An introduction to dimension reduction in nonparametric kernel regression

Stéphane Girard, Jerôme Saracco

▶ **To cite this version:**

Stéphane Girard, Jerôme Saracco. An introduction to dimension reduction in nonparametric kernel regression. D. Fraix-Burnet; D. Valls-Gabaud. Regression methods for astrophysics, 66, EDP Sciences, pp.167-196, 2014, EAS Publications Series, 10.1051/eas/1466012 . hal-00977512

HAL Id: hal-00977512
https://hal.science/hal-00977512

Submitted on 11 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

AN INTRODUCTION TO DIMENSION REDUCTION IN NONPARAMETRIC KERNEL REGRESSION

STÉPHANE GIRARD AND JÉRÔME SARACCO

Abstract. Nonparametric regression is a powerful tool to estimate nonlinear relations between some predictors and a response variable. However, when the number of predictors is high, nonparametric estimators may suffer from the curse of dimensionality. In this chapter, we show how a dimension reduction method (namely Sliced Inverse Regression) can be combined with nonparametric kernel regression to overcome this drawback. The methods are illustrated both on simulated datasets as well as on an astronomy dataset using the **R** software.

1. Introduction

This chapter is an introduction to the use of dimension reduction methods in nonparametric regression. However, the literature on this topic is huge and outside the scope of this chapter. Here, we focus on nonparametric regression using the kernel estimator and on dimension reduction using Sliced Inverse Regression (SIR).

SIR has been introduced by Li (1991). Chen and Li (1998) provide a very interesting discussion on this method and mention numerous references. There is still currently a lot of applied or theoretical works published on SIR approach in statistical journals.

For more details on nonparametric regression, the reader could refer to the following books: Schimek (2000) is an introduction to several nonparametric estimators (kernel, local polynomial, splines, wavelets ...) from the statistical point of view. The machine learning point of view is presented in Hastie *et al* (2009) while Härdle (1990) concentrates on the applied aspects.

More generally, the goal of nonparametric statistics is the estimation of functionals of the distribution function (such as the density, the regression function, quantiles ...). The problem is twofold. First, to estimate the quantity of interest as accurately as possible, and second, to keep the estimator as tractable and understandable as possible. Available methods can be divided in two main families: parametric and nonparametric. The parametric approach assumes that the underlying distribution function follows a certain parametric model, and the problem consists of estimating a finite number of parameters describing the model. For instance, linear regression is a particular case of parametric regression methods. At the opposite, the nonparametric approach makes little or no assumptions about the underlying distribution function.

Clearly, since nonparametric approaches require fewer assumptions, their potential applications are much wider than the parametric ones. This increased flexibility has however a price to pay. Nonparametric methods suffer from the curse of dimensionality. For instance, the efficiency of nonparametric regression strongly

decreases as the number of predictors increases, see Paragraph 2.3.2 for more details. To overcome this drawback, it is proposed to combine the nonparametric estimation method with a dimension reduction technique. Here, we focus on multiple indices models which assume that the response variable only depends on few linear combinations of the predictors (the so-called indices). One ends up with a two-step regression method. The first step consists of estimating the indices (here, using the SIR method) and the second step is the nonparametric estimation of the regression/link function (here, using the kernel method) on the small set of indices. The underlying regression model is called semi-parametric since it mixes parametric aspects (associated to the indices) with nonparametric ones (associated to the link function).

The chapter is organized as follows. Section 2 is dedicated to nonparametric kernel regression while Section 3 focuses on dimension reduction based on SIR. Finally, Section 4 illustrates how nonparametric kernel regression and SIR can be combined to analyse an astronomy dataset.

2. An introduction to nonparametric kernel regression

This section first presents the basic ideas of nonparametric kernel estimation in the density estimation framework. Kernel estimators are presented as an extension of the well-known histogram in Section 2.1. Section 2.2 is dedicated to unidimensional kernel regression. In this case, the unknown regression function is univariate. The multidimensional version is presented in Section 2.3. Finally, some extensions to kernel regression are reviewed in Section 2.4.

2.1. Nonparametric density estimation.

In this paragraph, X_i, $i = 1, \ldots, n$ are supposed to be independent and identically distributed observations from a density function f. Here, the unknown density function $f : \mathbb{R} \to \mathbb{R}_+$ is the object of interest.

Histogram. The simplest nonparametric density estimator is the histogram. It computes the number of observations that fall into distinct intervals, also called bins, and denoted by B_j, $j = 1, \ldots, J$. The histogram permits to rapidly visualize the general shape of the distribution. It can be computed as

$$\hat{f}(x) = \frac{1}{n} \sum_{j=1}^{J} \frac{1}{h_j} \sum_{i=1}^{n} \mathbb{I}\{(x, X_i) \in B_j \times B_j\},$$

where h_j is the length of the bin B_j. The main drawbacks of this estimator are i) $\hat{f}$ is never continuous even though the true density f is smooth, and ii) $\hat{f}$ heavily depends on the choice of the bins.

Moving average estimator. The moving average estimator can be seen as a first step towards the building of the kernel estimator. The idea is to start from the interpretation of the density f as the derivative of the cumulative distribution function F:

$$f(x) = \lim_{h \to 0} \frac{F(x+h) - F(x-h)}{2h}.$$

The moving average estimator is then obtained by approximating $f(x)$ by the right-hand side of the previous equation and by replacing F by its empirical counterpart

(the empirical cumulative distribution function):

$$\hat{f}(x) = \frac{1}{2nh} \sum_{i=1}^{n} \mathbb{I}\{|x - X_i|/h \leq 1\}. \tag{1}$$

The moving average estimator can be interpreted as an histogram where the bins would have equal length $h_1 = h_2 = \cdots = 2h$ and would depend on the estimation point x. Compared to the histogram, this estimator is simpler to use: it avoids the choice of a bins sequence. However, the moving average estimator suffers from the same drawback as the histogram, it is discontinuous.

Kernel estimator. The kernel density estimator is obtained by replacing the indicator function in (1) by a more general function K:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right). \tag{2}$$

In this context, K is a given density function called the kernel. In practice, K is often chosen to be symmetric around zero and with compact support (see Table 1 and Figure 1 for examples). Estimator (2) is also known as the Parzen-Rosenblatt estimator, since it has been first introduced in Parzen (1962) and Rosenblatt (1956). It is important to note that the kernel density estimator is itself a proper density function: it is positive and integrates to one. Moreover, since the estimator $\hat{f}$ has the same regularity as K, it is possible to build kernel density estimators of arbitrary smoothness. The parameter h is called the smoothing parameter or bandwidth, its role is discussed in Paragraphs 2.2.2, 2.3.2, 2.3.3.

2.2. **Unidimensional nonparametric regression.** In this section, (X_i, Y_i), $i = 1, \ldots, n$ are supposed to be independent and identically distributed observations from the model $Y = r(X) + \varepsilon$ where X, Y and ε are real random variables. This is the so-called random design framework. Here, the unknown regression function $r : \mathbb{R} \to \mathbb{R}$ is the object of interest. It is further assumed that X and ε are independent and that ε is centred. It immediately follows that

$$r(x) = \mathbb{E}(Y|X = x), \tag{3}$$

which is the key equation for building nonparametric estimators of r. Finally, recall that the density function of X is denoted by f. The simplest estimator of (3) is the moving average estimator described in Paragraph 2.2.1. It can be interpreted as a particular kernel estimator, as shown in Paragraph 2.2.2.

2.2.1. *Moving average estimator.* The idea of the moving average estimator consists in approximating the mathematical expectation in (3) by its empirical counterpart. The mean value of Y given $X = x$ is estimated by the mean of the Y_i's for which the corresponding X_i's are close to x:

$$\hat{r}(x) = \sum_{i=1}^{n} Y_i \mathbb{I}\{|x - X_i| \leq h\} \Bigg/ \sum_{i=1}^{n} \mathbb{I}\{|x - X_i| \leq h\}. \tag{4}$$

In this context, X_i is close to x if the distance $|x - X_i|$ is less than a given threshold $h > 0$. Clearly, the estimator (4) is the adaptation of the density estimator (1) to the regression framework. Both estimators suffer from the same drawback: they are not continuous even though the target function is smooth, see the left panel of Figure 2 for an illustration. The discontinuity can be avoided. Rather than give all

the points in the neighbourhood $[x-h, x+h]$ equal weight, it is possible to assign weights $w(x, X_i)$ that varies smoothly with distance $|x-X_i|$ to the estimation point x:

$$\hat{r}(x) = \sum_{i=1}^{n} Y_i w(x, X_i). \tag{5}$$

In the general case, it is only assumed that the weights sum to one:

$$\sum_{i=1}^{n} w(x, X_i) = 1,$$

for all $x \in \mathbb{R}$. In the following, we focus on weights given by

$$w(x, X_i) = K\left(\frac{x-X_i}{h}\right) \Bigg/ \sum_{j=1}^{n} K\left(\frac{x-X_j}{h}\right), \tag{6}$$

where K is a given positive function. Let us note that the denominator in (6) ensures the sum-to-one property.

2.2.2. *Kernel estimator.* Plugging (6) into (5), one obtains the so-called univariate kernel regression estimator

$$\hat{r}(x) = \sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right) Y_i \Bigg/ \sum_{i=1}^{n} K\left(\frac{x-X_i}{h}\right). \tag{7}$$

Recall that K is a given univariate positive function called the kernel, and $h > 0$ is a smoothing parameter called the bandwidth. Note that it is not required that K integrates to one since the estimator (7) only depends on ratios of the kernel function. Similarly to their roles in the kernel density estimator (2), the kernel function K determines the shape of the weights assigned to the neighbourhood of x while the bandwidth h controls the size of this neighbourhood. In particular, the estimator (2) directly inherits its regularity properties from those of the kernel K, see Table 1 for examples and Figure 1 for illustrations. The moving average estimator (4) can be seen as a particular case of (7) where the uniform kernel is used. Let us also note that the infinite differentiability property of the Gaussian kernel is obtained at the expense of a non compact support, which may cause some severe boundary problems, see the corresponding paragraph of this section. Estimator (7) is also known as the Nadaraya-Watson estimator. It has been introduced independently by the two authors in Nadaraya (1964) and Watson (1964). As an example, the estimation of the regression function using the Epanechnikov kernel is depicted on the right panel of Figure 2. It appears that, on the same sample, and using the same bandwidth $h = 0.1$, the resulting estimate is much smoother than the estimate obtained with the uniform kernel, *i.e* the moving average estimator.

