Nonparametric Regression Based Image Analysis

Book · October 2014

DOI: 10.1007/978-1-4939-0569-0_17

Nicolas W Hengartner
Los Alamos National Laboratory

E. Matzner-Løber
École Nationale de la Statistique et de l'Administration Économique

Nonparametric regression based image analysis

Cornillon, P-A. and Hengartner, N. and Matzner-Løber, E. and Thieurmel, B.

Abstract Multivariate nonparametric smoothers are adversely impacted by the sparseness of data in higher dimension, also known as the curse of dimensionality. Adaptive smoothers, that can exploit the underlying smoothness of the regression function, may partially mitigate this effect. We present an iterative procedure based on traditional kernel smoothers, thin plate spline smoothers or Duchon spline smoother that can be used when the number of covariates is important. However the method is limited to small sample sizes ($n < 2000$) and we will propose some thoughts to circumvent that problem using for example pre-clustering of the data. Applications considered here are image denoising.

1 Introduction

The recent survey paper [11] presents modern image filtering from multiple perspectives, from machine vision, to machine learning, to signal processing; from graphics to applied mathematics and statistics. It is noteworthy that an entire section of that paper is devoted to an iteratively refined smoother for noise reduction. A similar nonparametric regression estimator was considered independently in [5], who recognized it as a nonparametric iterative bias reduction method. When coupled with a cross-validation stopping rule, the resulting smoother adapts to the underlying smoothness of the regression function. The adaptive property of this iterative bias corrected smoother helps mitigate the curse of dimensionality, and that smoother has

Cornillon, P-A.
Univ. Rennes 2, 35043 Rennes, France, e-mail: pac@uhb.fr

Hengartner, N.
Los Alamos National laboratory e-mail: nickh@lanl.gov

Matzner-Løber, E. Univ. Rennes 2, 35043 Rennes, France, e-mail: eml@uhb.fr · Thieurmel, B.
Greenwich Statistics, 75007 Paris, France e-mail: benoit.thieurmel@greenwich-statistics.com

been successfully been applied to fully nonparametric regression with moderately large number of explanatory variables [4].

This paper presents a fully nonparametric regression formulation for denoising images. In section 2, we present iterative biased regression (IBR) with kernel smoother and Duchon splines smoother. In section 3, we consider image denoising as a regression problem and evaluates IBR in that context. The proposed procedure is compared to BM3D in section 4. Section 5 gathers concluding remarks.

2 Iterative bias reduction

For completeness, we recall the definition of the iterative bias corrected smoother. Consider the classical nonparametric regression model

$$Y_i = m(X_i) + \varepsilon_i, \tag{1}$$

for the pairs of independent observations $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$, $i = 1, \dots, n$. We want to estimate the unknown regression function m assuming that the disturbances $\varepsilon_1, \dots, \varepsilon_n$ are independent mean zero and finite variance σ^2 random variables. It is helpful to rewrite Equation (1) in vector form by setting $Y = (Y_1, \dots, Y_n)^t$, $m = (m(X_1), \dots, m(X_n))^t$ and $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^t$, to get $Y = m + \varepsilon$. Linear smoothers can be written in vector format as

$$\widehat{m} = S(X, \lambda)Y,$$

where $S(X, \lambda)$ is an $n \times n$ smoothing matrix depending on the explanatory variables X and a smoothing parameter λ or a vector of smoothing parameters. The latter is typically the bandwidth for kernel smoother or the penalty for splines smoothers. Instead of selecting the optimal value for λ, [5] and [4] propose to start out with a biased smoother that has a large smoothing parameter λ (ensuring that the data are over-smoothed) and then proceed to estimate and correct the bias in an iterative fashion. If one wants to use the same smoothing matrix (denoted it simply by S from now) at each iteration, the smoother $\hat{m}_k$ after $k-1$ bias correction iterations has a closed form expression:

$$\hat{m}_k = [I - (I - S)^k]Y. \tag{2}$$

The latter shows that the qualitative behavior of the sequence of iterative bias corrected smoothers $\widehat{m}_k$ is governed by the spectrum of $I - S$ [5]. If the eigenvalues of $I - S$ are in $[0, 1[$, then $\lim_{k \longrightarrow \infty} \widehat{m}_k(X_j) = Y_j$. Thus at the observations, the bias converges to 0 and the variance increases to σ^2.

