x}) d\mathbf{x} \simeq \frac{1}{N}\sum_{i=1}^{N} \lambda_{jj}(\boldsymbol{x}_i)\lambda_{j'j'}(\boldsymbol{x}_i),$$

where the approximation assumes that $z(\cdot)$ is an indicator of the support of $m(\cdot)$. Note that, since our bandwidth estimate is diagonal, we only need the diagonal elements of $\boldsymbol{\Lambda}(\mathbf{x})$. We define $\phi(\boldsymbol{M})$ as the solution of the unconstrained optimization problem

$$\phi(\boldsymbol{M}) = \underset{\boldsymbol{v}}{\text{argmin}} \ \frac{1}{4}\exp(\boldsymbol{v})^t \boldsymbol{M} \exp(\boldsymbol{v}) + \exp\left(\sum_{j=1}^{p} \frac{-v_j}{2}\right),$$

which can be proved to be convex because both terms are positive definite, and, thus, the Hessian with respect to $\boldsymbol{v}$ is positive definite. Then, this problem can be easily solved for example by a standard application of the Newton–Raphson algorithm.

Finally, if required, a kernel density estimate of $f_X(\boldsymbol{x}_i)$ can be obtained as

$$f_X(\boldsymbol{x}_i) = \frac{1}{N}\sum_{i'=1}^{N} K_{\boldsymbol{h}}(\boldsymbol{x}_{i'} - \boldsymbol{x}_i).$$

Yang and Tschernig (1999) proposed finding an approximation of $\boldsymbol{\Lambda}(\boldsymbol{x}_i)$ by performing, for each variable, a local cubic estimation with several cross-terms left out. This approximation itself requires a scalar pilot bandwidth which also depends on unknown functionals that need to be estimated. Although these approximations are low biased, if N is not high enough, they can have a big variance even for moderate values of p. In this paper, we use a simpler approximation of the diagonal Hessian $\boldsymbol{\Lambda}(\mathbf{x})$ based on numerical differentiation that may be more adequate in a high-dimensional setting,

$$\hat{\lambda}_{jj}(\boldsymbol{x}_i) = \frac{\hat{\beta}_j(\boldsymbol{x}_i + \boldsymbol{\epsilon}_j) - \hat{\beta}_j(\boldsymbol{x}_i - \boldsymbol{\epsilon}_j)}{2\epsilon}, \tag{6}$$

where $\epsilon > 0$ is some small constant and $\boldsymbol{\epsilon}_j$ is a p-length vector whose jth element is ϵ and the rest are zero. The estimation $\hat{\beta}_j(\boldsymbol{x}_i + \boldsymbol{\epsilon}_j)$ is obtained as the $(j+1)$th element of the vector

$$\left(\sigma^2\boldsymbol{\Sigma}^{-1} + \boldsymbol{X}^{*t}\boldsymbol{W}\boldsymbol{X}^*\right)^{-1}\boldsymbol{X}^{*t}\boldsymbol{W}\boldsymbol{y},$$

where the diagonal matrix $\boldsymbol{W}$ has elements $K_{\boldsymbol{h}}(\boldsymbol{x}_{i'} - \boldsymbol{x}_i - \boldsymbol{\epsilon}_j)$, $i' = 1, \ldots, N$. The estimation of $\hat{\beta}_j(\boldsymbol{x}_i - \boldsymbol{\epsilon}_j)$ is analogous. Hence, we need $2p$ simple first-order Taylor expansions for estimating $\boldsymbol{\Lambda}(\boldsymbol{x}_i)$. Although this approach is based on well-known numerical approximation techniques, it is novel for bandwidth estimation to the extent of our knowledge. Also, for a known σ^2, $\psi(s)$ is directly estimated by Eq. (5).

Note that this Hessian estimate is suboptimal, because we apply the current estimated bandwidth for $\boldsymbol{x}$ to all data points involved in Eq. (4), and the curvature of $m(\cdot)$ at these points can differ from that at $\boldsymbol{x}$. However, since we seek a local estimate of the bandwidth, we can take $z(\boldsymbol{x}_i) = 1$ if $\boldsymbol{x}_i$ is within some neighborhood of $\boldsymbol{x}$ and $z(\boldsymbol{x}_i) = 0$ otherwise. Then, we only need to estimate the Hessian in this neighborhood and, assuming that $m(\cdot)$ is not very wiggly, the current estimate $\hat{\boldsymbol{h}}$ is reasonable for this purpose.

Sparsity is considered here in the sense of Lafferty and Wasserman (2008): to discard a variable X_j amounts to using a sufficiently high bandwidth h_j. Then, its contribution for the calculation of $\boldsymbol{w}$ becomes negligible. Note that, for infinite data, the optimal bandwidth for an irrelevant covariate is ∞. The optimal bandwidth of a relevant covariate, on the other hand, will be finite (it will typically be small or moderate).

In this paper, we use the converse argument for detecting sparsity: if the contribution of X_j to the computation of $\boldsymbol{w}$ is insignificant, then we can safely drop X_j. Whereas (Lafferty and Wasserman, 2008) consider the changes in the estimate $\hat{y} = g(\boldsymbol{x})$, we take a more direct approach by considering the changes on $\boldsymbol{w}$, on which $\hat{y}$ ultimately depends.

Hence, for X_j to be dropped, we can use the criterion

$$\frac{K(0) - K(\eta_j/h_j)}{K(0)} < \varepsilon, \tag{7}$$

where $\eta_j = \max_{i=1}^{N} |x_j - x_{ij}|$ and $\varepsilon > 0$ is some small constant. The left-hand side of Eq. (7) is thus the percentage of decay of $K(\cdot)$ along this direction. If Eq. (7) holds for certain small $\epsilon > 0$ and direction j, then $K(\cdot)$ is almost constant in this direction, and X_j can be dropped. From Eq. (7), we can derive the following threshold:

$$\tau_j = \sqrt{-\frac{\eta_j^2}{2\log(1-\varepsilon)}},$$

so that Ω is updated to contain only those variables that hold $\hat{h}_j < \tau_j$. Variables not included in Ω will have their bandwidths clamped to τ_j and can be removed from the above-described computations.

Fig. 1, left, illustrates the value $K(\eta_j/h_j)/h_j$ for several $h_j = 0.1, \ldots, 4.0$ and $|x_j - x_{ij}|$ within the unit support; each line represents a different h_j. The value $K(\eta_j/h_j)$ has been normalized so that $\int_0^1 K(t)dt = 1$. The thick red line corresponds to $h_j = 4.0$. Fig. 1, right, shows the left-hand side of Eq. (7) for $h_j = 0.1, \ldots, 4.0$. Constant $\varepsilon = 0.1$ is represented by the dashed horizontal line. For $\varepsilon = 0.1$, we have $\tau_j = 1.089$, represented by the dotted vertical line.

As we shall show below, the proposed algorithm typically increases the bandwidth of the irrelevant variables at each iteration. In an ideal situation, this will occur unlimitedly, so that $h_j \to \infty$ and we can choose $\epsilon \to 0$. However, because real data are finite, the bandwidth of irrelevant variables will not probably increase further than to a certain limit. Then, a reasonably small value, such as $\epsilon = 0.1$, is adequate. Note that smaller values of ϵ imply more computational cost. In summary, this is a natural variable selection rule based on the definition of the bandwidth.