Bandwidth selection. The bandwidth selection for kernel estimators is as crucial as for the histogram estimator. As pointed out in Paragraph 2.1, one of the main issues when making a histogram is the choice of the bins length. Let us assume that all bins have equal length $h_1 = \cdots = h_J = 2h$. If the bins are too wide, they will contain a large number of observations, leading to an over-smoothed estimation of f, where the different modes of the distribution could disappear. In such a case, the estimation bias is important whereas the variance is small. At the opposite, if the bins are too small, the histogram will be under-smoothed and artificial modes may

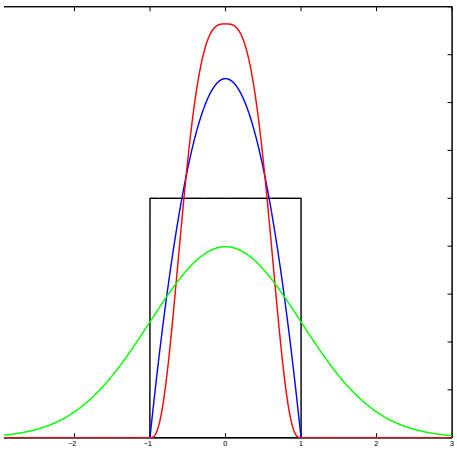

FIGURE 1. Example of kernels. They are normalized in order to integrate to one. Black: Uniform, Blue: Epanechnikov, Red: Tricube, Green: Gaussian.

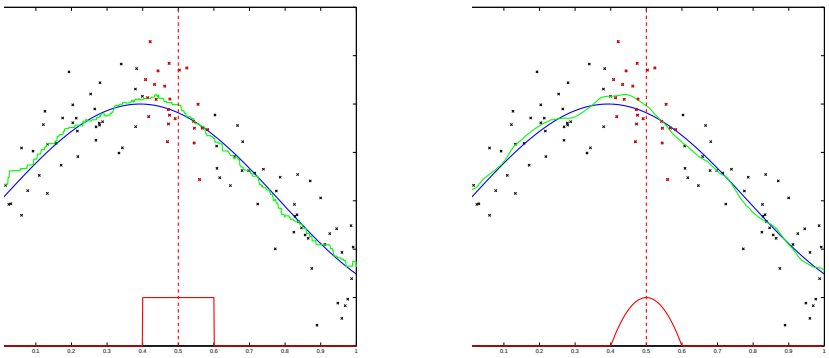

FIGURE 2. Comparison between two kernel estimators: Uniform - moving average estimator (left) and Epanechnikov (right). In both cases, the smoothing parameter is fixed to $h = 0.1$. $n = 100$ pairs (X_i, Y_i) are generated from the model $Y = \sin(4X) + \varepsilon$, $X \sim U[0,1]$ and $\varepsilon \sim N(0, 1/9)$. The blue curve is the true regression function $r(x) = \sin(4x)$. The green curve is the kernel estimate. The red curve is the kernel function (translated and rescaled for visualization purposes). The red dashed line indicates the point $x_0 = 1/2$ and the red points represent the observations contributing to the fit at x_0.

Name	$K(u)$	Regularity
Uniform	$\mathbb{I}\{\|u\| \leq 1\}$	discontinuous
Epanechnikov	$(1-u^2)\ \mathbb{I}\{\|u\| \leq 1\}$	continuous
Bi-quadratic	$(1-u^2)^2\ \mathbb{I}\{\|u\| \leq 1\}$	differentiable
Tri-cube	$(1-\|u\|^3)^3\ \mathbb{I}\{\|u\| \leq 1\}$	twice differentiable
Gaussian	$\exp(-u^2/2)$	infinitely differentiable

TABLE 1. Example of kernels

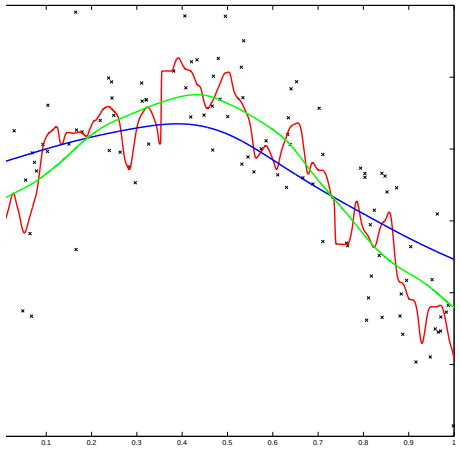

FIGURE 3. Bandwidth selection. $n = 100$ pairs (X_i, Y_i) are generated from the model $Y = \sin(4X) + \varepsilon$, $X \sim U[0,1]$ and $\varepsilon \sim N(0, 1/9)$. The kernel estimator has been computed using the tri-cube kernel. Three bandwidths have been tested $h = 0.5$ (blue line), $h = 0.2$ (green line), $h = 0.03$ (red line). Large h implies low variance (average over many observations) but high bias (the true function f is assumed to be constant within the window).

appear. Here, the estimation bias is small, but the variance is large since, in each bin, the estimator uses only few observations. This is the classical bias/variance trade-off in statistical estimation, see Paragraph 2.3.2 for mathematical derivations.

The same problem occurs with kernel estimators, both in the density estimation and in the regression estimation frameworks. In all cases, one has to find a bandwidth that produces an estimator with a good balance between bias and variance. In the regression case, a small bandwidth essentially yields an interpolation of the data, while a large bandwidth leads to a flat line. This phenomena is illustrated on Figure 3 where the kernel estimator is computed three times on the same dataset but with three different bandwidths. We refer to Paragraph 2.3.3 for a selection method of h in practical situations.

Boundary effects. Kernel estimators suffer from bias problems near the boundaries of the domain. This phenomena is illustrated on the left panel of Figure 4. On

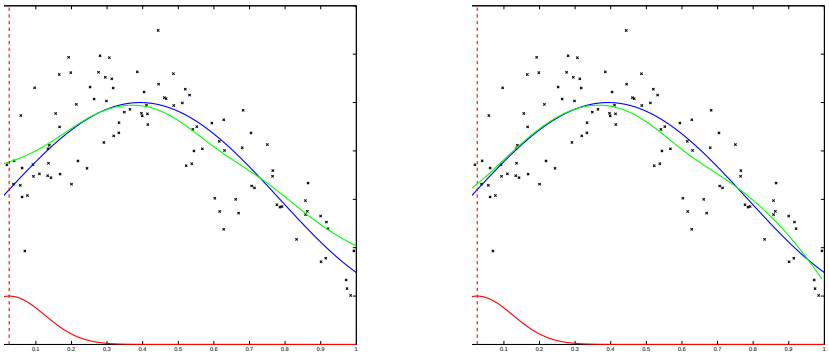

FIGURE 4. Comparison between kernel (left) and local-linear estimators (right). In both cases, the Gaussian kernel is used and the smoothing parameter is fixed to $h = 0.1$. $n = 100$ pairs (X_i, Y_i) are generated from the model $Y = \sin(4X) + \varepsilon$, $X \sim U[0,1]$ and $\varepsilon \sim N(0, 1/9)$. The blue curve is the true regression function $r(x) = \sin(4x)$. The green curve is the estimate. The red curve is the kernel function (translated and rescaled for visualization purposes). The red dashed line indicates the point $x_0 = 0.025$.

this example, the kernel estimator over-estimates the regression function at the boundaries of the $[0,1]$ interval. It appears that most observations Y_i in the neighbourhood of $x_0 = 0.025$ have a higher mean than the target point $r(x_0)$. As a consequence, their weighted mean $\hat{r}(x_0)$ is biased upwards. This so-called boundary effect is even worse in the multidimensional case, since the fraction of points on the boundary of the estimation domain is larger. Moreover, the fraction of points close to the boundary tends to one as the dimension increases. This is the curse of dimensionality. The first order of the bias induced by boundary effects can be removed using local polynomial regression, see Paragraph 2.4.2.

2.3. **Multidimensional nonparametric regression.** The framework is very similar to the one of Section 2.2: (X_i, Y_i), $i = 1, \ldots, n$ are still supposed to be independent and identically distributed observations from the model $Y = r(X) + \varepsilon$ where Y and ε are real random variables, but from now on, X is a p-dimensional random vector. Thus, the unknown regression function defined by (3) is $p-$ variate, $r : \mathbb{R}^p \to \mathbb{R}$.

2.3.1. *Kernel estimator.* Formally, the multidimensional kernel regression estimator is the same as is the unidimensional setting (7):

$$\hat{r}(x) = \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) Y_i \Bigg/ \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right), \tag{8}$$

except that the kernel $K : \mathbb{R}^p \to \mathbb{R}^+$ is a $p-$ variate positive function. For the sake of simplicity, the same smoothing parameter h has been used for all the X coordinates, see Paragraph 2.4.1 for more general cases. Let us note that the

estimator can be rewritten as $\hat{r}(x) = \hat{\psi}(x)/\hat{f}(x)$ with

$$\hat{\psi}(x) = \frac{1}{nh^p} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) Y_i$$

and where $\hat{f}$ is a multidimensional kernel estimator of the density f:

$$\hat{f}(x) = \frac{1}{nh^p} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right),$$

see also (2) for the unidimensional case. Here, $\hat{\psi}(x)$ can be interpreted as an estimator of $\psi(x) = \int y\phi(x, y)dy$ where ϕ is the joint density of (X, Y). As a consequence, estimator (8) can be also read as an estimator of $r(x) = \psi(x)/f(x)$ where both upper and lower parts are themselves estimated with kernel estimators.