This convergence of the smoother to the data (as a function of the number of iterations) raises the question of how to select the number of iterations k. For univariate thin-plate spline base smoothers, [3] showed that there exists a $k^\star$ such that the iterative bias corrected smoother $\widehat{m}_{k^\star}$ converges in mean square error to the true regression function at the minimax convergence rate. An extension of that result to the multivariate case is presented in [5]. This optimal number of iterations can be selected from data using classical model selection criterion such as: GCV, AIC , BIC or gMDL [10]. All these rules are implemented in **ibr** package.

2.1 Adaptation to smoothness

The resulting estimator adapts to the true (unknown) smoothness of the regression function. Since points in higher dimensions are more separated from one another than their lower dimensional projections, smoothers in higher dimensions need average over larger volumes than in lower dimensions to smooth over the same number of points. Thus smoother in higher dimensions have, for similar variances, larger biases.

This effect is at the heart of the curse of dimensionality, which can also be quantified via the minimax Mean Integrated Square Error (MISE): For ν-times continuously differentiable regression functions in $\mathbb{R}^d$, the minimax MISE is of order $n^{2\nu/(2\nu+d)}$. While this rate degrades as a function of the dimension d, it is improved with increasing smoothness. For example, the minimax MISE rate of convergence of a 40-times differentiable function on $\mathbb{R}^{20}$ is the same as the minimax MISE rate of convergence of a twice differentiable function on $\mathbb{R}$. While in practice, the smoothness of regression function are not known, adaptive smoothers behave (asymptotically) as if the smoothness were known. Thus in higher dimensional smoothing problems, such as those arising in image denoising and image inpainting, adaptive smoothers are desirable as they partially mitigate the curse of dimensionality.

Iterative bias reduction can be applied to any linear smoother. For kernel smothers, the behavior of the sequence of iterative bias corrected kernel smoothers depend critically on the properties of the smoother kernel. Specifically, the smoothing kernel needs to be positive definite [7, 5]. Examples of positive definite kernels include the Gaussian and the triangle densities, and examples of kernel that are not definite positive include the uniform and the Epanechnikov kernels. In this paper we focus only on the Gaussian kernel smoother:

$$S_{ij} = \frac{K_{ij}}{\sum_{i=1}^{n} K_{ij}}, \quad \text{where } K_{ij} = \exp\{-\sum_{k=1}^{d} (X_{ik} - X_{jk})^2/(2\lambda_k^2)\},$$

and on the Duchon splines smoother presented in the next section.

2.2 Splines smoothers

The theoretical results given in [5] are given for Thin Plate Splines (TPS) smoother. Suppose the unknown function m from $\mathbb{R}^d \to \mathbb{R}$ belongs to the Sobolev space

$\mathscr{H}^{(\nu)}(\Omega) = \mathscr{H}^{(\nu)}$, where ν is an unknown integer such that $\nu > d/2$ and Ω is an open subset of $\mathbb{R}^d$, TPS is a solution of the minimization problem

$$\frac{1}{n}\|Y_i - f(X_i)\|^2 + \lambda J_\nu^d(f),$$

see for example [9], where

$$J_\nu^d(f) = \sum_{\alpha_1+\cdots+\alpha_d=\nu} \frac{\nu!}{\alpha_1!\cdots\alpha_d!} \int\cdots\int \left(\frac{\partial^\nu f}{\partial x_1^{\alpha_1}\cdots\partial x_d^{\alpha_d}}\right)^2 dx_1\cdots dx_d.$$

