LA-UR- 09-01326

Title: ibr: Iterative Bias Reduction Multivariate Smoothing

Author(s): Nicolas Hengartner
Pierre-Andre Cornillon
Eric Matzner-Lober

Intended for: Journal of Statistical Software

Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

ibr: Iterative Bias Reduction Multivariate Smoothing

Pierre-André Cornillon
Montpellier SupAgro

Nicolas Hengartner
Los Alamos National Laboratory

Eric Matzner-Løber
University Rennes

Abstract

The abstract of the article.

Keywords: multivariate smoothing, L_2 boosting, thin plate splines, kernel regression, R.

1. Introduction

Regression is a fundamental data analysis tool for relating a univariate response variable Y to a multivariate predictor $X \in \mathbb{R}^d$ from the observations $(X_i, Y_i), i = 1, \ldots, n$. Traditional nonparametric regression use the assumption that the regression function varies smoothly in the independent variable x to locally estimate the conditional expectation $m(x) = E[Y|X = x]$. The resulting vector of predicted values $\widehat{Y}_i$ at the observed covariates X_i is called a regression smoother, or simply a smoother, because the predicted values $\widehat{Y}_i$ are less variable than the original observations Y_i.

Linear smoothers are linear in the response variable Y and are operationally written as

$$\widehat{m} = S_\lambda Y,$$

where S_λ is a $n \times n$ smoothing matrix. The smoothing matrix S_λ typically depends on a tuning parameter which we denote by λ, and that governs the tradeoff between the smoothness of the estimate and the goodness-of-fit of the smoother to the data by controlling the effective size of the local neighbourhood over which the responses are averaged. We parameterise the smoothing matrix such that large values of λ are associated to smoothers that averages over larger neighbourhood and produce very smooth curves, while small λ are associated to smoothers that average over smaller neighbourhood to produce a more wiggly curve that wants to interpolate the data. The parameter λ is the bandwidth for kernel smoother, the span size for running-mean smoother, bin smoother, and the penalty factor λ for spline smoother.

Ideally, we want to choose the smoothing parameter λ to minimise the expected squared prediction error, but here, we take a different approach. Instead of optimally selecting the tuning parameter λ, we fix it to some reasonably large value that ensures that the resulting smoothers *over-smooths* the data so that the resulting smoother will have a relatively small variance but a substantial bias, and focus on correcting that bias. Our approach to bias correction rests on the observation that the conditional expectation of minus the residuals $-(Y - \widehat{Y})$, given X, is the bias of the smoother. This provides us with the opportunity to estimate the bias by smoothing the residuals R. The bias of the original smoother can be partially corrected by subtracting from it the estimated bias. This bias correction can be iteratively applied, producing a sequence of iterative bias corrected smoothers that are formally defined in Section 2.

It is well known in multivariate data analysis that the distance between typical covariates increases with increasing dimensions d of the covariates X. The resulting sparseness of the covariates, often called *the curse of dimensionality,* forces one to use larger smoothing parameters in higher dimensions, which in term leads to more biased smoothers. Optimally selecting the smoothing parameter does not alleviate this problem, and therefore, the common wisdom is to avoid general non-parametric smoothing in higher dimension and focus instead on fitting structurally constrained regression models, such as additive models Hastie and Tibshirani (1995); Linton and Nielsen (1995). Iterative Bias Reduction Smoothers depart from the classical multivariate structural regression models, and focus instead on estimating very smooth fully non-parametric regression functions.