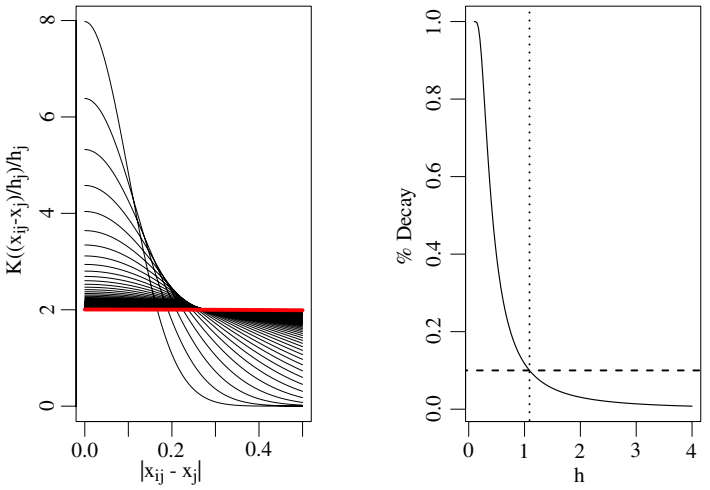

Fig. 1. Left, values $K(\eta_j/h_j)/h_j$ for $h_j = 0.1, \ldots, 4.0$ and $\eta_j = 1.0$. Right, left-hand side of Eq. (7) for $h_j = 0.1, \ldots, 4.0$; $\varepsilon = 0.1$ is represented by the dashed horizontal line and $\tau_j = 1.089$ is represented by the dotted vertical line.

4. MAP estimation

In this section, we derive an expectation–maximization algorithm to obtain the parameters $\boldsymbol{\beta}^*, \sigma^2$ and $\boldsymbol{\alpha}^2$. Let us assume that by now we have an estimate of the optimal diagonal bandwidth $\hat{\boldsymbol{h}}$, and, hence, we have weight values $w_i, i = 1, \ldots, N$. Then, we can obtain the parameters $\boldsymbol{\beta}^*, \sigma^2$ and $\boldsymbol{\alpha}^2$ that maximize the expected MAP function by means of the expectation–maximization (EM) algorithm (Dempster et al., 1977). We consider $\boldsymbol{\beta}^*$ to be the latent variable.

From Eqs. (2) and (3), the expectation of the complete-data log likelihood function is

$$\begin{aligned}\mathrm{E}[\log p(\boldsymbol{\beta}^*, \boldsymbol{y}|\boldsymbol{\alpha}^2, \sigma^2)] &= \mathrm{E}[\log p(\boldsymbol{y}|\boldsymbol{\beta}^*, \sigma^2)] + \mathrm{E}[\log p(\boldsymbol{\beta}|\boldsymbol{\alpha}^2)] \\ &= \underbrace{-\frac{N_{\boldsymbol{w}}}{2}\log(2\pi\sigma^2) - \frac{\mathrm{E}[(y-\hat{y})^2]}{2\sigma^2}}_{\mathrm{E}[\log p(\boldsymbol{y}|\boldsymbol{\beta}^*, \sigma^2)]} \underbrace{-\frac{p}{2}\log(2\pi) - \frac{1}{2}\log\det(\boldsymbol{\Sigma}) - \frac{\mathrm{E}[\boldsymbol{\beta}^t\boldsymbol{\Sigma}^{-1}\boldsymbol{\beta}]}{2}}_{\mathrm{E}[\log p(\boldsymbol{\beta}|\boldsymbol{\alpha}^2)]}, , \end{aligned} \tag{8}$$

where $N_{\boldsymbol{w}} = \sum_{i=1}^{N} w_i^2$, and we approximate (Loader, 1999)

$$\mathrm{E}[(y-\hat{y})^2] \simeq \sum_{i=1}^{N}\left(w_i y_i - w_i\beta_0 - w_i\boldsymbol{x}_i^t\boldsymbol{\beta}\right)^2.$$

In the E-step, we obtain the values of $\boldsymbol{\beta}^*$ that maximize the expectation in Eq. (8). Setting the derivatives of $\mathrm{E}[\log p(\boldsymbol{\beta}^*, \boldsymbol{y}|\boldsymbol{\alpha}^2, \sigma^2)]$ (in Eq. (8)) with respect to $\boldsymbol{\beta}^*$ to zero, for some estimates $\hat{\sigma}^2$ and $\hat{\boldsymbol{\alpha}}^2$, we obtain the parameter estimate at this step as

$$\hat{\boldsymbol{\beta}}^* = \hat{\sigma}^{-2}\hat{\boldsymbol{S}}\boldsymbol{X}^{*t}\boldsymbol{W}\boldsymbol{y}, \tag{9}$$

where $\boldsymbol{W} = \mathrm{diag}(\boldsymbol{w}^2)$ and $\hat{\boldsymbol{S}}$ is the estimated covariance matrix of the posterior distribution of $\boldsymbol{\beta}$, computed as

$$\hat{\boldsymbol{S}} = \left(\mathrm{diag}^{-1}(0, \hat{\boldsymbol{\alpha}}^2) + \frac{1}{\hat{\sigma}^2}\boldsymbol{X}^{*t}\boldsymbol{W}\boldsymbol{X}^*\right)^{-1}. \tag{10}$$

In the M-step, we estimate the values of σ^2 and $\boldsymbol{\alpha}^2$ that maximize Eq. (8) for the current estimation of $\boldsymbol{\beta}^*$. We set the derivatives of Eq. (8) with respect to σ^2 to zero to obtain

$$\hat{\sigma}^2 = \frac{\mathrm{E}[(y-\hat{y})^2]}{N_{\boldsymbol{w}}} \simeq \frac{\sum_{i=1}^{N}(w_i y_i - w_i\hat{\beta}_0 - w_i\boldsymbol{x}_i{}^t\hat{\boldsymbol{\beta}})^2}{N_{\boldsymbol{w}}}. \tag{11}$$

To estimate $\boldsymbol{\alpha}^2$, we first have

$$\frac{\partial\ \log\det(\boldsymbol{\Sigma})}{\partial\alpha_j^2} = \frac{\partial\ \log\left(\prod_{j'=1}^{p}\alpha_{j'}^2\right)}{\partial\alpha_j^2} = \frac{1}{\alpha_j^2}. \tag{12}$$

Algorithm 1 Iterative algorithm for regularized local MAP estimation

Initialize $\boldsymbol{h} = h_0$, $\sigma^2 = \sigma_0^2$, $\boldsymbol{\alpha}^2 = \infty$.
Repeat until convergence:
  Compute $\hat{\boldsymbol{h}}$ as described in Section 3.
  Compute $\boldsymbol{w}$ with Eq. (1).
  Repeat until convergence:
    Update $\hat{\boldsymbol{\beta}}^*$ with Eq. (9).
    Update σ^2 and $\boldsymbol{\alpha}^2$ with, respectively, Eqs. (11) and (14).

On the other hand, using the definition of variance, we have

$$\frac{\partial \, \mathrm{E}[\boldsymbol{\beta}^t \boldsymbol{\Sigma}^{-1} \boldsymbol{\beta}]}{\partial \alpha_j^2} = \frac{\partial \, \mathrm{E}[\Sigma_{j'=1}^{p} \beta_{j'}^2 / \alpha_{j'}^2]}{\partial \alpha_j^2} = \frac{\partial \sum_{j'=1}^{p} \mathrm{E}[\beta_{j'}^2]/\alpha_{j'}^2}{\partial \alpha_j^2} = -\frac{\hat{\beta}_j^2 + \hat{\mathrm{s}}_{j+1,j+1}}{\alpha_j^4}, \tag{13}$$

where $\hat{\mathrm{s}}_{j+1,j+1}$ is the $(j+1)$th element of the diagonal of $\hat{\boldsymbol{S}}$.