The first part of the functional controls the data fitting while $J_\nu^d(f)$ controls the smoothness. The trade-off between these two opposite goals is ensured by the choice of the smoothing parameter λ. The main problem of TPS is that the null space of $J_\nu^d(f)$ consists of polynomials with maximum degree of $(\nu-1)$, so its finite dimension is $M = \binom{\nu+d-1}{\nu-1}$. As $\nu > d/2$, the dimension of the null space increases exponentially with d. In his seminal paper, [8] presented a mathematical framework that extends TPS. Noting that the Fourier transform (denoted by $\mathscr{F}(.)$) is isometric, the smoothness penalty $J_\nu^d(f)$ can be replaced by its squared norm in Fourier space:

$$\int \|D^\nu f(t)\|^2 dt \quad \text{can be replaced by} \quad \int \|\mathscr{F}(D^\nu f)(\tau)\|^2 d\tau.$$

In order to solve the problem of exponential growth of the dimension of the null space of $J_\nu^d(.)$, Duchon introduced a weighting function to define:

$$J_{\nu,s}^d(f) = \int |\tau|^{2s} \|\mathscr{F}(D^\nu f)(\tau)\|^2 d\tau.$$

The solution of the new variational problem: $\frac{1}{n}\|Y_i - f(X_i)\|^2 + \lambda J_{\nu,s}^d(f)$is now

$$g(x) = \sum_{j=1}^{M_0} \alpha_j \phi_j(x) + \sum_{i=1}^{n} \delta_i \eta_{\nu,s}^d(\|x - X_i\|),$$

provided that $\nu + s > d/2$ and $s < d/2$. The $\left\{\phi_j(x)\right\}$ are a basis of the subspace spanned by polynomial of degree $\nu - 1$ and

$$\eta_{\nu,s}^d(r) \propto \begin{cases} r^{2\nu+2s-d}\log(r) & d \text{ if } 2\nu+2s-d \text{ is even,} \\ r^{2\nu+2s-d} & d \text{ otherwise} \end{cases}$$

with the same constraint as TPS that is $T\delta = 0$ with the matrix T defined as $T_{ij} = \phi_j(X_i)$. For the special case $s = 0$, Duchon splines reduce to the TPS. But if one wants to have a lower dimension for the null space of $J_{\nu,s}^d$ one has to increase s; for instance to use a pseudo-cubic splines (with an order $\nu = 2$), one can choose $s = \frac{d-1}{2}$ as suggested by Duchon [8].

3 Image denoising

An image is a matrix of pixels values, in which each pixel measures a gray scale, or a vector of color levels. Pixel are spatially defined by their positional coordinates (i, j) where $(i, j) \in \{1, \ldots, p\}^2$. In order to avoid complex notations we restrict ourselves to squared image. As an example, consider the picture of Barbara below, defined by 512×512 pixels ($p = 512$), each pixel value in $\{0, 1, \ldots, 255\}$, representing 256 gray levels. The left-hand panel of Figure 1 displays the original image, and the right-hand panel shows a noisy version, which we wish to *denoise*. The numerical measure to quantify the error is the **PSNR** (Peak Signal to Noise Ratio):

$$PSNR = 10 \times \log_{10}\left(\frac{c^2}{MSE}\right),$$

where c is the maximum possible pixel value of the image and MSE is the mean squared error between the original image and the treated image. The quality of the reconstruction is given in decibel (**dB**) and a well reconstructed image have a $PSNR$ within $[30, 40]$.

Fig. 1 Barbara, original image - noisy image, PSNR=28.14 dB.

The noisy image given Fig-1 is obtained by adding a Gaussian noise to the original image. Let us denote by Y the vector of all the gray value (for all the pixels). This vector Y is of length $n = p^2$ and its k^{th} coordinate is the gray value of pixel (i, j) where $k = (i-1) \times p + j$. Let us denote by X the $n \times d$ matrix gathering all the explanatory variables. The k^{th} row of length d (corresponding to pixel (i, j)) is usually made with the gray value of neighboring pixels ; the coordinates i and j can be added as explanatory variables. For instance, when one uses the 8 immediate neighbors and the two coordinates, the number of columns of X is $d = 10$ (see Fig 2).