2.3.2. *Bias/variance trade-off.* In this paragraph, we show how to derive the asymptotic bias and variance of kernel estimators. Elementary considerations yield

(9)
$$\mathbb{E}\hat{f}(x) = \frac{1}{h^p}\mathbb{E}\left(K\left(\frac{x - X}{h}\right)\right) = \int_{\mathbb{R}} \frac{1}{h^p} K\left(\frac{x - t}{h}\right) f(t)dt = \int_{\mathbb{R}} K(u)f(x - hu)du.$$

If f is twice continuously differentiable and the kernel is centred *i.e* $\int_{\mathbb{R}} uK(u)du = 0$, then a second order Taylor expansion of $f(x - hu)$ yields

$$\begin{aligned}
\mathbb{E}\hat{f}(x) &= \int_{\mathbb{R}} K(u)\left[f(x) - hu^t\nabla f(x) + \frac{1}{2}h^2u^tH_f(x)u(1 + o(1))\right]du \\
&= \int_{\mathbb{R}} K(u)du\ f(x) + \frac{1}{2}h^2 \int_{\mathbb{R}} u^tH_f(x)u\ du\ (1 + o(1)),
\end{aligned}$$

where ∇f is the gradient of f and H_f is the Hessian matrix. It follows that, as $h \to 0$,

$$\mathbb{E}\hat{f}(x) = \int_{\mathbb{R}} K(u)du\ f(x) + O(h^2).$$

It thus appears that $\hat{f}(x)$ is an asymptotically unbiased estimator of $f(x)$ provided that $\int_{\mathbb{R}} K(u)du = 1$, that is, K is itself a density function. Similarly, assuming that r is twice continuously differentiable, we have

$$\begin{aligned}
\mathbb{E}\hat{\psi}(x) &= \frac{1}{h^p}\mathbb{E}\left(K\left(\frac{x - X}{h}\right)Y\right) = \frac{1}{h^p}\mathbb{E}\left(\mathbb{E}\left(K\left(\frac{x - X}{h}\right)Y|X\right)\right) \\
&= \frac{1}{h^p}\mathbb{E}\left(K\left(\frac{x - X}{h}\right)\mathbb{E}\left(Y|X\right)\right) = \frac{1}{h^p}\mathbb{E}\left(K\left(\frac{x - X}{h}\right)r(X)\right) \\
&= \int_{\mathbb{R}} \frac{1}{h^p} K\left(\frac{x - t}{h}\right) r(t)f(t)dt = \int_{\mathbb{R}} \frac{1}{h^p} K\left(\frac{x - t}{h}\right) \psi(t)dt.
\end{aligned}$$

Comparing with (9), the same kind of integral has to been evaluated, f being replaced by ψ. It thus follows that

$$\mathbb{E}\hat{\psi}(x) = \int_{\mathbb{R}} K(u)du\ \psi(x) + O(h^2),$$

and $\hat{\psi}(x)$ is an asymptotically unbiased estimator of $\psi(x)$ provided that $\int_{\mathbb{R}} K(u)du = 1$. The calculations of variances are similar:

$$\begin{aligned}\mathbb{V}\hat{f}(x) &= \frac{1}{nh^{2p}}\mathbb{V}\left(K\left(\frac{x-X}{h}\right)\right)\\ &= \frac{1}{nh^{2p}}\left[\mathbb{E}\left(K^2\left(\frac{x-X}{h}\right)\right)-\mathbb{E}^2\left(K\left(\frac{x-X}{h}\right)\right)\right]\\ &= \frac{1}{nh^{p}}\left[\mathbb{E}\left(\frac{1}{h^p}K^2\left(\frac{x-X}{h}\right)\right)-h^p\mathbb{E}^2\left(\frac{1}{h^p}K\left(\frac{x-X}{h}\right)\right)\right].\end{aligned}$$

Both terms can be evaluated with (9). In the first one, the kernel K^2 is involved, while in the second term, the kernel K is used. One has:

$$\begin{aligned}\mathbb{V}\hat{f}(x) &= \frac{1}{nh^{p}}\left[\int_{\mathbb{R}} K^2(u)du\ (f(x)+o(1))-h^p\left(\int_{\mathbb{R}} K(u)du(f(x)+o(1))\right)^2\right]\\ &= \frac{1}{nh^{p}}\int_{\mathbb{R}} K^2(u)du\ f(x)(1+o(1)).\end{aligned}$$

Let us remark that $\mathbb{V}\hat{f}(x)\to 0$ as $h\to 0$ under the condition $nh^p\to\infty$. In such a case, the kernel density estimator is consistent. To conclude, we shall admit that the previous results remain true for the kernel regression estimator. First, the asymptotic bias of $\hat{r}(x)$ is proportional to h^2 and second, the variance of $\hat{r}(x)$ is proportional to $1/(nh^p)$. The interpretation of these results is clear. If the bandwidth h is small, $\hat{r}(x)$ is the average of a small number of Y_i and the variance is large. At the opposite, the bias is small since, for the selected observations in the neighbourhood of x, one has $Y_i\simeq r(X_i)\simeq r(x)$. If the bandwidth h is large, the variance of $\hat{r}(x)$ is small because of the averaging effect. The bias is large since observations X_i far from x are used while there is no guarantee that $r(X_i)\simeq r(x)$.

The best compromise is obtained by balancing the squared bias and the variance, leading to $h^4\simeq 1/(nh^p)$ and thus $h\simeq n^{-1/(p+4)}$. The asymptotic mean-squared error is thus of order

$$\mathbb{E}(\hat{r}(x)-r(x))^2=(\mathbb{E}\hat{r}(x)-r(x))^2+\mathbb{V}\hat{r}(x)\simeq n^{-\frac{4}{p+4}}. \tag{10}$$

We refer to Ferraty & Vieu (2006), Chapter 6 for a complete proof of this result. From a theoretical point of view, it appears that the speed of convergence decreases as the dimension p of the predictor X increases. Again, this is an illustration of the curse of dimensionality. From a practical point of view, the previous asymptotic derivations do not help for choosing the bandwidth h in finite sample situations. A simple selection technique is proposed hereafter.

2.3.3. *Bandwidth selection with cross-validation.* The cross-validation (or leave-one-out) method is a popular way to select the bandwidth in nonparametric statistics. Consider a set $\mathcal{H}=\{h_1,h_2,\dots\}$ of possible values for the bandwidth. The idea is to remove one observation (X_j,Y_j) from the original sample $(X_1,Y_1),\dots,(X_n,Y_n)$, to compute the estimator on the sample of size $(n-1)$ and then to compute the prediction error for the removed point (X_j,Y_j). The selected bandwidth $h\in\mathcal{H}$ is the one minimizing the prediction error. The algorithm can be written as follows:

- For each $h\in\mathcal{H}$
- For $j\in\{1,\dots,n\}$

- Compute the kernel regression estimator at point X_j on the sample excluding (X_j, Y_j) with bandwidth h:

$$\hat{r}_{-j}(X_j) = \sum_{i \neq j} K\left(\frac{X_j - X_i}{h}\right) Y_i \Bigg/ \sum_{i \neq j} K\left(\frac{X_j - X_i}{h}\right)$$

- Compute the associated prediction error: $\hat{\varepsilon}_j^2 = (Y_j - \hat{r}_{-j}(X_j))^2$
- Choose h such that $\sum_{j=1}^n \hat{\varepsilon}_j^2$ is the smallest.

2.3.4. *An illustration on simulated data.* As an illustration, the above cross-validation procedure is implemented in **R** and tested on a simulated dataset (with size $n = 100$) generated from the model $Y = \sin(4X) + \varepsilon$ where $X \sim U[0,1]$ and $\varepsilon \sim N(0, 1/9)$. The code writes as follows:

```
# Simulation of the data
n <- 100
x <- runif(n,0,1)
epsilon <- rnorm(n,0,sd=1/9)
y <- sin(4*x)+epsilon

# Cross-validation function
CV.prog <- function(x,y,hmin=(max(x)-min(x))/20,
                    hmax=(max(x)-min(x))/2,nbh=25,graph=TRUE){
   n <- length(x)
   vecth <- seq(from=hmin,to=hmax,length=nbh)
   matCV <- cbind(vecth,rep(0,nbh))
   for (h in 1:nbh){
        ypred <- rep(0,n)
        for (i in 1:n){
               ypred[i] <- ksmooth(x[-i],y[-i],kernel="normal",
                           band=vecth[h],x.points=x[i])$y
               }
        matCV[h,2] <- sum((y-as.matrix(ypred,ncol=1))^2)
        }
   hopt <- matCV[which(matCV[,2]==min(matCV[,2],na.rm=TRUE)),1]
   if (graph==TRUE) {
               plot(matCV,xlab="h",ylab="CV(h)")
               abline(v=hopt,col=2,lwd=3)
               }
   list(matCV=matCV,hopt=hopt)
   }

# Estimation of the link function via kernel smoothing :
# we use "ksmooth" function with a Gaussian kernel
# and the optimal CV bandwidth.
resCV <- CV.prog(x,y)
resKhopt <- ksmooth(x,y,kernel="normal",bandwidth=resCV$hopt)
par(mfrow=c(1,2)) #  split the graphical device in one row
                  # and two columns
plot(x,y,pch=4)  # scatterplot of the data
```

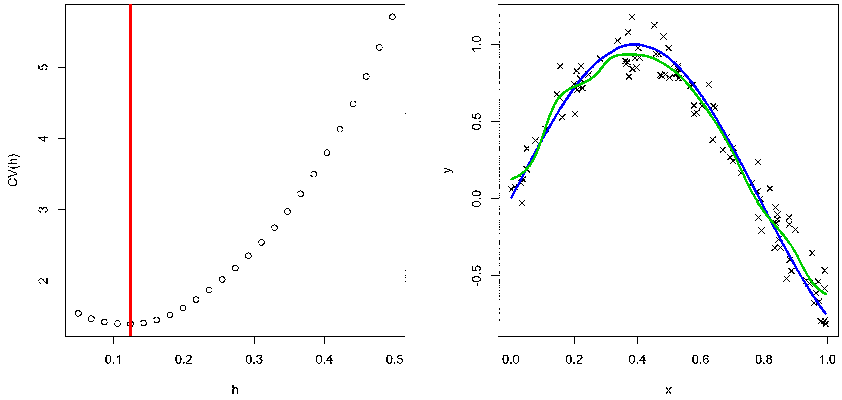

FIGURE 5. Bandwidth selection. $n = 100$ pairs (X_i, Y_i) are generated from the model $Y = \sin(4X) + \varepsilon$, $X \sim U[0,1]$ and $\varepsilon \sim N(0, 1/9)$. Left panel: Cross-validation criterion computed for 25 candidate values of h. The red vertical line indicates the selected bandwidth. Right panel: The blue curve is the true regression function $r(x) = \sin(4x)$. The green curve is the kernel estimate computed with the Gaussian kernel and the bandwidth parameter selected by cross-validation.

```
# plot of the true link function (blue line)
lines(sort(x),sin(4*sort(x)),col=4,lwd=3)
# plot of the estimated link function (green line)
lines(resKhopt,col=3,lwd=3)
```

The plots produced by this code are displayed on Figure 5. It appears on the left panel that the cross-validation criterion selects a bandwidth $h \simeq 0.12$. The associated kernel estimator is plotted on the right panel and compared to the true regression function. The regression curves are very similar.

2.4. **Some extensions.** Three extensions to the original kernel estimator (8) are reviewed. The use of structured kernels is presented in Paragraph 2.4.1. We show how they may allow to dampen the curse of dimensionality or deal with functional covariates. It is shown in Paragraph 2.4.2 that the kernel estimator can be seen as a particular case of local polynomial regression (constant polynomial). It is also illustrated how using a local linear regression permits to reduce boundary effects. Finally, a more general regression framework is presented in Paragraph 2.4.3 including some classification problems.