2. Iterative bias reduction smoothers

2.1. Method

Suppose that the pairs $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$ are related through the non-parametric regression model

$$Y_i \quad = \quad m(X_i) + \varepsilon_i, \quad i = 1, \ldots, n, \tag{1}$$

where $m(\cdot)$ is an unknown smooth function, and the disturbances ε_i are independent mean zero and variance σ^2 random variables that are independent of all the covariates. It is helpful to rewrite Equation (1) in vector form by setting $Y = (Y_1, \ldots, Y_n)^t$, $m = (m(X_1), \ldots, m(X_n))^t$ and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_n)^t$, to get

$$Y \quad = \quad m + \varepsilon. \tag{2}$$

Linear smoothers can be written as

$$\widehat{m}_1 = S_\lambda Y, \tag{3}$$

where S_λ is an $n \times n$ smoothing matrix with smoothing parameter λ and $\widehat{m} = \widehat{Y} = (\widehat{Y}_1, \ldots, \widehat{Y}_n)^t$, denotes the vector of fitted values. Typical smoothing matrices are bin smoothers, regression splines, smoothing splines, thin plate splines, Nadaraya Watson kernels or local polynomials. The linear smoother (3) has bias

$$B(\widehat{m}_1) = E[\widehat{m}_1|X] - m = (S_\lambda - I)m$$

and variance

$$V(\widehat{m}_1|X) = S_1 S_1' \sigma^2,$$

respectively. To estimate the bias, observe that the residuals $R_1 = Y - \widehat{m}_1 = (I - S_\lambda)Y$ have expected value $E[R_1|X] = m - E[\widehat{m}_1|X] = (I - S_\lambda)m = -B(\widehat{m}_1)$. This suggests estimating the bias by smoothing the negative residuals

$$\widehat{b}_1 := -S_\lambda R_1 = -S_\lambda (I - S_\lambda)Y.$$

As the same smoother is assumed, the resulting estimate for the bias is zero whenever the smoothing matrix S_λ is a projection, as is the case for linear regression, bin smoothers and regression splines.

Thus in the **ibr** package, we will focus only on multivariate smoother of two types: *thin plate splines and Nadaraya Watson (product) kernel smoother*. The parameter λ is either the smoothing parameter or the bandwidths.

Repeating the bias reduction step k times produces to the linear smoother

$$\begin{aligned} \widehat{m}_k &= S_\lambda Y + S_\lambda (I - S_\lambda)Y + \cdots + S_\lambda (I - S_\lambda)^{k-1} Y \\ &= (I - (I - S_\lambda)^k)Y. \end{aligned}$$

It is useful to extend regression smoothers to enable predictions at arbitrary locations $x \in \mathbb{R}^d$ of the covariates. Such an extension allows us to assess and compare the quality of various smoothers by how well the smoother predicts new observations. To this end, write the prediction of the linear smoother S at an arbitrary location x as

$$\widehat{m}(x) = S(x)^t Y,$$

where $S(x)$ is a vector of size n whose entries are the weights for predicting $m(x)$. The vector $S(x)$ is readily computed for many of the smoothers used in practice.

Next, write the iterative bias corrected smoother $\widehat{m}_k$ as

$$\begin{aligned} \widehat{m}_k &= \widehat{m}_0 + \widehat{b}_1 + \cdots + \widehat{b}_k \\ &= S[I + (I - S) + (I - S)^2 + \cdots + (I - S)^{k-1}]Y \\ &= S\widehat{\beta}_k, \end{aligned}$$

to conclude that

$$\widehat{m}_k(x) = S(x)^t \widehat{\beta}_k \tag{4}$$

predicts $m(x)$.

2.2. Kernel smoothers

We have two types of smoother implemented in package **ibr**: Nadaraya kernel smoothers and thin plate splines smoothers. The smoothing matrix S of Nadaraya kernel type estimators has entries

$$S_{ij} = \frac{\prod_{k=1}^d K\left(\frac{(X_{ik} - X_{jk})^2}{h_k}\right)}{\sum_j \prod_{k=1}^d K\left(\frac{(X_{ik} - X_{jk})^2}{h_k}\right)}$$

where $K(.)$ is to be chosen as Gaussian, Epanechnikov, quadratic or uniform kernels.

We strongly advise the use of a Gaussian kernel because the corresponding spectrum of $I - S$ is less than 1. All kernels do not share that property (see Cornillon, Hengartner, and Matzner-Løber 2009).