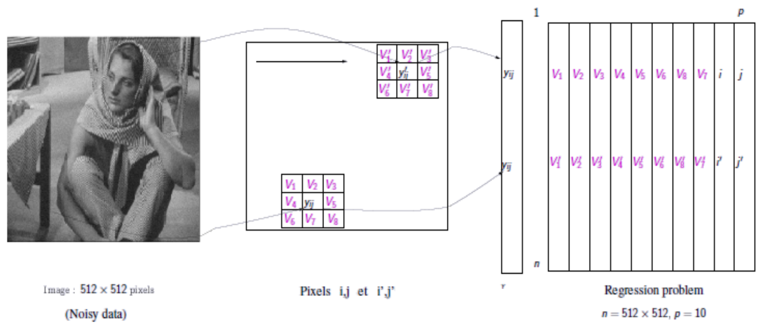

Fig. 2 Image analysis as a regression problem

Thus image analysis can be recast as a regression problem, and we can use a nonparametric smoothing approach to fit Y; thereby denoising the image. Taking a nonparametric smoothing approach brings forth two major challenges:

1. the size of the neighboring pixels can be large (one can think to use 24 or 48 neighbors) and thus the dimension d can be large. This problem is known as curse of dimensionality in the statistical literature;
2. the size $n \times n$ of the smoothing matrix S is usually very large. For the Barbara example, we have $n = 512 \times 512 = 262144$.

Nonparametric smoothing at point Y_k can roughly be thought as doing local averages of data points $Y_{k'}$ measured at covariate locations $X_{k'}$ near a given point X_k. When the number of data points n is fixed and when the dimension d increases, the expected number of points falling into a ball of center Y_k (with a given radius) is decreasing exponentially. Thus to have a constant number of points one have to increase exponentially the size of sample n or to increase the size of neighborhood which leads to biased smoother (and the finding of optimal smoothing parameter λ^* does not alleviate this problem). Thus, the use of classical nonparametric smoother such as kernel regression is far from easy with 24 or 48 neighbors and one has to use smoother that can cope with moderate dimension d such as Duchon Splines or Iterative Biased Regression (kernel or Duchon splines).

Concerning the size of smoothing matrix n, we only need to partition the dataset into sub-images. It is obvious that smoothing assumes that the signal varies smoothly with the values of the covariates. Thus, to avoid smoothing over edges which are numerous in an image, we need to partition the image into smaller homogeneous sub-images, leading to tremendous decrease of the sample size n (of sub-image). For instance, when the size of each sub-image is chosen equal to 30×30 the resulting 900 pixels in such sub-image can be used as a dataset for regression and one can apply the iterative regression estimate (Eq. 2), with Y being the level of gray of the current pixel and X being the level of gray of its 8 (or more) neighbors.

In order to partition the image into homogeneous sub-images, we used the CART algorithm [2] and regression trees. The resulting partition is data dependent and the size and shape of the sub-image varies from one sub-region to another. We used the package **rpart** to define homogeneous regions (so the explanatory variables are only the position i and j and the independent variable is the gray level Y) and within each sub-image, we applied **ibr**. We did modify the **rpart** function in order to control the maximal and minimal size of each sub-region. Specifically, we decide to partition the image of Barbara into sub-regions with at least 100 pixels and at most 700 pixels. Figure (3) shows the evolution of the partition with **rpart**; the picture on the rightmost panel has 686 regions.

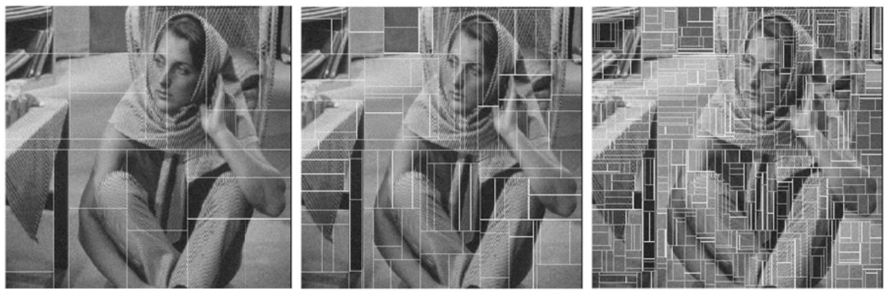

Fig. 3 Partition of the image of different size using **rpart**. The right-hand panel displays the final partition, which is composed of 686 regions.