2.4.1. *Structured kernels.*
Multidimensional kernels. As mentioned in Section 2.3, the estimator (8) assumes a common bandwidth h for all the p components of X (predictors). A simple solution to give different weights to different predictors (or combinations of predictors) is to combine a positive semi-definite matrix A with a univariate kernel $\tilde{K}$. The

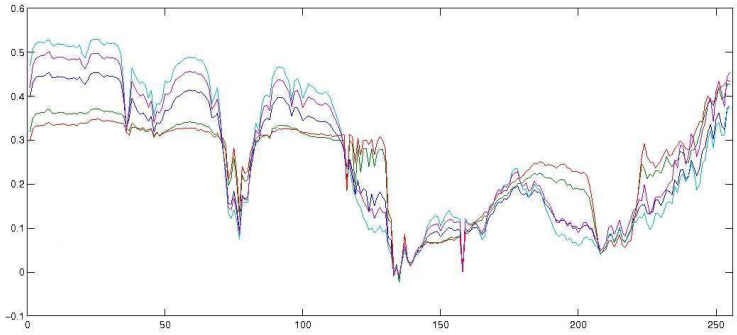

FIGURE 6. Example of hyperspectral data. Five spectra are depicted, they are discretized on 256 bands.

corresponding estimator can be written as

$$\hat{r}(x) = \sum_{i=1}^{n} \tilde{K}\left(\frac{(X_i - x)^t A (X_i - x)}{h}\right) Y_i \Bigg/ \sum_{i=1}^{n} \tilde{K}\left(\frac{(X_i - x)^t A (X_i - x)}{h}\right).$$

where $\tilde{K} : \mathbb{R}^+ \to \mathbb{R}^+$. Particular directions can be downgraded or omitted by imposing appropriate restrictions on A. For example, when A is diagonal, then one can increase or decrease the influence of the jth predictor by increasing or decreasing A_{jj}. Similarly, SIR method (see Section 3) can be used to build low rank matrices A and thus reduce the curse of dimensionality. In such a case, the initial dimension p of the predictor is replaced by the effective rank of A in the mean-squared error, see (10).

Functional covariates. In the case where the covariate X is no longer a $p-$ dimensional random vector but a random function, the estimator (8) can still be used under a slightly different form. Introducing a semi-metric d between functions, a possible functional kernel regression estimator is

$$\hat{r}(x) = \sum_{i=1}^{n} \tilde{K}\left(\frac{d(X_i, x)}{h}\right) Y_i \Bigg/ \sum_{i=1}^{n} \tilde{K}\left(\frac{d(X_i, x)}{h}\right),$$

where $\tilde{K} : \mathbb{R}^+ \to \mathbb{R}^+$. A popular choice of semi-metric between two functions is the L_2 distance but semi-metrics based on derivatives are also of interest, see Ferraty & Vieu (2006) for more details on functional estimation. The case of functional covariates appears for instance when dealing with hyperspectral data, see Figure 6 for an illustration. In such a case, the observations $X_1, \ldots, X_n$ are spectra. Modeling such observations by functions permits to take into account the natural order between spectral bands.

2.4.2. *Local polynomial regression.* Local polynomial regression is an extension of kernel regression which permits to overcome some of its drawbacks. For the sake of simplicity, it is presented here in the unidimensional setting (see Section 2.2) but it is also available in higher dimension.

Another interpretation of kernel regression estimator. Consider the following minimization problem at point $x \in \mathbb{R}$:

$$\hat{\beta}_0(x) = \arg\min_{\beta_0} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)(Y_i - \beta_0)^2$$

and set $\hat{r}(x) = \hat{\beta}_0(x)$. This regression scheme consists in approximating the unknown regression function $r(x)$ by a constant β_0 in a neighbourhood of x. Clearly, this optimization problem is quadratic and thus benefits from a close-form solution

$$\hat{r}(x) = \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) Y_i \Bigg/ \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right),$$

which is exactly the kernel regression estimator (7). It thus appears that the kernel regression estimator can be interpreted as a local approximation of $r(x)$ by a constant. Local polynomial estimators are just an extension of this principle to polynomials of arbitrary degree.

Local polynomial estimator. Let $d \in \mathbb{N}$, consider the following minimization problem at point $x \in \mathbb{R}$:

$$(\hat{\beta}_0(x), \ldots, \hat{\beta}_d(x)) = \arg\min_{\beta_0,\ldots,\beta_d} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)\left(Y_i - \sum_{j=0}^{d} \beta_j (X_i - x)^j\right)^2 \tag{11}$$

and set

$$\hat{r}(x) = \sum_{j=0}^{d} \hat{\beta}_j(x)(X_i - x)^j. \tag{12}$$

The quadratic optimization problem (11) can be rewritten using some matrix notations. To this end, let us introduce $\mathbf{X}$ the $n \times (d+1)$ matrix with ith line given by

$$\mathbf{X}_{i,\cdot} = \left(1, (X_i - x), (X_i - x)^2, \ldots, (X_i - x)^d\right)$$

and $\mathbf{W}$ the $n \times n$ diagonal matrix of weights given by

$$\mathbf{W} = \operatorname{diag}\left(K\left(\frac{x - X_1}{h}\right), \ldots, K\left(\frac{x - X_n}{h}\right)\right).$$

Finally, let $\beta \in \mathbb{R}^{d+1}$ defined by $\beta = (\beta_0, \beta_1, \ldots, \beta_d)^t$. The optimization problem (11) can be rewritten as

$$\hat{\beta}(x) = \arg\min_{\beta} (\mathbf{X}\beta - Y)^t \mathbf{W} (\mathbf{X}\beta - Y).$$

This least-squared optimization problem benefits from a closed-form solution

$$\hat{\beta}(x) = (\mathbf{X}^t \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^t \mathbf{W} Y. \tag{13}$$

Of course, when $d = 0$, one can check that plugging (13) in (12) gives back the kernel regression estimator. In practice, one may consider $d = 1$ (local linear regression) or $d = 2$ (local quadratic regression). Local linear regression permits to cancel the fist order of the bias at the boundaries, see the right panel of Figure 4. Local quadratic regression may be used to reduce the estimation bias in the regions where the regression function has a high curvature. Nevertheless, these bias corrections also imply larger variances, this is why the case $d > 2$ is usually not considered.

2.4.3. *Local likelihood.* In this paragraph, a more general model than the classical regression context is considered. It is assumed that Y given X follows a parametric distribution indexed by $\theta(X)$. The function θ is unknown and is the object of interest. Starting from a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ from this model, $\theta(x)$ is estimated by minimizing the local negative log-likelihood:

$$\hat{\theta}(x) = \arg\min_{\theta} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)(-\log \mathcal{L}(Y_i, \theta)), \tag{14}$$

where $\mathcal{L}(y, \theta)$ is the likelihood associated to the parametric model. This general framework may encompass various situations.

Example 1. Let us consider the regression model $Y = \mu(X) + \varepsilon$, with $\varepsilon \sim N(0, \sigma^2)$ and where μ is the target function. Here, Y given X follows a $N(\mu(X), \sigma^2)$ distribution and thus (14) can be rewritten as

$$\hat{\mu}(x) = \arg\min_{\mu} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)(Y_i - \mu)^2.$$

A close-form solution can be obtained, and we find back the kernel regression estimator (8):

$$\hat{\mu}(x) = \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) Y_i \Bigg/ \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right).$$

As a conclusion, the kernel regression estimator can be interpreted as a maximum local-likelihood estimator in a Gaussian regression model.

Example 2. Let us consider the classification model where $Y \in \{0, 1\}$ given X follows a Bernoulli $\mathcal{B}(p(X))$ distribution. The function of interest is p and the maximum local-likelihood estimator (14) can be simplified as

$$\hat{p}(x) = \arg\min_{p} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)(-Y_i \log(p) + (Y_i - 1)\log(1 - p)),$$

leading back to the kernel estimator (8):

$$\hat{p}(x) = \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) Y_i \Bigg/ \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right).$$

3. Dimension reduction based on sliced inverse regression

In this section, we focus on a dimension reduction regression to model the link between a response variable Y and a p-dimensional covariate X. This considered model is a semi-parametric one since it contains a parametric part (via the presence of a small number of indices, that is linear combinations of the covariate $X^t\beta_k$, $k = 1, \ldots, K$ with $K \leq p$) and a functional part (through the link function between the response variable and the indices). The dimension reduction will be effective when $K < p$. The underlying motivation behind this model is

- to keep practical and visual aspects in order to have an easier interpretation: it is possible thanks to the indices which allow the user to produce graphics (when $K = 1$ or 2), for instance scatterplots of Y versus the indices, and to quantify the effect of each component of X on Y using the estimated indices;

- and to circumvent the curve of dimensionality in nonparametric estimation of the link function: this goal is also reached since the dimension of the explanatory part in the regression model has been reduced thanks to the indices again. For instance, the kernel estimation of the link function is more efficient when the dimension K of the explanatory part of the model is low.

In Section 3.1, the dimension reduction model is presented. The Sliced Inverse Regression (SIR) approach is described in Section 3.2, this method provides a way to estimate the indices in the model without estimating the link function. Some extensions of SIR are given in Section 3.3. Then, in Section 3.4, we introduce the closest submodel selection (CSS) methodology which provides a way to select the most informative components of X in the estimated indices. Finally, Section 3.5 illustrates how nonparametric kernel regression and SIR can be combined to estimate the euclidean and functional parameters of the semi-parametric dimension reduction model on a simulated dataset.

3.1. **The semi-parametric regression model.** Let us consider a univariate response variable Y and a p-dimensional covariate $X \in \mathbb{R}^p$ with $\mu = \mathbb{E}(X)$ and $\Sigma = \mathbb{V}(X)$. Let $\beta = [\beta_1, \ldots, \beta_K]$ be a matrix $p \times K$ (with $K \leq p$) where the β_k's are unknown p-dimensional vectors assumed linearly independent. The semi-parametric regression model

$$Y = g(X^t\beta, \epsilon) \tag{15}$$

is an attractive dimension reduction approach to model the effect of X on Y. The function g is an unknown arbitrary link function (with no shape assumptions) and the error term ϵ is assumed to be independent of X (with no distribution assumption). This kind of model is called link-free and distribution-free. Another way to understand the underlying dimension reduction framework is to write

$$Y \perp X \mid \beta^t X, \tag{16}$$

which means that Y is independent to X given $\beta^t X$. Therefore, we can replace $X \in \mathbb{R}^p$ by the index $X^t\beta \in \mathbb{R}^K$ without loss of information on the regression of Y on X. Note that the matrix β always exists: by default, $\beta = I_p$ and there is no dimension reduction. However, when $K < p$, there is an effective dimension reduction.