Putting results (12) and (13) together and reorganizing terms, we can estimate α_j^2 as

$$\hat{\alpha}_j^2 = \hat{\beta}_j^2 + \hat{\mathrm{s}}_{j+1 j+1}. \tag{14}$$

By means of the automatic relevance determination principle, values $\hat{\alpha}_j^2$ that correspond to irrelevant variables for the estimation of $\boldsymbol{\beta}^*$ will be close to zero at convergence.

Note that the estimation of $\boldsymbol{\beta}$ depends on σ^2, whose estimation depends itself on $\boldsymbol{\beta}$. The same happens with $\boldsymbol{\alpha}^2$, whose estimation depends recursively on itself through $\boldsymbol{S}$. Iterating the estimation of $\hat{\boldsymbol{\beta}}$ with Eq. (9) and the estimation of σ^2 and $\boldsymbol{\alpha}^2$ with Eqs. (11) and (14), and repeating until convergence, the EM algorithm is able to find a MAP solution for a fixed $\boldsymbol{w}$ in a finite number of steps. We summarize the MAP estimation in Algorithm 1.

We can make use of sparsity as defined in Section 2 by restricting the above computations to those variables that are currently included in Ω. This can save computation time and, since it increases the ratio N/p, can improve the estimation. Note that we are handling sparsity at two levels. First, the sparsity of function $m(\cdot)$ is determined by the magnitude of the bandwidths. Second, the sparsity of $\hat{\boldsymbol{\beta}}$ is handled by automatic relevance determination. For example, $\beta_j = 0$ does not necessarily means that X_j is dispensable. Hence, we cannot use sparsity in $\hat{\boldsymbol{\beta}}$ to simplify the estimation of $\boldsymbol{h}$.

5. Estimation by Monte Carlo sampling

In this section, we evolve from the MAP approach to a sampling method for determining the posterior distribution of $\boldsymbol{\beta}^*$, σ^2 and $\boldsymbol{\alpha}^2$, and then the predictive distribution.

5.1. Parameter distribution

First, we estimate the posterior parameter distribution of $\boldsymbol{\beta}^*$, defined by the sufficient statistics $\mathrm{E}[\boldsymbol{\beta}^*] = \hat{\boldsymbol{\beta}}^*$ and $\mathrm{Var}[\boldsymbol{\beta}^*] = \hat{\boldsymbol{S}}$. Instead of sampling from the joint posterior of $\boldsymbol{h}$, $\boldsymbol{\beta}^*$, σ^2 and $\boldsymbol{\alpha}^2$, we sample from the complete-data parameter posterior of σ^2 and $\boldsymbol{\alpha}^2$, given the current estimates of $\boldsymbol{\beta}^*$ and $\boldsymbol{h}$. We alternate between two steps.

In the first step, given the current posterior density estimate, the objective is to obtain L samples $\Theta_l \equiv \{\sigma^{2(l)}, \boldsymbol{\alpha}^{2(l)}\}$.

First, from the current estimate of $\boldsymbol{h}$, we compute $\boldsymbol{w}$ by Eq. (1) and $N_{\boldsymbol{w}} = \sum_{i=1}^{N} w_i^2$. We shall now be able to sample from $p(\sigma^2, \boldsymbol{\alpha}^2 | \boldsymbol{y})$ by following the hierarchy defined above. For each l, we can sample $1/\sigma^{2(l)}$ from a Gamma distribution with parameters

$$a_{\sigma^2} = \frac{N_{\boldsymbol{w}}}{2} + 1 \quad \text{and} \quad b_{\sigma^2} = \frac{1}{2} \sum_{i=1}^{N} (w_i y_i - w_i \hat{\beta}_0 - w_i \boldsymbol{x}_i^t \hat{\boldsymbol{\beta}})^2, \tag{15}$$

where $\hat{\boldsymbol{\beta}}^*$ is the current MAP estimate of $\boldsymbol{\beta}^*$ obtained from Eq. (9). Since the mode of the Gamma distribution is given by $(a_{\sigma^2} - 1)/b_{\sigma^2}$, this is consistent with the result in Eq. (11). Also, we sample $1/\alpha_j^{2(l)}$, for $j = 1, \ldots, p$, from a Gamma distribution with parameters

$$a_{\alpha_j^2} = \frac{3}{2} \quad \text{and} \quad b_{\alpha_j^2} = \frac{1}{2}(\hat{\beta}_j^2 + \hat{\mathrm{s}}_{j+1 j+1}), \tag{16}$$

where $\hat{\mathrm{s}}_{j+1,j+1}$ is the $(j+1)$th element of the diagonal of $\hat{\boldsymbol{S}}$.

Algorithm 2 Sampling algorithm for finding the posterior distribution of $\boldsymbol{\beta}^*$, σ^2 and $\boldsymbol{\alpha}^2$

Initialize the posterior distribution of $\boldsymbol{\beta}^*$ and $\hat{\boldsymbol{h}}$. Obtain $\boldsymbol{w}$.

Step (i):
Draw L samples Θ_l given that:
σ^2 is Gamma distributed with parameters given by Eq. (15).
$\boldsymbol{\alpha}^2$ are Gamma distributed with parameters given by Eq. (16).
For each Θ_l, estimate $\hat{\boldsymbol{h}}^{(l)}$ as described in Section 3.
For each $\hat{\boldsymbol{h}}^{(l)}$, compute $\boldsymbol{w}^{(l)}$ with Eq. (1).

Step (ii):
For each l, using $\boldsymbol{w}^{(l)}$, iterate until convergence:
Compute $\hat{\boldsymbol{\beta}}^{(l)}$ and $\hat{\boldsymbol{S}}^{(l)}$ with Eqs. (9) and (10).
Update $\hat{\sigma}^{2^{(l)}}$ and $\hat{\boldsymbol{\alpha}}^{2^{(l)}}$ with Eqs. (11) and (14).
Update the posterior distribution of $\hat{\boldsymbol{\beta}}^{*^{(l)}}$ and $\hat{\boldsymbol{S}}^{(l)}$ with Eq. (17).
Compute $\hat{\boldsymbol{h}} = \sum_{l=1}^{L} \hat{\boldsymbol{h}}^{(l)}$.
Update the posterior distribution of σ^2 and $\boldsymbol{\alpha}^2$ with (15) and (16).

Repeat steps (i) and (ii) until the posterior distribution of $\boldsymbol{\beta}$ stabilizes.

Eqs. (15) and (16) were obtained by "completing the square" on the product of the logarithm of the non-informative prior density (i.e., a Gamma distribution with $a = b = 0$) and the log likelihood given by Eq. (8).

Finally, for each sample Θ_l, we estimate the optimal bandwidth, $\hat{\boldsymbol{h}}^{(l)}$, as described in Section 3, and compute $\boldsymbol{w}^{(l)}$ by Eq. (1).