The bandwidth in each component of the covariate depends on its scale. It is common to first re-scale the data before selecting the bandwidth, but here we found it preferable to leave the scales unchanged, and to *select the bandwidth based on the effective degree of freedom* (trace of the smoothing matrix) of the univariate smoother in each of the components, with typical values for the degree of freedom ranging from 1.05 to 1.2. A further advantage of the latter choice is that there is no explicit reference to sample size.

2.3. Thin plate splines smoothers

The smoother matrix S_λ of thin plate splines of order ν_0 is readily calculated using a classical method (see Gu 2002) and functions of package **fields** (Reinhard Furrer and Sain 2009). The smoother depends only on the chosen order ν_0 and the smoothing parameter λ. Recall that the thin-plate smoother of degree ν_0 minimises

$$\min_f \sum_{i=1}^{n} (Y_i - f(X_i))^2 \quad + \lambda\Gamma(\nu_0), \tag{5}$$

where

$$\Gamma(\nu_0) \quad = \quad \sum_{\substack{i_1,\dots,i_d=0\\ i_1+\dots+i_d\leq\nu_0}} \int_{\mathbb{R}^d} \left| \frac{\partial^{i_1+\dots+i_d}}{\partial x_{i_1}\dots\partial x_{i_{\nu_0}}} f(x) \right|^2 dx.$$

As the procedure is adaptive (see Cornillon *et al.* 2009), the smallest ν_0 is the best choice. Recall that the order ν_0 must be strictly greater than $d/2$ and the minimum degree of freedom is greater than $M_0 = \binom{\nu_0+d-1}{\nu_0-1}$ (see Utreras 1988). Thus, for small to moderate n and $d > 3$, usually the smoother S_λ is not smooth enough for the procedure to work. In that case Gaussian kernel smoother have to be used.

2.4. Stopping rules: choice of number of iterations k

The fitted values obtained by iterated bias reduction smoother depends on the number of iteration k. The first iteration gives by construction a very biased estimate (too much bias). When k grows to infinity, if the base smoother is well chosen (ie Gaussian kernel or thin plate splines), $\hat{m}_k$ tends to Y (too much variance). In order to have useful fitted value, we have to stop the iteration whenever a good balance between bias and variance is achieved. This is simply a choice of model and the package **ibr** offers the following stopping rules methods: GCV (on log scale), AIC, corrected AIC, BIC and gMDL (see Gu 2002; ?). The selected iteration is thus:

$$
\begin{aligned}
\hat{k}_{AIC} &= \operatorname*{argmin}_{k\in\mathcal{K}} \left\{ \widehat{\sigma}^2 + 2\frac{\mathrm{trace}(S_k)}{n} \right\}, \\
\hat{k}_{GCV} &= \operatorname*{argmin}_{k\in\mathcal{K}} \left\{ \log \widehat{\sigma}^2 - 2\log\left(1 - \frac{\mathrm{trace}(S_k)}{n}\right) \right\}, \\
\hat{k}_{AICc} &= \operatorname*{argmin}_{k\in\mathcal{K}} \left\{ \log \widehat{\sigma}^2 + 1 + \frac{2(\mathrm{trace}(S_k)+1)}{n - \mathrm{trace}(S_k) - 2} \right\}. \\
\hat{k}_{gMDL} &= \log(V) + \frac{\mathrm{trace}(S_k)\log(F)}{n}, \quad V = \frac{n\hat{\sigma}^2}{n - \mathrm{trace}(S_k)}, \quad F = \frac{\sum_{i=1}^n Y_i^2 - n\hat{\sigma}^2}{\mathrm{trace}(S_k)V}.
\end{aligned}
$$