Remark on the identifiability of β. Let $S(\beta)$ be the K-dimensional linear subspace of $\mathbb{R}^p$ spanned by the columns of β. Without additional assumptions on g and β, the parameter β is not entirely identifiable. Only the subspace $S(\beta)$ is identifiable. Duan and Li (1991) and Li (1991) called this subspace the effective dimension reduction (EDR) subspace. Moreover any direction belonging to this subspace is called an EDR direction. Other authors refer to this subspace as the dimension reduction subspace (DRS) or the central subspace (which is defined as the smallest DRS), see Cook (1998) for more details. To illustrate this point, let us consider the example of a single index model ($K = 1$). Letting $a \neq 0$, we have

$$y = g(\beta^t x, \varepsilon) = g\left(\frac{1}{a}(a\beta)^t x, \varepsilon\right) = \tilde{g}((a\beta)^t x, \varepsilon).$$

Clearly, only the direction of β is identifiable since we can not differentiate β from $a\beta$ without additional assumptions on g and β.

When the dimension p of X is high, it is often difficult to have knowledge about the structure of the relationship between the response and the covariate. Hence, this semi-parametric regression model appears to be a nice alternative to parametric modeling and nonparametric modeling (which suffers from the well-known curse of dimensionality, see Paragraph 2.3.2). Note that the idea of dimension reduction in model (15) is intuitive since it aims at constructing a low dimensional projection of the covariate without losing information to predict the response Y. When the dimension K of the EDR subspace is sufficiently small ($K \ll p$), it first facilitates data visualization and explanation, and it alleviates, in a second step, the curse of the dimensionality to nonparametrically estimate g, see Section 2.

In model (15), an important purpose is to estimate the EDR subspace from a sample $\{(X_i, Y_i),\ i = 1, \ldots, n\}$ in a first step. We thus obtain estimated indices $\{\hat{B}^t X_i,\ i = 1, \ldots, n\}$ where $\hat{B} = [\hat{b}_1, \ldots, \hat{b}_K]$ denotes an estimated basis of the EDR subspace. Then in a second step, the link function g is estimated using the sample $\{(Y_i, \hat{B}^t X_i),\ i = 1, \ldots, n\}$ with nonparametric kernel estimator (8) for instance. Hence, for a given value x_0 of X, the prediction $\hat{f}(\hat{B}' x_0)$ of Y can be provided.

3.2. **The basic ideas behind SIR.** Most of the existing approaches to estimate the EDR subspace are usually based on the eigen-decomposition of a specific matrix of interest. The most popular one is SIR introduced by Duan and Li (1991) and Li (1991), respectively for single index models ($K = 1$) and multiple indices models ($K \geq 1$). SIR has been extensively studied, see for instance Carroll and Li (1992), Chen and Li (1998), Zhu *et al* (2007), Bercu *et al* (2011), Azais *et al* (2012), among others.

In the following, we only focus on the SIR approach when the sample size n is larger than the dimension p of the covariate X.

We first give a characterization of the EDR subspace based on SIR from a population point of view. Then we will provide the corresponding estimation process of the EDR subspace. In the name of the SIR method, the *Inverse* corresponds to the use of a geometrical property of the expectation of X given Y, that is of $\mathbb{E}[X|Y]$, while the word *Sliced* stands for the discretization of Y in order to simplify the expression (and consequently the estimation) of the moments used in the geometrical property.

Inverse regression step. The basic principle of the SIR method is to reverse the role of Y and X, that is, instead of regressing the univariate variable Y on the p-dimensional covariate X, the covariate X is regressed on the response variable Y. The price we have to pay to succeed in inverting the role of X and Y in order to retrieve the EDR subspace is an additional assumption on the distribution of X, named the linearity condition (described hereafter).

Let us now recall the geometric property on which SIR is based. To this end, let us introduce the linearity condition:

$$\text{(LC): } \forall b \in \mathbb{R}^p, \mathbb{E}(X^t b | x^t \beta) \text{ is linear in } x^t \beta. \tag{17}$$

The reader can find an interesting discussion on this linearity condition in Chen and Li (1998). Note that this condition is satisfied when X is elliptically distributed (for instance normally distributed). Moreover, simulation studies showed that SIR is robust to violation of (LC) from a practical point of view, and Hall and Li (1993) mentioned that, for large dimension p, that (LC) is fulfilled from a theoretical points of view.

Assuming model (15) (or model (16)) and (LC), Li (1991) showed that the centred inverse regression curve is contained in the linear subspace spanned by the $p\times K$ matrix $\Sigma\beta$. He considered the eigen-decomposition of the Σ-symmetric matrix $\Sigma^{-1}M$ where $M = \mathbb{V}(\mathbb{E}(X|T(Y)))$, where T denotes a monotonic transformation of Y. Straightforwardly, the eigenvectors associated with the largest K eigenvalues of $\Sigma^{-1}M$ are some EDR directions.
Some theoretical details for the single index model ($K = 1$). From (LC), it follows that $\mathbb{E}[X|X^t\beta] = \mu + \frac{\Sigma\beta\beta^t(X-\mu)}{\beta^t\Sigma\beta}$. Under (16), we have:

$$\mathbb{E}[X|T(Y)] = \mathbb{E}\left[\mathbb{E}\{X|X^t\beta, T(Y)\}|T(Y)\right] = \mathbb{E}\left[\mathbb{E}\{X|X^t\beta\}|T(Y)\right],$$

and thus

$$\mathbb{E}[X|T(Y)] = \mu + c(Y)\Sigma\beta \text{ with } c(Y) = \frac{\mathbb{E}\left[\beta^t(X-\mu)|T(Y)\right]}{\beta^t\Sigma\beta}.$$

As a consequence, we obtain:

$$\begin{aligned} M &= \mathbb{V}(\mathbb{E}[X|T(Y)]) \\ &= \mathbb{E}[\mathbb{E}[X|T(Y)]\mathbb{E}[X|T(Y)]^t] - \mathbb{E}[\mathbb{E}[X|T(Y)]]\mathbb{E}[\mathbb{E}[X|T(Y)]]^t \\ &= \mathbb{E}[(\mu + c(Y)\Sigma\beta)(\mu + c(Y)\Sigma\beta)^t] - \mu\mu^t \\ &= \mu\mu^t + \mathbb{E}[c(Y)^2]\Sigma\beta\beta^t\Sigma + \mathbb{E}[c(Y)]\mu\beta^t\Sigma + \mathbb{E}[c(Y)]\Sigma\beta\mu^t - \mu\mu^t \\ &= \mathbb{E}[c(Y)^2]\Sigma\beta\beta^t\Sigma \end{aligned}$$

Hence, for any vector $\tilde{b}$ which is Σ-orthogonal to β, we have $\Sigma^{-1}M\tilde{b} = 0$. Then the eigenvector b associated with the non-null eigenvalue of $\Sigma^{-1}M$ is an EDR direction (i.e. is collinear to β).
Slicing step. To easily estimate the matrix M, Li (1991) proposed a transformation T, called a slicing, which categorizes the response Y into a new response with $H > K$ levels (in order to avoid an artificial reduction of dimension). The support of Y is partitioned into H non-overlapping slices $s_1, \ldots, s_H$. With such transformation T, the matrix of interest M can be easily now written as $M = \sum_{h=1}^{H} p_h(m_h - \mu)(m_h - \mu)^t$ where $p_h = \mathbb{P}(Y \in s_h)$ and $m_h = \mathbb{E}(X|Y \in s_h)$. Let us denote by $b_k,\ k = 1, \ldots, K$ the eigenvectors associated with the largest K eigenvalues of $\Sigma^{-1}M$ which are EDR directions.
Sample version of SIR: estimation process. When a sample $\{(X_i, Y_i), i = 1, \ldots, n\}$ is available, matrices Σ and M are estimated by substituting empirical versions of the moments for their theoretical counterparts. Let

$$\widehat{M} = \sum_{h=1}^{H} \hat{p}_h(\hat{m}_h - \hat{\mu})(\hat{m}_h - \hat{\mu})^t, \tag{18}$$

where $\hat{p}_h = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[Y_i \in s_h]$ and $\hat{m}_h = \frac{1}{n\hat{p}_h}\sum_{i=1}^{n} X_i\mathbb{I}[Y_i \in s_h]$. Therefore the estimated EDR directions are the eigenvectors $\hat{b}_k,\ k = 1, \ldots, K$associated with the K largest eigenvalues of $\widehat{\Sigma}^{-1}\widehat{M}$. They span the K-dimensional estimated EDR subspace.

Note that the practical choice of the slicing T is discussed in Li (1991) or Saracco (2001), but theoretically, there is no optimal one. In practice, the number of observations per slice is fixed to $\lfloor n/H \rfloor$ where $\lfloor a \rfloor$ stands for the integer part of a. If the sample size n is not proportional to the number H of slices, some slices will then contain $\lfloor n/H \rfloor + 1$ observations. Finally let us also highlight that the main

advantage of SIR method from the numerical point of view is that this method is computationally very fast.
Some asymptotic properties. The convergence at rate $\sqrt{n}$ of the estimated EDR directions has been shown: $\hat{b}_k = b_k + 0_p(n^{-1/2})$ for $k = 1, \ldots, K$. Moreover the asymptotic normality of the estimated EDR directions has been also obtained, see Li (1991), Hsing and Carroll (1992) , Zhu and Ng (1995) or Saracco (1997) for instance.
Determination of the dimension K of the EDR subspace. Concerning the determination of the dimension K of the EDR subspace (which is unknown in practice), several works are available in the literature. Some of them are based on hypothesis testing procedures, see for example Li (1991), Schott (1994), Ferré (1998) or Bai and He (2004). Liquet and Saracco (2008, 2012) propose a graphical tool for selecting the number of slices and the dimension which is available in the **R** package `edrGraphicalTools`.