In the second step, we update the posterior distribution of the parameters using the $\boldsymbol{w}^{(l)}$ samples. For each vector $\boldsymbol{w}^{(l)}$, we iteratively estimate $\hat{\boldsymbol{\beta}}^{*^{(l)}}$ and $\hat{\boldsymbol{S}}^{(l)}$ with Eqs. (9) and (10), and update $\hat{\sigma}^{2^{(l)}}$ and $\hat{\boldsymbol{\alpha}}^{2^{(l)}}$ with Eqs. (11) and (14) until convergence.

Once we have the posterior estimate of the parameters for each sample Θ_l, we perform the following update:

$$\hat{\boldsymbol{\beta}}^* = \frac{1}{L}\sum_{l=1}^{L} \hat{\boldsymbol{\beta}}^{*^{(l)}} \quad \text{and} \quad \hat{\boldsymbol{S}}^{-1} = \frac{1}{L}\sum_{l=1}^{L} \hat{\boldsymbol{S}}^{(l)^{-1}}. \tag{17}$$

Since L is finite, this involves an approximation, which is biased (because so is the estimation of $\boldsymbol{\beta}^*$), and whose variance is proportional to $1/L$.

Note that it is not straightforward to formulate an estimate of the posterior distribution of σ^2 and $\boldsymbol{\alpha}^2$ as a function of the samples $\sigma^{2^{(l)}}$ and $\boldsymbol{\alpha}^{2^{(l)}}$. However, we can still approximate these distributions via $\hat{\boldsymbol{\beta}}^*$ and $\hat{\boldsymbol{S}}^{-1}$, using Eqs. (15) and (16). We also need $\hat{\boldsymbol{h}}$, which can be estimated as $\hat{\boldsymbol{h}} = \sum_{l=1}^{L} \hat{\boldsymbol{h}}^{(l)}/L$.

We summarize the method in Algorithm 2.

5.2. Predictive distribution

Next, we formulate the predictive distribution

$$p(y|\boldsymbol{y}) = \int p(y|\boldsymbol{\beta}^*, \sigma^2, \boldsymbol{y})\, d\boldsymbol{\beta}^* d\sigma^2,$$

which can be shown to be Student t,

$$y|\boldsymbol{y} \sim \mathrm{St}\big(\mu = \hat{\beta}^0, \iota = \hat{\sigma}^{-2} + \hat{s}_{00}, \nu = N - p\big),$$

where μ is the mean, ι is the precision, and ν is the number of degrees of freedom of the distribution.

6. Computational complexity

The dominant step of the two proposed algorithms with regard to the computational cost is the calculation of the diagonal of the Hessian with Eq. (6).

For the point estimation in Section 4, an iteration takes $2N_z|\Omega|$ matrix inversions, of dimension $(|\Omega| - 1) \times (|\Omega| - 1)$, where N_z is the number of data points such that $z(\boldsymbol{x}_i) \neq 0$. The cost of each matrix inversion is $(|\Omega| - 1)^{2.376}$ with the Coppersmith–Winograd algorithm (Coppersmith and Winograd, 1990). Then, the cost per iteration is $2N_z|\Omega|(|\Omega| - 1)^{2.376}$. The overall cost of the algorithm thus depends on the sparsity degree of $m(\cdot)$ and the choice of $z(\cdot)$ and ε. For the sampling method, the cost per iteration is $2LN_z|\Omega|(|\Omega| - 1)^{2.376}$.

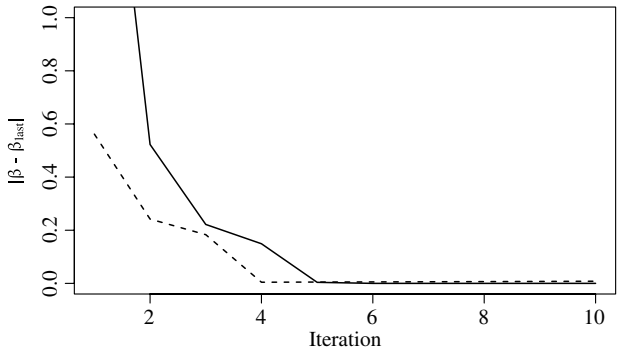

Fig. 2. Absolute change of $\hat{\boldsymbol{\beta}}^*$ per iteration with respect to the last iteration. The solid line represents Algorithm 1 and the dashed line represents Algorithm 2.

Table 1
Regression function for each model.

Model 1	$m(\mathbf{x}) = \mathrm{x}_1^2 + \mathrm{x}_2^2$
Model 2	$m(\mathbf{x}) = \sin(\pi \mathrm{x}_1 + \pi \mathrm{x}_2)$
Model 3	$m(\mathbf{x}) = \sin(\pi \mathrm{x}_1)\sin(2\pi \mathrm{x}_2)$
Model 4	$m(\mathbf{x}) = 0.5\sin(\pi \mathrm{x}_1) + 0.5\sin(4\pi \mathrm{x}_2)$
Model 5	$m(\mathbf{x}) = \left((\mathrm{x}_1 - 0.5)^2 + \mathrm{x}_2^2\right)\sin(2\pi \mathrm{x}_3)$
Model 6	$m(\mathbf{x}) = \sin(\pi \mathrm{x}_1 + 0.5\pi \mathrm{x}_2 + 2\mathrm{x}_3 + 0.5\pi \mathrm{x}_4)$
Model 7	$m(\mathbf{x}) = \mathrm{x}_1^2 \mathrm{x}_2^2$
Model 8	$m(\mathbf{x}) = 2(\mathrm{x}_1 + 1)^3 + 2\sin(10\mathrm{x}_2)$
Model 9	$m(\mathbf{x}) = -K\left((\mathrm{x}_1 - 0.3)/0.3\right)\sin(\pi \mathrm{x}_2) + K\left((\mathrm{x}_1 - 0.7)/0.3\right)\sin(\pi \mathrm{x}_3)$

To give an example, the computation time for some data set with $N = 500$ samples, $p = 10$ covariates, and 2 relevant variables, taking $z(\boldsymbol{x}_i) \neq 0$ for 50 data points, $\varepsilon = 0.1$, and $L = 5$, is 21.73 s for the point estimation and 97.39 s for the sampling method. *Rodeo* takes 12.322 s. Times correspond to an Intel Core 2 Duo processor (2.26 GHz).

The memory requirements of Algorithm 1 are slightly higher than for *rodeo*, because we need to store the estimated Hessian at each point such that $z(\boldsymbol{x}_i) \neq 0$. Since we need to keep L estimates of the parameters, the amount of memory is higher for the sampling algorithm.

Fig. 2 compares Algorithms 1 and 2 in terms of convergence, showing the absolute change of the parameters $\hat{\boldsymbol{\beta}}^*$ with respect to the last iteration for some synthetic toy data set with $N = 500$ samples and $p = 10$ covariates.

Note that the current estimate is only valid for $\boldsymbol{x}$. If an estimation is required over a wide range of input values, then the proposed approach, like most local methods, can be computationally demanding. In this case, we can still run the method for a selected set of points in the data set and extrapolate the incoming data points to the closest data items in this set, whose bandwidths have already been estimated. If such data items are close enough, it is reasonable to use their bandwidth estimates for the new data points.

7. Experiments

In order to show the performance of the proposed approach, we have run Algorithm 1 on both synthetic and real data sets.

First, we consider 100 data sets generated from nine different models. In all cases, there are $N = 500$ samples and $p = 10$ covariates, sampled from the uniform distribution on $[0, 1]$. We have set $\sigma^2 = 0.1$. The test point is randomly sampled, for each data set, within the interval $[0.3, 0.7]^p$. Table 1 shows the regression function $m(\cdot)$ for each model.