A direct application of the CART algorithm produces rectangular regions. Such a partition will readily follow horizontal and/or vertical boundaries in the image. The Figure 4 left shows the result from applying **ibr** within each region, whereas the right side shows the difference between the denoised image and the original image.

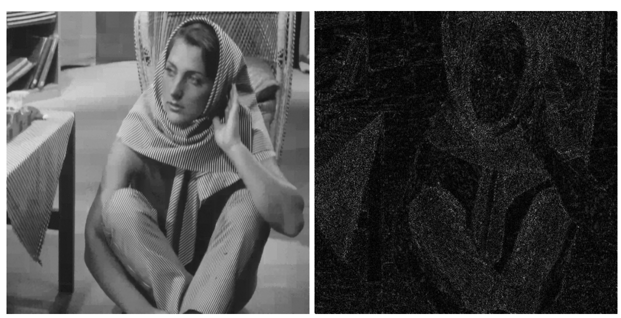

Fig. 4 Left: Denoised image (i.e fitted values) using kernel IBR applied to a noisy image. Right: the image of the residuals.

While IBR is effective at denoising the image, the denoised image has a $PSNR = 33.22dB$, one can see the boundaries of the partition. To alleviate this artifact, which is due to the fact that CART uses vertical and horizontal splits, we propose to partition the image several times at various angle of rotation, and for each partition, use **ibr** for denoising. Specifically, consider the following rotation with angle α

$$i' = i \times \cos\alpha + j \times \sin\alpha$$
$$j' = -i \times \sin\alpha + j \times \cos\alpha.$$

We now apply **rpart** to the image in that new coordinate system (i', j'). While the partition will still be rectangular (in that coordinate system), we can rotate back the partition into the original coordinate system in which the partition are now slanted. Figure 5 presents two examples of partitions.

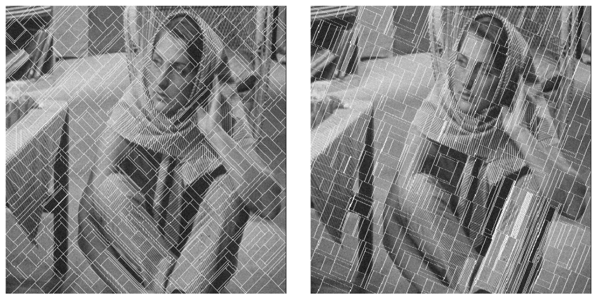

Fig. 5 Partition of the image using rotation with angle $\alpha = 30$ (left) and $\alpha = -60$ (right)

We compute a denoised image by averaging, pixel-wise, the smoothers obtained for the various rotated partitions. We found empirically that using a small number of rotations leads to better smoothers than one that use a large number of rotations. Using 3 rotations, with maximal size of 300 pixel for each region, we obtain the following result

Fig. 6 Denoised image (i.e fitted values) with IBR kernel ($PSNR = 33.72dB$) and image of the residuals (right)

The PSNR is 33.72 dB using kernel smoother and 33.60 dB using Duchon spline smoother.

4 Comparison with BM3D

BM3D algorithm, developed by [6], is the current state-of-the-art for image denoising. This method mixes block-matching (BM) and 3D filtering. This image strategy is based on an enhanced sparse representation in transform domain. The enhancement of the sparsity is achieved by grouping similar 2D image fragments (or blocks) into 3D data arrays called groups. Then, in order to deal with the 3D groups, a collaborative filtering procedure is developed. The codes and related publications could be found at http://www.cs.tut.fi/ foi/. That algorithm requires knowledge of the standard deviation of the noise. By default, the standard value is 25. Alternatively, when the noise level is unknown, one can use an estimate of the standard deviation in BM3D. We compare our regression based denoting algorithm to BM3D, using the following settings,

- BM3D, with the true standard deviation of the noise. We expect to outperform the classical BM3D for which the standard deviation is unknown and has to be estimated
- BM3D with default standard deviation of the noise
- IBR kernel, 3 rotations, maximal size = 700
- IBR Duchon, 3 rotations, maximal size = 700

on several images with different noise levels. The Matlab code for our comparison is available upon request. This comparison is conducted for 5 levels of Gaussian noise: $\sigma = 5, 10, 15, 20$ and 30 and replicated 10 times.