3.3. **Some extensions of SIR approach.** There is a huge literature on SIR approach, more than 190 papers in statistical journals. Let us here mention a few extensions (to usual SIR method described previously) among others.
Alternative to slicing step. In order to avoid the choice of a slicing, alternative SIR methods have been investigated. For instance, one can mention kernel-based methods of SIR proposed by Zhu and Fang (1996) or Aragon and Saracco (1997). However, these methods are hard to implement and are computationally slow. Moreover, Bura and Cook (2001) introduced a parametric version of SIR while Hsing (1999) proposed nearest neighbour inverse regression. Note that Aragon and Saracco (1997) introduced the pooled slicing approach which consists in using information from several slicings in state of only one arbitrary one. Kuentz *et al* (2010) recommend to use bagging versions of SIR.
Use of higher conditional moment of X given Y. The usual SIR method is based on the conditional expectation of X given Y. It is also possible to retrieve information from higher (inverse) conditional moments of X given Y to estimate the EDR space. For instance, SIR-II approach is based on property of $\mathbb{V}(X|T(Y))$, see Li (1991) or Yin and Seymour (2005) for details. Other alternative methods exist such as sliced average variance estimation (SAVE), see Cook (2000), Zhu and Zhu (2007), Li and Zhu (2007), Prendergast (2007) for example. Note also that Zhu *et al* (2007) proposed hybrid methods of inverse regression-based algorithms.
Tackle the issue when $n < p$. Usual SIR requires the inverse of Σ. Then, from a practical point of view, it is necessary to inverse an estimate $\widehat{\Sigma}$ of Σ. This matrix is singular when $n < p$. Moreover, it is also often ill-conditioned when $n \approx p$. Thus SIR naturally fails in these cases. A way to overcome this issue is to consider regularized version of SIR. Among others, Bernard-Michel *et al* (2009a, 2009b), Zhong *et al* (2005), Scrucca (2007), Li and Yin (2008) proposed regularizations added to the SIR method to find EDR estimates. Coudret *et al* (2014) introduced the SIR-QZ method based on the use of the QZ algorithm (see Moler and Stewart (1973) for instance) which allows to solve generalized eigenvalue problem without requiring any matrix inversion.
Multivariate SIR.. When the dependent variable Y is q-dimensional, several SIR approaches have been developed. For instance, we can mention complete slicing method, pooled marginal slicing method (which combines information from the q marginal SIR based on X and each component of Y), alternating SIR,... For

more details, the reader can refer to Aragon *et al* (2003), Setodji and Cook (2004), Saracco (2005), Barreda *et al* (2007), Lue (2009) or Coudret *et al* (2013) for instance.

3.4. **Closest Submodel Selection (CSS).** Concerning the selection of useful predictors in the indices, Zhong *et al* (2005) use a chi-square test to find which components of X affect Y, while the approach of Li and Yin (2008) relies on a Lasso penalization. In this section, we focus on another procedure, named CSS and proposed by Coudret *et al* (2014). The idea of the procedure described here is to select submodels of (15) with only a given number p_0 of components of X which are the closest to the initial one based on all the p components of X. The latter model is thus taken as a benchmark. The components of X that appear the most in these submodels are naturally asserted to have a significant effect on Y. To do this, we propose the following algorithm, named CSS for closest submodel selection.

Initialize the dimension $p_0 \in]1, p[$ of each submodel , the number $N_0 \in \mathbb{N}^*$ of submodels that will be evaluated, and $\zeta \in]0, 1[$ or $\rho \in]0, 1[$ in order to determine the number N_1 of the closest submodels to analyse (see Step 5 for details).

Step 1. Compute the estimated indices $\hat{\gamma} = (X_1^t\hat{b}, \ldots, X_n^t\hat{b})^t \in \mathbb{R}^n$ X using SIR-QZ with the whole covariate X.
Let $a = 1$.

Step 2. Select randomly p_0 components of X and build the corresponding covariate $X^{(a)} \in \mathbb{R}^{p_0}$.

Step 3. Compute the SIR-QZ indices $\hat{\gamma}^{(a)} \in \mathbb{R}^n$ based on the selected components in $X^{(a)}$.

Step 4. Calculate the linear correlation between the indices $\hat{\gamma}$ and $\hat{\gamma}^{(a)}$. Let us denote by $\hat{r}^{(a)}$ the square of this correlation.
Let $a = a + 1$.
Repeat N_0 times steps 2-4.

Step 5. Consider the submodels corresponding to the N_1 largest correlations $\hat{r}^{(a)}$.
Either the user set $\zeta \in]0, 1[$ and then gets $N_1 = \zeta N_0$, or the user chose a value for ρ and then N_1 is the number of submodels such that $\hat{r}^{(a)} > \rho$.

Step 6. Count the number of occurrences of each component of X in these N_1 closest submodels. The components that affect Y are the ones that have the greater number of occurrences.

For example, in our application to astronomy data in Section 4, we have $p = 47$ and we set $p_0 = 10$, $N_0 = 5 \times 10^5$ and $\zeta = 5\%$ to determine the closest $N_1 = \zeta N_0 = 2500$ submodels. Note that choosing $p_0 \ll n$ allows the user to use classical SIR instead of SIR-QZ in Step 3 which significantly improves the computational time. Moreover, it is also possible to use SIR instead of SIR-QZ in Step 1 when $p \ll n$.

3.5. **An illustration on simulated data.** As an illustration, a dataset (with size $n = 200$) is simulated from the following single index regression model $Y = g(X^t\,\beta) + \varepsilon$ where $X \sim U[0, 1]^5$, $\varepsilon \sim N(0, 1/2)$ with $g(x) = x^3$ and $\beta = (1, 2, -1, -2, 0)^t$. The plot of the (true) index $X^t\,\beta$ versus Y is given in the left panel of Figure 7. In a first step, the direction of β is estimated by $\hat{b}$ using SIR. To this end, the function `edr` of the **R** package `edrGraphicalTools` is used. In a second step, the link function g is estimated with an univariate kernel regression based on the estimated index $X^t\,\hat{b}$ and Y. The selection of the bandwidth is achieved thanks to the `CV.prog` function given in Paragraph 2.3.4. The code writes as follows:

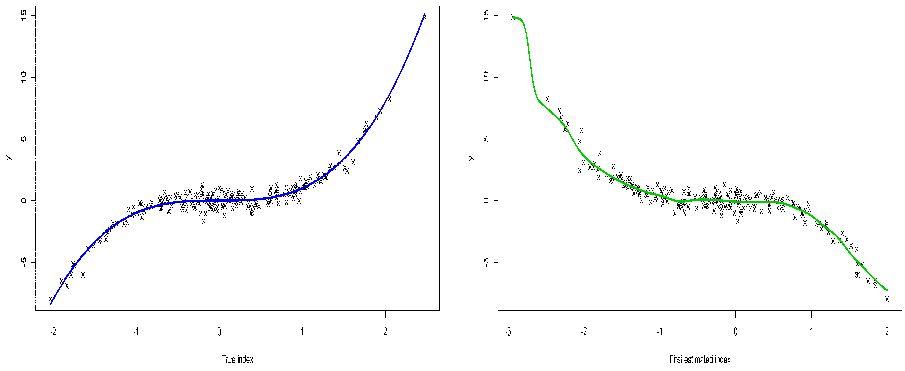

FIGURE 7. Left panel: Scatterplot of the true index versus Y, $(X_i^t\ \beta, Y_i)$ for $i = 1, \ldots, 200$ and true link function (blue line). Right panel: Scatterplot of the first estimated index versus Y, $(X_i^t\ \hat{b}_1, Y_i)$ for $i = 1, \ldots, 200$ and estimated link function (green line).

```
# Simulation of the data
n <- 200
p <- 5
beta <- matrix(c(1,2,-1,-2,0),ncol=1)
X <- matrix(runif(p*n),ncol=p)
index <- X%*%beta     # true index
epsilon <- rnorm(n,0,0.5)
y <- (index)^3+epsilon

matYX <- cbind(y,X)  # dataset
dimnames(matYX) <- list(c(1:n),c("y","x1","x2","x3","x4","x5"))
pairs(matYX,pch=4)   # matrix of scatterplots

# Plot of true index versus Y
plot(index,y,xlab="True index",ylab="y",pch=4)
u<-seq(min(index),max(index),length=50)
lines(u,u^3,,col=4,lwd=3)

library(edrGraphicalTools) # Loads the package
# Compute the EDR direction using SIR with H=5 slices
# Note that the option K=1 is only for visualization purpose.
resSIR <- edr(Y=y,X=X,H=5,K=1,method="SIR-I")
# Eigenvalue screeplot
plot(resSIR$eigvalEDR,xlab=" ",ylab="Eigenvalue")
# Display the first estimated EDR direction
resSIR

# Plots of first estimated index versus Y and
```

```
# second estimated index versus Y
indexpred1 <- X%*%resSIR$matEDR[,1]
indexpred2 <- X%*%resSIR$matEDR[,2]
par(mfrow=c(1,2))
plot(indexpred1,y,xlab="First estimated index",ylab="y",pch=4)
plot(indexpred2,y,xlab="Second estimated index",ylab="y",pch=4)

# Plot of the true index versus the first estimated index
par(mfrow=c(1,1))
plot(index,indexpred1)
title(paste("correlation=",round(cor(index,indexpred1),digit=3)))

# Select the bandwidth with the cross-validation function
# using the first estimated index and Y
resCVsir <- CV.prog(indexpred1,y)

# Estimation and plot of the link function
# using the first estimated index and Y
res1Khopt <- ksmooth(indexpred1,y,kernel="normal",
             bandwidth=resCVsir$hopt)
plot(indexpred1,y,xlab="First estimated index",ylab="y",pch=4)
lines(res1Khopt,col=3,lwd=3)
```

In Figure 8, matrix scatterplot of all the variables Y, $X_1, \ldots, X_5$ is displayed. It does not exhibit obvious structure between any component of X and Y. Figure 9 provides the eigenvalues screeplot. It clearly appears that only one EDR direction is necessary. One can observe that the right panel of Figure 7 shows a strong structure between the first estimated index and Y, while the right panel of Figure 9 does not reveal any structure. Recall that SIR is only able to retrieve the linear subspace spanned by the true direction β. The first estimated index obtained with SIR is $\hat{b}_1 = (-1.057, -2.396, 0.869, 2.200, -0.084)^t$ which is very close to $-\beta$. Similarly the sample linear correlation between the true index $X^t\ \beta$ and the first estimated one $X^t\ \hat{b}_1$ is equal to -0.995. Finally, the link function is estimated by kernel estimator using the sample $\{(X_i^t\ \hat{b}_1, Y_i),\ i = 1, \ldots, 200\}$ and the optimal bandwidth obtained with the cross-validation criterion. This function is plotted in the right panel of Figure 7. Recall that the left panel of Figure 7 presents the true link function plotted on the sample $\{(X_i^t\ \beta, Y_i),\ i = 1, \ldots, 200\}$. The two regression curves are very similar even though they are not plotted with the same horizontal axis.