The six former regression functions were taken from the experimental section of Yang and Tschernig (1999), the seventh and eighth functions were taken from the work by Lafferty and Wasserman (2008), and the last function was introduced for evaluating the method's behavior when the sparsity pattern varies over the input domain.

We have also tested *rodeo*, the asymptotically optimal bandwidth from Yang and Tschernig (1999) (*ob* for short, where we use oracle values for $\Lambda(\boldsymbol{x}_i)$ and σ^2), and some non-local non-parametric regression methods: the additive model (*am*) (Hastie and Tibshirani, 1990), random forests for regression (*rf*) (Breiman, 2001), the component selection and smoothing operator (*cosso*) (Lin and Zhang, 2006), and projection pursuit regression (*ppr*) (Friedman and Stuetzle, 1981). As a benchmark, we present results from ordinary least squares (*ols*). For our method, we have used $\varepsilon = 0.1$ and $h_0 = 1/\log(\log N)$.

Table 2 reports the mean absolute error (and standard deviation) along the 100 data sets for *sbase* (MAP estimation) and the other methods. Statistical significance is checked by means of the t-test. The performance of the proposed approach is better than that of *ob* in six out of nine models, and always better than that of *rodeo*. Compared to non-local methods, *sbase* always outperforms *am* and *ols*, and is mostly better than *rf*. The accuracy of *sbase* is somewhat comparable to that of *cosso* and *ppr*. For Model 9, however, where the sparsity pattern is locally defined, *sbase* clearly beats the non-local approaches.

Table 2
Mean (and standard deviation) of the absolute error for each model and method. The best result for each row is highlighted in bold.

	sbase	*rodeo*	*ob*	*am*
Model 1	0.021 (±0.02)	0.097 (±0.08)	0.028 (±0.02)	0.174 (±0.06)
Model 2	0.042 (±0.03)	0.170 (±0.16)	0.045 (±0.04)	0.252 (±0.13)
Model 3	**0.068** (±0.04)*	0.285 (±0.20)	0.087 (±0.05)	0.389 (±0.19)
Model 4	0.134 (±0.11)	0.292 (±0.20)	0.117 (±0.07)	0.402 (±0.17)
Model 5	0.094 (±0.08)	0.126 (±0.15)	0.054 (±0.05)	0.111 (±0.13)
Model 6	0.151 (±0.10)	0.437 (±0.33)	0.206 (±0.18)	0.510 (±0.32)
Model 7	**0.018** (±0.02)*	0.059 (±0.05)	0.023 (±0.02)	0.032 (±0.02)
Model 8	0.396 (±0.45)	0.805 (±0.54)	0.300 (±0.20)	1.656 (±1.03)
Model 9	**0.070** (±0.05)*	0.261 (±0.20)	0.087 (±0.06)	0.355 (±0.17)
	rf	*cosso*	*ppr*	*ols*
Model 1	0.032 (±0.02)	**0.009** (±0.01)*	0.041 (±0.05)	0.174 (±0.06)
Model 2	0.110 (±0.07)	0.198 (±0.10)	**0.017** (±0.01)*	0.252 (±0.13)
Model 3	0.130 (±0.07)	0.168 (±0.08)	0.146 (±0.09)	0.389 (±0.19)
Model 4	0.133 (±0.09)	**0.019** (±0.02)*	0.070 (±0.07)	0.402 (±0.17)
Model 5	**0.052** (±0.04)	0.112 (±0.08)	0.103 (±0.08)	0.111 (±0.13)
Model 6	0.260 (±0.18)	0.372 (±0.29)	**0.041** (±0.10)*	0.510 (±0.32)
Model 7	0.023 (±0.02)	0.023 (±0.02)	0.030 (±0.02)	0.032 (±0.02)
Model 8	0.624 (±0.41)	**0.026** (±0.02)*	1.076 (±0.89)	1.656 (±1.03)
Model 9	0.149 (±0.11)	0.233 (±0.1)	0.324 (±0.25)	0.355 (±0.17)

* The difference to the second best method is statistically significant with a significance level of 0.05.

Table 3
For each model, accuracy ranking of the methods.

	sbase	*rodeo*	*ob*	*am*	*rf*	*cosso*	*ppr*	*ols*
Model 1	2	6	3	7	4	1	5	8
Model 2	2	5	3	7	4	6	1	8
Model 3	1	6	2	8	3	5	4	7
Model 4	5	6	3	7	4	1	2	8
Model 5	3	8	2	5	1	7	4	6
Model 6	2	6	3	7	4	5	1	8
Model 7	1	8	4	7	2	3	5	6
Model 8	3	5	2	7	4	1	6	8
Model 9	1	5	2	7	3	4	6	8
Mean	**2.2**	6.1	2.7	6.9	3.2	3.7	3.8	7.4

Table 4
Percentage of times that each variable has been selected by *sbase* across the 100 experiment replications.

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
Model 1	1.00	1.00	0.35	0.25	0.32	0.27	0.25	0.31	0.21	0.29
Model 2	0.99	0.99	0.30	0.24	0.13	0.32	0.17	0.17	0.21	0.17
Model 3	1.00	1.00	0.29	0.17	0.20	0.15	0.05	0.26	0.20	0.12
Model 4	1.00	0.96	0.14	0.18	0.14	0.16	0.16	0.17	0.15	0.17
Model 5	0.21	0.12	0.97	0.32	0.31	0.34	0.33	0.24	0.26	0.48
Model 6	1.00	0.75	1.00	0.88	0.13	0.08	0.08	0.08	0.05	0.06
Model 7	0.94	0.93	0.33	0.29	0.24	0.37	0.24	0.22	0.27	0.23
Model 8	0.96	0.96	0.34	0.42	0.31	0.31	0.23	0.24	0.32	0.27
Model 9	0.99	1.00	1.00	0.06	0.08	0.03	0.03	0.08	0.06	0.06

Table 3 shows, for each model, an accuracy ranking of the methods. It can be observed that *sbase* is the best ranked method on average (last row).

Table 4 shows the number of times that each variable has been selected by *sbase* across the 100 experiment replications. Note that, for most models, *sbase* basically selects the correct covariates. For Model 5, however, it often discards variables X_1 and X_2. Interestingly, the accuracy of *sbase* (ranked the third) is no much worse here than that of *ob* and *rf*, the best methods for this data set.

Fig. 3 shows, for some run on Model 1, the progression of the bandwidths. Note that the bandwidths of most of the irrelevant variables (dashed lines) are increased at each iteration. Conversely, the bandwidth of the relevant covariates become steady at a low value after a few iterations. For illustration purposes, we have not removed the covariates from the computation once $h_j > \tau_j$.