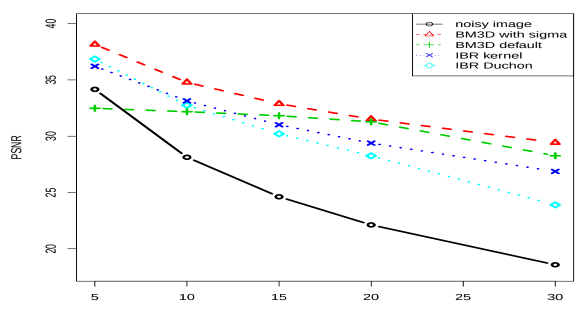

Fig. 7 Comparison BM3D versus IBR kernel: true noise level known (red) or unknown (blue).

Figure 7 shows that as the noise increases the PSNR of the denoised image decreases. The BM3D algorithm, with the true standard deviation of the noise (in red), is better than IBR (in blue) with a difference of around 2 dB. This case is not very realistic since we never know the true noise level. When the noise level is unknown, the IBR procedure is better than BM3D (with standard values) whenever the noise level is low (less than 15). IBR with kernel smoother or Duchon splines smoother give similar results.

5 Conclusion

General statistical lore suggested that fully nonparametric regression with many covariates (more than 5) should be generally avoided. The recent realization that it is possible to design simple adaptive regression smoothers, makes it now practical to smooth data in higher dimensions. While having good statistical properties, the current implementation of the IBR algorithm is limited by number of observations n.

Image denoising presents both problems simultaneously: a large number of covariates and large sample sizes. We present in this paper how to resolve both of these issues in order to apply the IBR algorithm. The initial results are promising, but the simulation study presented in this paper is done only for illustration purpose and somewhat limited. Recall that the noise assumed here is Gaussian and with the same level in the whole image.

In practical situations, the noise is usually connected to the gray level and thus its level is different within the image see for example [1]. If the partitioning into sub-images is done in the right way, the level of noise can be thought to be almost the same within sub-images but different from one sub-image to another. Thus, the IBR procedure that make the assumption of the same level of noise within sub-images can be thought to be realistic and the performance of this method have to be investigated further. The regression formulation also appears useful for image completion (inpainting) and we will investigate this topic in a future work.

References

1. Boulanger, J., Kervrann, C., Bouthemy, P., Elbau, P., Sibarita, J.B., Salamero, J.: Patch-based non-local functional for denoising fluorescence microscopy image sequences. IEEE Trans. on Medical Imaging (2009)
2. Breiman, L., Freiman, J., Olshen, R., Stone, C.: Classification and regression trees, 4 edn. CRC Press (1984)
3. Bühlmann, P., Yu, B.: Boosting with the l_2 loss: Regression and classification. J. Amer. Statist. Assoc. **98**, 324–339 (2003)
4. Cornillon, P.A., Hengartner, N., Jegou, N., Matzner-Løber: Iterative bias reduction : a comparative study. Statistics and Computing (2012)
5. Cornillon, P.A., Hengartner, N., Matzner-Løber, E.: Recursive bias estimation for multivariate regression. arXiv:1105.3430v2 (2011)
6. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3d transform-domain collaborative filtering. IEEE Trans. Image Process **16**(8), 2080–2095 (2007)
7. Di Marzio, M., Taylor, C.: On boosting kernel regression. J. Stat. Plan. Infer. **138** (2008)
8. Duchon, J.: Splines minimizing rotation-invariant semi-norms in Solobev spaces (1977)
9. Gu, C.: Smoothing spline ANOVA models. Springer (2002)
10. Hansen, M., Yu, B.: Model selection and minimal description length principle. J. Amer. Stat. Assoc. **96**, 746–774 (2001)
11. Milanfar, P.: A tour of modern image filtering. IEEE Signal Processing Magazine (2013)