4. Application to astronomy data

The SIR and kernel regression methods are illustrated on the NYU Value-Added Galaxy Catalog dataset already considered in this book. The response variable Y is the star formation rate (variable number 43) and the covariate X is $p = 47$ dimensional (variables $2, \ldots, 42$ and $45, \ldots, 50$). The original dataset is split into a training sample and a test sample both centred and with size $n = 97$ galaxies. Our methodology involves two steps. First, SIR and nonparametric kernel regression are successively applied on the whole covariates from the training sample. Since n and p are of the same order, the sample variance matrix $\widehat{\Sigma}$ of X is close to singularity and thus the SIR-QZ version of SIR is preferred. The accuracy of the

prediction is then assessed by the Root Mean-Squared Error (RMSE) evaluated on the test sample. Second, the most informative covariates are selected in the previously computed index using the CSS procedure (see Section 3.4). SIR and nonparametric kernel regression are then applied again on the subset of selected covariates. The associated RMSE is also computed. To this end, we use the same **R** package `edrGraphicalTools` as in Section 3.5. Let us highlight that the additional function `edrSelect` from this package has been used. The **R** code writes as follows:

```
library(edrGraphicalTools)
# the matrices x and xnew are centered.
# Step 1a: computation of SIR-QZ on the whole set of covariates
resSIR <- edr(Y=y,X=x,H=10,K=1,method="SIR-I",submethod="SIR-QZ")
index1 <- x%*%resSIR$matEDR[,1]
index2 < -x%*%resSIR$matEDR[,2]
plot(index1,y,pch=4,xlab="First estimated index",ylab="Y")
plot(index2,y,pch=4,xlab="Second estimated index",ylab="Y")


# Step 1b: estimation of the link function via kernel smoothing
resCV <- CV.prog(index1,y)
resKhopt <- ksmooth(index1,y,kernel="normal",bandwidth=resCV$hopt)

# Step 1c: computation of the RMSE on the test sample
index1new <- xnew%*%resSIR$matEDR[,1]
Resnewpred <- ksmooth(index1,y,kernel="normal",bandwidth=resCV$hopt,
                      x.points=index1new)
ynewpred <- Resnewpred$y
# Warning: this vector is "sorted" accordingly to the x.points vector
# ynew.sorted has thus to be sorted accordingly.
ynew.sorted <- ynew[sort(index1new,index.return=TRUE)$ix]
RMSE1 <- sqrt(mean((ynew.sorted-ynewpred)^2))

# Step 2a: selection of a subset of covariates via CSS procedure
res2<-edrSelec(Y=y,X=x,H=10,K=1,"CSS",pZero=10,NZero=50000,zeta=0.05)
plot(res2)  # plot of the CSS output
x.simple<-x[,c(41,3,2)]  # matrix of selected covariates

# Step 2b: computation of SIR on the subset of selected covariates
resSIRsimple <- edr(Y=y,X=x.simple,H=10,K=1,method="SIR-I")
index1simple<-x.simple%*%resSIRsimple$matEDR[,1]
# plot of the estimated indices obtained in steps 1a and 2b
plot(index1, index1simple,pch=4,xlab="First estimated
       index (first step)",ylab="First estimated index (second step)")
cor(index1, index1simple)  # linear correlation between these indices

# Step 2c: kernel estimation of the link function
resCVsimple <- CV.prog(index1simple,y)
resKhoptsimple <- ksmooth(index1simple,y,kernel="normal",
                          bandwidth=resCVsimple$hopt)
```

```
# Step 2d: computation of the RMSE on the test sample
xnew.simple<-xnew[,c(41,3,2)]
index1newsimple<-xnew.simple%*%resSIRsimple$matEDR[,1]
Resnewpredsimple<-ksmooth(index1simple,y,kernel="normal",
     bandwidth=resCVsimple$hopt,x.points=index1newsimple)
ynewpredsimple<-Resnewpredsimple$y
ynew.sortedsimple<-ynew[sort(index1newsimple,index.return=TRUE)$ix]
RMSE2 <- sqrt(mean((ynew.sortedsimple-ynewpredsimple)^2))

# Step 2e: plot of the link function on the training and test samples
range.y <- c(min(y,ynew),max(y,ynew))
range.x <- c(min(index1simple, index1newsimple),
             max(index1simple, index1newsimple))
plot(index1simple,y,pch=4,xlim=limit.x,ylim=limit.y,col=1,
     xlab="Estimated index",ylab="Y")
lines(resKhoptsimple,col=3,lwd=3)
par(new=TRUE)
plot(index1newsimple,ynew,pch=4,xlim=limit.x,ylim=limit.y,col=2,
     xlab="Estimated index",ylab="Y")
par(new=FALSE)
```

When comparing the left and right panels of Figure 10, only the first estimated index clearly provides a structure on the scatterplot. This indicates that only the first EDR direction $\hat{b}_1$ should be considered. The predicted values of Y are computed using an univariate kernel regression performed on this estimated index, and the associated root mean-squared error is given by $\text{RMSE}_1 \simeq 0.4283$. The index is then analysed using the CSS procedure whose results are displayed on the left panel of Figure 11. It appears that three covariates can be selected: 41, 3 and 2 which respectively correspond to `d4000n`, `Jabs` and `uabs`. The new SIR index computed on this selected subset is denoted by $\hat{b}_*$. The indices associated to $\hat{b}_1$ and $\hat{b}_*$ are compared on the right panel of Figure 11. They are visually highly correlated, which is confirmed by the linear correlation coefficient equal to -0.942. Finally, new predicted values of Y are computed using an univariate kernel regression performed on this simplified index, and the associated root mean-squared error is given by $\text{RMSE}_2 \simeq 0.3454$. In this study, simplifying the index thus permits to improve the prediction accuracy in terms of mean-squared error. Although the estimated link function seems to be linear (see Figure 12), our approach based on semi-parametric regression model outperforms the methods based on parametric (linear) models, see the numerical results in the Chapter "Linear regression in high dimension and/or for correlated inputs" of this book.

References

[1] Aragon, Y. & Saracco, J. *Sliced Inverse Regression (SIR): an appraisal of small sample alternatives to slicing.*. Computational Statistics, 12, 109–130, 1997.

[2] Azaïs, R., Gégout-Petit, A. & Saracco, J. *Optimal quantization applied to sliced inverse regression.*, Journal of Statistical Planning and Inference, 142(2), 481–492, 2012.

[3] Bai, Z. D. & He, X. *A chi-square test for dimensionality for non-Gaussian data.* Journal of Multivariate Analysis, 88, 109–117, 2004.

[4] Barreda, L., Gannoun, A. & Saracco, J. *Some extensions of multivariate SIR.* Journal of Statistical Computation and Simulation, 77(1-2), 1–17, 2007.

[5] Bercu, B., Nguyen, T.M.N. & Saracco, J. *A new approach of recursive and non recursive SIR methods.*, Journal of the Korean Statistical Society. 41, 17–36, 2011.

[6] Bernard-Michel, C., Douté, S., Fauvel, M., Gardes, L. & Girard, S. *Retrieval of Mars surface physical properties from OMEGA hyperspectral images using Regularized Sliced Inverse Regression.* Journal of Geophysical Research - Planets, 114, 2009a.

[7] Bernard-Michel, C., Gardes, L. & Girard, S. *Gaussian Regularized Sliced Inverse Regression.* Statistics and Computing, 19, 85–98, 2009b.

[8] Bura, E. & Cook, R. D. *Estimating the structural dimension of regressions via parametric inverse regression.* Journal of the Royal Statistical Society. Series B. Statistical Methodology, 63, 393–410, 2001.

[9] Carroll, R.J. & Li, K.-C. *Measurement error regression with unknown link: dimension reduction and data visualization.* Journal of the American Statistical Association, 87(420), 1040–1050, 1992.

[10] Chen, C.-H. & Li, K.-C., *Can SIR be as popular as multiple linear regression?* Statistica Sinica, 8(2), 289–316, 1998.

[11] Cook, R.D. *SAVE: a method for dimension reduction and graphics in regression.* Communications in statistics - Theory and methods, 29, 2109–2121, 2000.

[12] Cook, R.D. *Principal Hessian directions revisited (with discussion).* Journal of the American Statistical Association, 93, 84–100, 1998.

[13] Coudret, R., Girard, S. & Saracco, J. *A new sliced inverse regression method for multivariate response*, `http://hal.inria.fr/hal-00714981`, 2013.

[14] Coudret, R., Liquet, B. & Saracco, J. *Comparison of sliced inverse regression approaches for underdetermined cases.* Journal de la Société Française de Statistique, in press, 2014.

[15] Duan, N. & Li, K.-C. *Slicing regression: a link-free regression method.* Annals of Statistics, 19, 505–530, 1991.

[16] Ferraty, F. & Vieu, P. *Nonparametric functional data analysis*, 2006, Springer.

[17] Ferré, L. *Determining the dimension in Sliced Inverse Regression and related methods.*, Journal of the American Statistical Association, 93, 132–140, 1998.

[18] Hall, P. & Li, K.-C. *On almost linearity of low dimensional projections from high dimensional data.*, Annals of Statistics, 21, 867–889, 1993.

[19] Härdle, W. *Applied Nonparametric Regression*, 1990, Cambridge University Press.

[20] Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer Series in Statistics, 2009, Springer.

[21] Hsing, T. *Nearest neighbor inverse regression.* Annals of Statistics, 27, 697–731, 1999.

[22] Hsing, T. & Carroll, R.J. *An asymptotic theory for sliced inverse regression.* Annals of Statistics, 20, 1040–1061, 1992.

[23] Kuentz, V., Liquet, B. & Saracco, J. *Bagging versions of sliced inverse regression.* Communications in statistics - Theory and methods, 39(11), 1985–1996, 2010.

[24] Li, K.-C. *Sliced inverse regression for dimension reduction, with discussion.* Journal of the American Statistical Association, 86, 316–342, 1991.

[25] Li, K.-C., Aragon, Y., Shedden, K. & Agnan, C.T. *Dimension reduction for multivariate response data.* Journal of the American Statistical Association, 98(461), 99–109, 2003.

[26] Li, L. & Yin, X. *Sliced inverse regression with regularizations.* Biometrics, 64(1), 124–131, 2008.

[27] Li, Y. & Zhu, L. *Asymptotics for sliced average variance estimation.* Annals of Statistics, 35, 41–69, 2007.

[28] Liquet, B. & Saracco, J. *A graphical tool for selecting the number of slices and the dimension of the model in SIR and SAVE approaches.* Computational Statistics, 27, 103–125, 2012.

[29] Liquet, B. & Saracco, J. *Application of the bootstrap approach to the choice of dimension and the α parameter in the SIR$_\alpha$ method.* Communications in statistics - Simulation and Computation, 37(6), 1198–1218, 2008.

[30] Lue, H.-H. *Sliced inverse regression for multivariate response regression.* Journal of Statistical Planning and Inference, 139(8), 2656–2664, 2009.

[31] Moler, C.B. & Stewart, G.W., *An algorithm for generalized matrix eigenvalue problems.* SIAM Journal on Numerical Analysis, 10(2), 241–256, 1973.