Fig. 4 shows boxplots of the bandwidth vector for *sbase* and *rodeo*. For *ob* (not shown), there is a much bigger difference between the relevant and the irrelevant covariates. We can observe that, even when *rodeo* tends to assign higher bandwidths

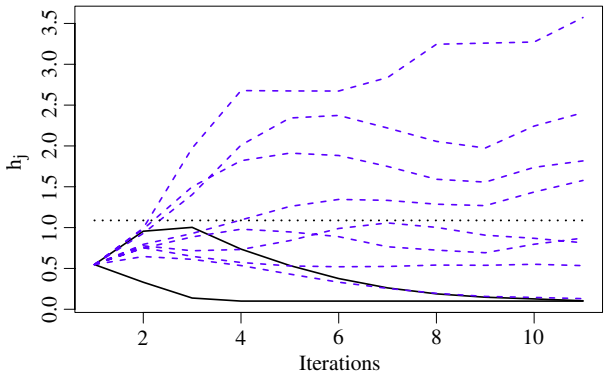

Fig. 3. Progression of the bandwidths for the *sbase* algorithm, for Model 1. The solid lines represent the relevant variables, the dashed lines represent the irrelevant variables, and the horizontal dotted line represents the threshold τ_j.

to the irrelevant covariates, the separation between relevant and irrelevant covariates is much more obvious for *sbase*, which explains its better accuracy.

Fig. 5 (top left graph) shows, for Model 9, the predicted response of *sbase*, *rodeo*, and *rf* (the best non-local method for Model 9) over a grid of values of X_1, the variable that defines the local sparsity pattern. Note that *sbase* exhibits a smoother and more accurate prediction than *rodeo*.

The other three graphs of Fig. 5 illustrate how model selection varies in different parts of the data set. For clarity, we have included only variables $X_1, \ldots, X_5$. Interestingly, the top right graph indicates that *sbase* assigns a higher bandwidth for variable X_2 (or X_3) in those parts of the input domain where it is less present. The bandwidths of the irrelevant variables are not shown here because their orders of magnitude are ten-fold higher, and this tarnishes the appreciation of the differences between X_2 and X_3. Recall that a high bandwidth amounts to discarding this variable. The bottom left graph displays the bandwidths estimated by *rodeo*. Note that it is not clear where X_1 dominates X_2 and vice versa. The bottom right graph shows a measure of importance proposed by Breiman (2001) for each variable across the data set. It can be observed that *rf* fails to locally discriminate variables, giving an almost identical importance to X_2 and X_3 in the entire data range.

In these experiments, the sampling method introduced in Section 5 produces very similar results than the point estimation method of Section 4, and is not shown. To gain more insight into the Monte Carlo sampling method introduced in Section 5, we have run Algorithm 2 on data generated from the models described above, using different values for the variance of e_i: $\sigma^2 \in \{0.1, 0.2, 0.4\}$. Also, to introduce more uncertainty, we add some Gaussian noise to the relevant covariates (after computing the response $\boldsymbol{y}$ with the regression functions in Table 1), and then we redo the experiments. More precisely, we make $x_{ij} := x_{ij} + \varrho_{ij}$, where j corresponds to a relevant variable, $i = 1, \ldots, N$, and $\varrho_{ij} \sim \mathcal{N}(0, 0.2)$.

To give an example, Fig. 6 shows the predictive distribution for some execution and Model 1. As expected, the more uncertainty the model has, the higher the variance of the response distribution is. This applies both for σ^2 and covariate noise.

Next, we consider the *Computer Hardware* data set, taken from the UCI repository (http://archive.ics.uci.edu/ml). The data set has $N = 209$ samples and $p = 6$ covariates. The response is the measured relative CPU performance for a number of machines. We have evaluated the methods at 30 data points, which we selected to be inner to the input domain. Specifically, the chosen points are those with the minimum infinity-norm distance to the average, defined as $\|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|_\infty = \max_j |x_{ij} - \bar{x}_j|$, where $\bar{\boldsymbol{x}} = \sum_i^N \boldsymbol{x}_i / N$. We have again set $\epsilon = 0.1$.

Fig. 7 shows a boxplot of the absolute and squared errors for each method. The proposed approach performs the best among the local methods, although the difference is not big. It also exhibits a lower error than all the non-local methods excepting *rf*, which has the lowest error of all methods. It is worth noting that *sbase* selects only three covariates (minimum main memory, maximum main memory, and cache memory), whereas *rodeo* and *ob* do not discard any covariate. These three variables match the three variables that are given the highest importance coefficient by *rf*, the most accurate method for this data set.

Finally, we consider the *Concrete Compressive Strength* data set (from the UCI repository), whose response function is highly nonlinear. Here, we have $N = 1030$ and $p = 8$. We have evaluated the methods on 30 data points, selected as for the *Computer Hardware* data set. We have used $\epsilon = 0.1$.

Fig. 8 displays a boxplot of the absolute and squared errors for each method for this data set. In this case, the difference between *sbase* and the rest of the methods, both local and non-local, is more obvious. Note that *rf* is again the exception. This method clearly outperforms the other global methods and is comparable to *sbase*. With regard to model selection, *sbase* selects variables X_1, X_5, X_7, and X_8 for most testing points. On the other hand, *rf* gives the highest importance to variables X_1, X_5, X_7, X_8, and X_4. These results are relatively coherent for the two algorithms, excepting for variable X_4.

Like in the synthetic experiments, the results for the sampling method (not shown) are very similar to those for the simpler *sbase* in both real data sets.

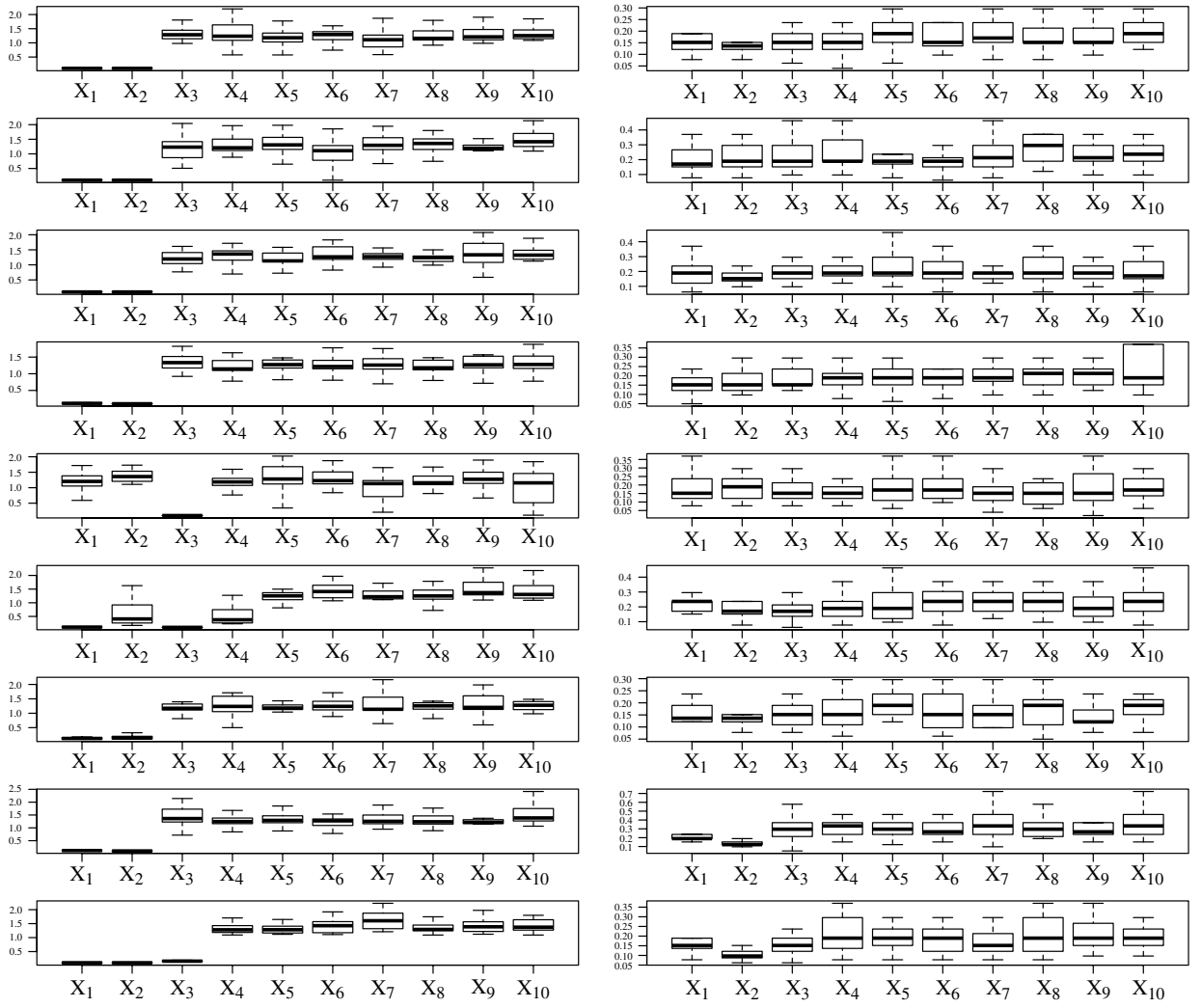

Fig. 4. Boxplots of the bandwidth vector. From top to bottom, Models 1–9 are illustrated. Left graphs correspond to *sbase* and right graphs correspond to *rodeo*.

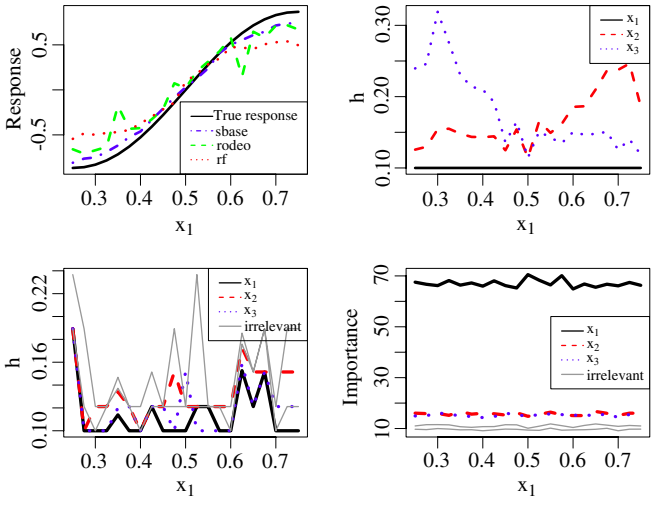

Fig. 5. Top left, predicted response by *sbase*, *rodeo*, and *rf*, for Model 9. Top right, bandwidth estimation for *sbase*. Bottom left, bandwidth estimation for *rodeo*. Bottom right, measures of importance for *rf*.

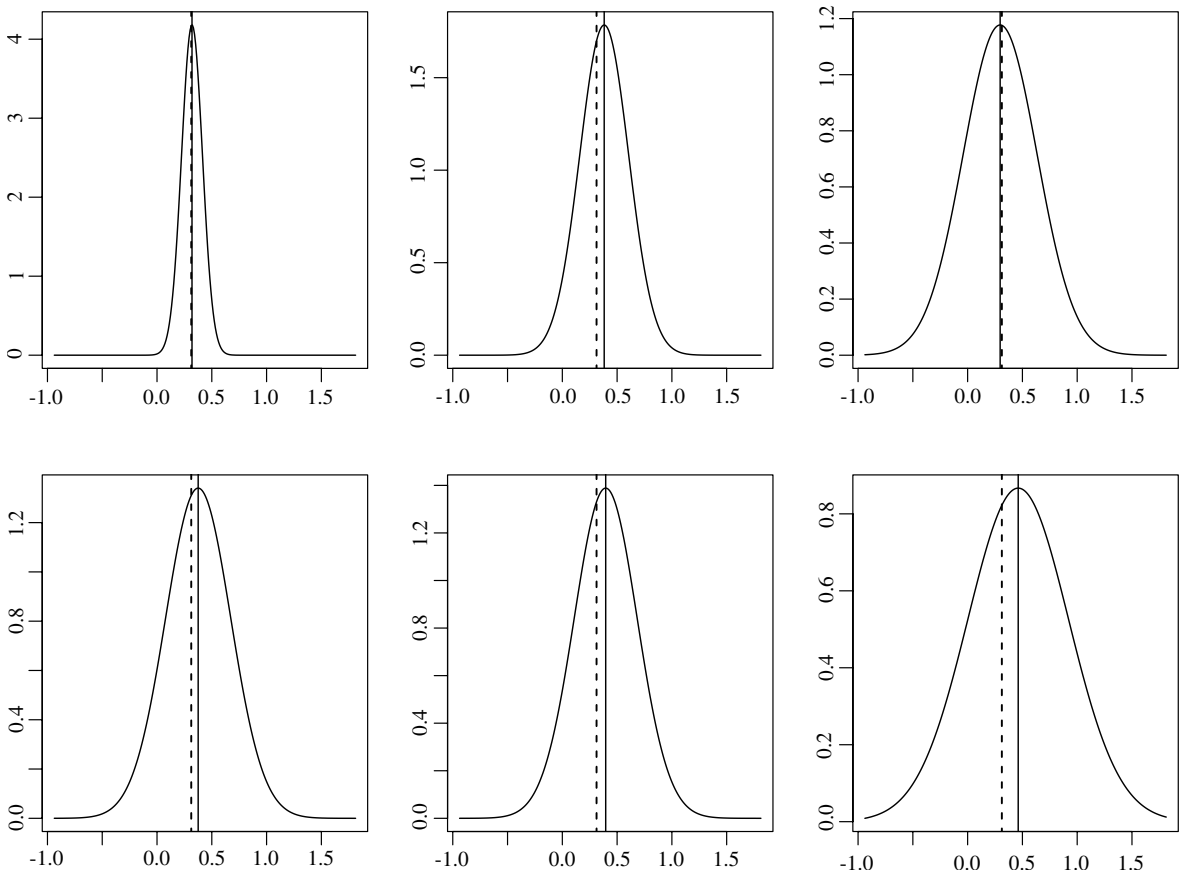

Fig. 6. Estimated distribution of the responses for Model 1, with $\sigma^2 = 0.1$ (left), $\sigma^2 = 0.2$ (middle), and $\sigma^2 = 0.4$ (right). The top graphs were generated from data with non-noisy covariates and the bottom graphs were generated from data with noisy covariates. The vertical solid line indicates the mode of the distribution and the vertical dotted line indicates the true response.

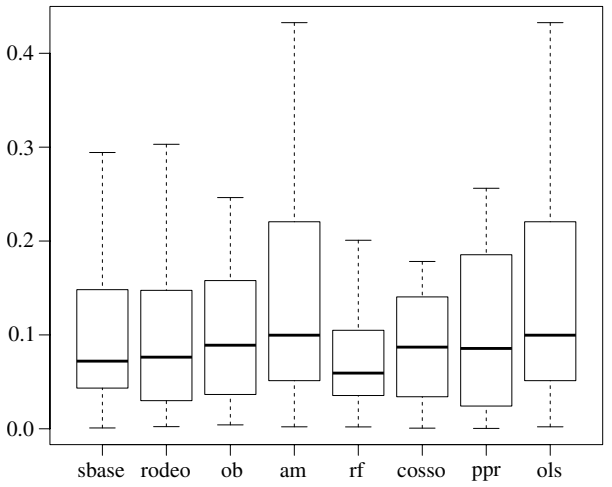

Fig. 7. Boxplots of the absolute and squared errors for each method and the *Computer Hardware* data set.