[32] Nadaraya, E. A. *On Estimating Regression*. Theory of Probability and its Applications, 9(1), 141–142, 1964.
[33] Parzen, E. *On estimation of a probability density function and mode*. The Annals of Mathematical Statistics, 33,1065–1076, 1962.
[34] Prendergast, L.A. *Implications of influence function analysis for sliced inverse regression and sliced average variance estimation.* Biometrika, 94(3), 585–601, 2007.
[35] Rosenblatt, M. *Remarks on some nonparametric estimates of a density function*. The Annals of Mathematical Statistics, 832–837, 1956.
[36] Saracco, J. *Asymptotics for pooled marginal slicing estimator based on SIR_α approach.* Journal of Multivariate Analysis, 96, 117–135, 2005.
[37] Saracco, J. *Pooled slicing methods versus slicing methods.* Communications in statistics - Simulation and Computation, 30, 489–511, 2001.
[38] Saracco, J. *An asymptotic theory for Sliced Inverse Regression.* Communications in statistics - Theory and methods, 26, 2141–2171, 1997.
[39] Schott, J. R. *Determining the dimensionality in sliced inverse regression.* Journal of the American Statistical Association, 89, 141–148, 1994.
[40] Schimek, M. *Smoothing and regression: approaches, computation, and application*, 2000, Wiley.
[41] Scrucca, L. *Class prediction and gene selection for DNA microarrays using regularized sliced inverse regression.* Computational Statistics & Data Analysis, 52(1), 438–451, 2007.
[42] Setodji, C.M. & Cook, R.D. *K-means inverse regression.* Technometrics, 46, 421–429, 2004.
[43] Shao, Y., Cook, R.D. & Weisberg, S. *Partial central subspace and sliced average variance estimation.* Journal of Statistical Planning and Inference, 139(3), 952–961, 2009.
[44] Yin, X. & Seymour, L. *Asymptotic distributions for dimension reduction in the SIR-II method.* Statistica Sinica, 15, 1069–1079, 2007.
[45] Watson, G. S. it Smooth regression analysis. Sankhya: The Indian Journal of Statistics, Series A, 26(4), 359–372, 1964.
[46] Zhong, W., Zeng, P., Ma, P., Liu, J.S. & Zhu, Y. *RSIR: regularized sliced inverse regression for motif discovery.* Bioinformatics, 21(22), 4169–4175, 2005.
[47] Zhu, L.X. & Fang, K. T. *Asymptotics for kernel estimate of sliced inverse regression.* Annals of Statistics, 24, 1053–1068, 1996.
[48] Zhu, L.X. & Ng, K. W. *Asymptotics of sliced inverse regression.* Statistica Sinica, 5, 727–736, 1995.
[49] Zhu, L.X., Ohtaki, M. & Li, Y. *On hybrid methods of inverse regression-based algorithms.* Computational Statistics, 51, 2621–2635, 2007.
[50] Zhu, L. & Zhu, L. *On kernel method for sliced average variance estimation.* Journal of Multivariate Analysis, 98, 970-991, 2007.

Inria Grenoble Rhône-Alpes & Laboratoire Jean Kuntzmann

Institut Polytechnique de Bordeaux & Inria Bordeaux Sud Ouest & Institut de Mathématiques de Bordeaux

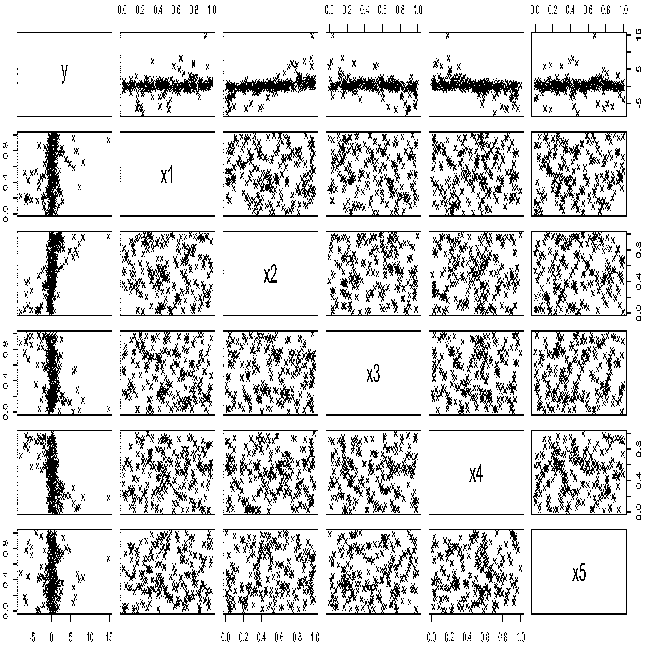

FIGURE 8. Matrix scatterplot of all the variables Y, $X_1, \ldots, X_5$.

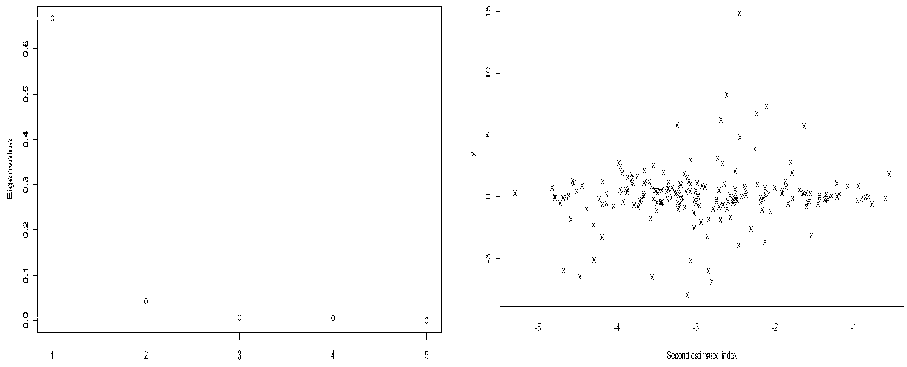

FIGURE 9. Left panel: Eigenvalues screeplot. Right panel: Scatterplot of the second estimated index versus Y, $(X_i^t\ \hat{b}_2, Y_i)$ for $i = 1, \ldots, 200$.

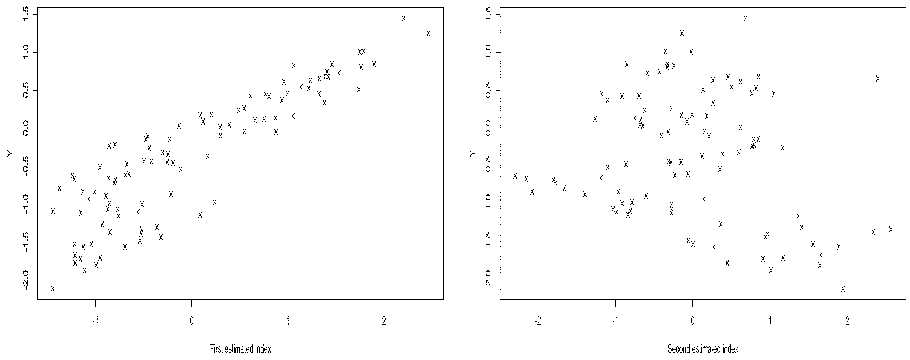

FIGURE 10. Left panel: Scatterplot (drawn on the training sample) of the first estimated index versus Y, $(X_i^t\ \hat{b}_1, Y_i)$ for $i = 1, \ldots, 97$. Right panel: Scatterplot (drawn on the training sample) of the second estimated index versus Y, $(X_i^t\ \hat{b}_2, Y_i)$ for $i = 1, \ldots, 97$.

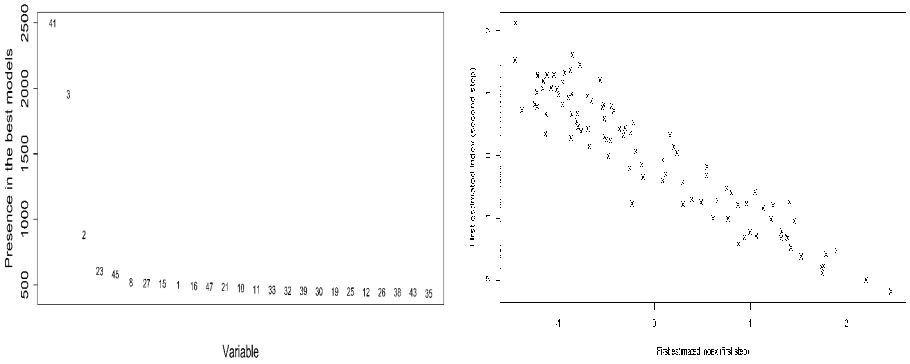

FIGURE 11. Left panel: CSS plot. The three most informative covariates (41, 3, 2) are respectively `d4000n`, `Jabs` and `uabs`. Right panel: Scatterplot (drawn on the training sample) of the first estimated index versus the simplified index $(X_i^t\ \hat{b}_1, X_i^t\hat{b}_*)$ for $i = 1, \ldots, 97$.

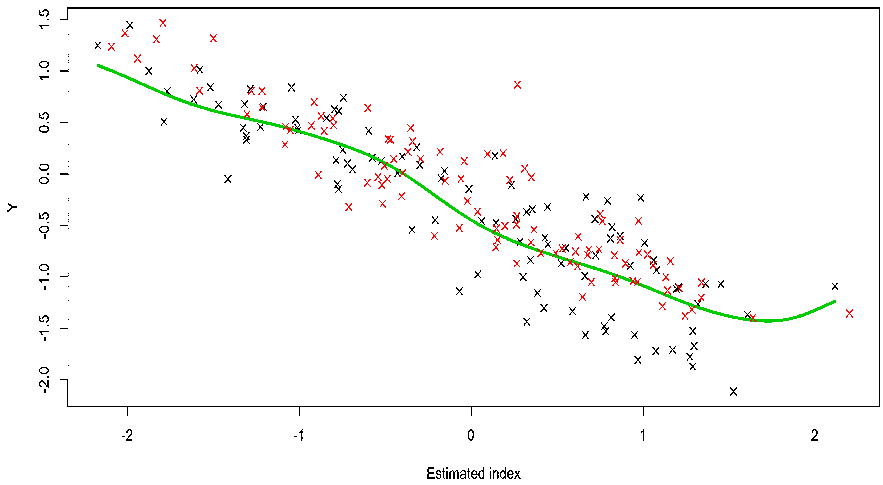

FIGURE 12. Scatterplot of the simplified index versus Y, $(X_i^t\ \hat{b}_*, Y_i)$ for $i = 1, \ldots, 97$ drawn on the training sample (black crosses) and on the test sample (red crosses). The link function estimated with nonparametric kernel regression is plotted in green.