8. Concluding remarks

This paper introduces a sparse regularized local regression method, showing how to give both a point estimation and the posterior distribution estimation of the regression coefficients $\boldsymbol{\beta}^*$, the noise variance σ^2, and parameter variance $\boldsymbol{\alpha}^2$. For bandwidth selection, we devise an approach that is adequate for inputs of moderate dimension and sparse regression

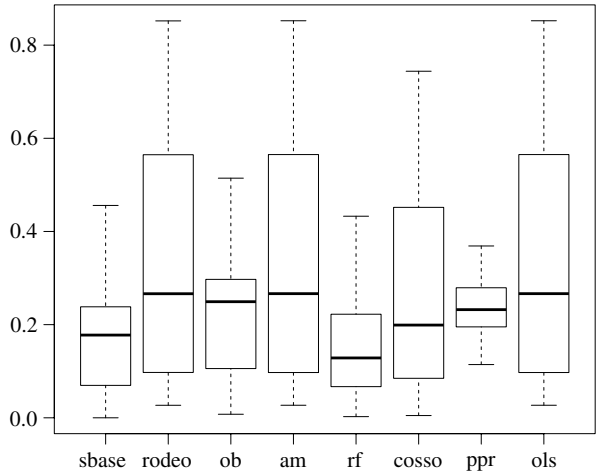

Fig. 8. Boxplots of the absolute and squared errors for each method and the *Concrete Compressive Strength* data set.

functions, which is integrated with the estimation of the regression parameters. This sparse bandwidth selection procedure is based on optimal bandwidth selection methodology. For the method introduced in Section 5 to be Bayesian, we should define a proper distribution on the bandwidth $\boldsymbol{h}$, so that, eventually, we would obtain the posterior distribution of $\boldsymbol{h}$ and also the posterior inclusion probabilities for the covariates. Unfortunately, it is not straightforward to give a coherent distribution on $\boldsymbol{h}$ and derive reasonable algorithms for inference in this setting. The method introduced in Section 5, however, still produces the posterior distributions of the remaining parameters and a sample of $\boldsymbol{h}$.

Note that other estimation methods for the subproblem of obtaining the regression parameters are possible as long as they provide an estimate of σ^2 and $\boldsymbol{\alpha}^2$, which are required for the bandwidth selection procedure introduced in Section 3. The Bayesian formulation is, however, a good choice, as it naturally provides estimates of σ^2 and $\boldsymbol{\alpha}^2$.

It is worth noting that *sbase* relies on some assumptions that can be removed in exchange for computational cost. For example, we assume homoscedasticity. This is to avoid the need for estimating σ_i^2 at each point. Also, we are implicitly assuming that the estimated optimal regularization parameters $\hat{\boldsymbol{\alpha}}^2$ are adequate for all points of the data set, whereas, in fact, they are locally estimated. Note that the estimated optimal regularization parameters $\hat{\boldsymbol{\alpha}}^2$ are globally used to estimate the Hessian for each data point within the optimal bandwidth estimation procedure. One possibility would be to perform the described EM algorithm for each data point, i.e., to alternate Eqs. (9), (11) and (14) for all $\boldsymbol{x}_i$. We have empirically observed, however, that, with regard to $\hat{\boldsymbol{\alpha}}^2$, the results of the algorithm do not change much with this modification.

A further step would be to consider a fully adaptive ridge regression procedure, where the regularization penalty is a symmetric positive definite matrix (no longer diagonal). The minimax efficiency of adaptive ridge regression for quadratic losses was studied, for example, by Strawderman (1978). As future work, we plan to examine the statistical efficiency of this method for both vector and full matrix penalties.

Another potential extension of the proposed approach is to use full-matrix bandwidths instead of diagonal bandwidths. For example, Horová et al. (2013) have recently explored the estimation of full matrix bandwidths in the related context of density estimation. How to integrate this sort of approach within the proposed methodology is also a possible future work direction. Note that variable selection may be more involved in this setting.

Acknowledgments

Research partially supported by the Spanish Ministry of Science and Innovation, projects TIN2010-20900-C04-04, Consolider Ingenio 2010-CSD2007-00018 and Cajal Blue Brain. We thank Lijian Yang and Rolf Tschernig for their useful comments about optimal bandwidth selection in multivariate local regression.

References

Breiman, L., 2001. Random forests. Machine Learning 45, 5–32.
Cleveland, W.S., Devlin, S.J., 1988. Locally weighted regression: an approach to regression analysis by local fitting. Journal of the American Statistical Association 83, 596–610.
Coppersmith, D., Winograd, S., 1990. Matrix multiplication via arithmetic progressions. Journal of Symbolic Computation 3, 251–280.
Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B 39, 1–38.
Fan, J., 1993. Local linear regression smoothers and their minimax efficiencies. Annals of Statistics 21, 196–216.
Friedman, J.H., Stuetzle, W., 1981. Projection pursuit regression. Journal of the American Statistical Association 76, 817–823.
Györfi, L., Kohler, M., Krzyzak, A., Walk, H., 2002. A Distribution-Free Theory of Nonparametric Regression. Springer.
Hall, P., Li, Q., Racine, J.S., 2007. Nonparametric estimation of regression functions in the presence of irrelevant regressors. The Review of Economics and Statistics 89, 784–789.

Hastie, T., Loader, C., 1993. Local regression: automatic kernel carpentry. Statistical Science 8, 120–143.
Hastie, T., Tibshirani, R., 1990. Generalized Additive Models. Chapman and Hall.
Horová, I., Kolácek, J., Vopatová, K., 2013. Full bandwidth matrix selectors for gradient kernel density estimate. Computational Statistics & Data Analysis 57, 364–376.
Lafferty, J., Wasserman, L., 2008. Rodeo: sparse, greedy nonparametric regression. The Annals of Statistics 36, 28–63.
Lin, Y., Zhang, H.H., 2006. Component selection and smoothing in smoothing spline analysis of variance models. The Annals of Statistics 34, 2272–2297.
Loader, C., 1999. Local Regression and Likelihood. Springer.
Ruppert, D., Wand, M., 1994. Multivariate locally weighted least squares regression. Annals of Statistics 22, 1346–1370.
Sain, S.R., Baggerly, K.A., Scott, D.W., 1994. Cross-validation of multivariate densities. Journal of the American Statistical Association 89, 807–817.
Strawderman, W.E., 1978. Minimax adaptive generalized ridge regression estimators. Journal of the Royal Statistical Society: Series B 73, 623–627.
Vidaurre, D., Bielza, C., Larrañaga, P., 2012. Lazy lasso for local regression. Computational Statistics 27, 531–550.
Wand, M.P., Jones, M.C., 1994. Multivariate plug-in bandwidth selection. Computational Statistics 9, 97–117.
Yang, L., Tschernig, R., 1999. Multivariate bandwidth selection for local linear regression. Journal of the Royal Statistical Society: Series B 61, 793–815.