3. Simulated example in $\mathbb{R}^2$

Define Wendelberger's test function Wendelberger (1982)

```
R> f <- function(x, y) { .75*exp(-((9*x-2)^2 + (9*y-2)^2)/4) +
+                        .75*exp(-((9*x+1)^2/49 + (9*y+1)^2/10)) +
+                        .50*exp(-((9*x-7)^2 + (9*y-3)^2)/4) -
+                        .20*exp(-((9*x-4)^2 + (9*y-7)^2)) }
```

and 50×50 grid of evaluation equally spaced between 0 and 1 (0 and 1 excluded)

```
R> ngrid <- 50; xf <- seq(0,1, length=ngrid+2)[-c(1,ngrid+2)]
R> yf <- xf ; zf <- outer(xf, yf, f)
R> grid <- cbind(rep(xf, ngrid), rep(xf, rep(ngrid, ngrid)))
```

We can plot this test function on the grid using

```
R> persp(xf, yf, zf, theta=130, phi=20, expand=0.45,main="True Function")
```

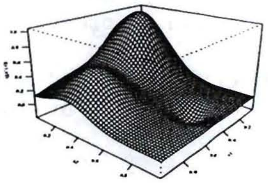

Figure 1: True regression function $m(x_1, x_2)$ (Wendelberger's test function) on the square $[0,1] \times [0,1]$.

Simulate some observation at 100 locations on the regular grid $\{0.05, 0.15, \ldots, 0.85, 0.95\}^2$ with Gaussian disturbances which have zero mean and standard deviation producing a signal to noise ratio of five.

```
R> noise <- .2 ; N <- 13
R> xr <- seq(1/26, 25/26, length=N); yr <- xr ; zr <- outer(xr,yr,f);
R> set.seed(2857)
R> std <- sqrt(0.2*var(as.vector(zr))) ; noise <- rnorm(length(zr),0,std)
R> Z <- zr + matrix(noise,N,N)
```

Transpose the data to a matrix of 2 explanatory variables (of length 100):

```
R> xc <- rep(xr, N) ; yc <- rep(yr, rep(N,N))
R> X <- cbind(xc, yc) ; Zc <- as.vector(Z)
```

With the data `X` and `Zc` we can use the iterated bias reduction smoother with thin plate splines as base smoother S_λ. As the procedure is adaptive, the best choice is to choose for ν_0 the minimum order which is 2, which is the default. As we do not have any idea of the value of λ we can choose a low degree of freedom, for instance 1.2 times the minimum degree of freedom $M_0 = \binom{\nu_0+d-1}{\nu_0-1}$ (which is 3 here):

```
R> res.ibr <- ibr(X,Zc,df=1.2,smoother="tps")
```

The number of iterations is chosen by GCV (the default) and the summary can be obtained using `summary` function:

```
R> summary(res.ibr)
```

The following summary is obtained

```
Residuals:
      Min        1Q    Median        3Q       Max
-0.252993 -0.061420  0.007702  0.065658  0.211733
Residual standard error: 0.1204 on 67.1 degrees of freedom

Initial df: 3.6 ; Final df: 32.89
   gcv
-3.437

Number of iterations: 335 chosen by gcv
Base smoother: Thin plate spline of order 2 (with 3.6 df)
```

giving the residuals standard error, the degree initial freedom (3.6) the final degree of freedom (32.89) and the value of (log) GCV is equal to -3.437 at the chosen number of iterations $\hat{k}_{GCV} = 335$.

We can evaluate the fitted value

```
R> predict(res.ibr)
```

and the Mean Absolute Error on the grid

```
R> mean(abs(predict(res.ibr,grid)-as.vector(zf)))
[1] 0.04908602
```

Obviously we can plot the fitted value on the grid as follows

```
R> persp(xf, yf, zf, theta=130, phi=20, expand=0.45)
```

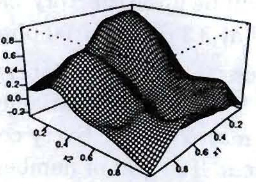

Figure 2: Fitted regression function $\hat{m}_k(x_1, x_2)$ on the square $[0,1] \times [0,1]$, the number of iteration is chosen by GCV: $\hat{k}_{GCV} = 335$.

It can be seen (Figure 2) that the fitted function is not smooth enough compared to the true curve (Figure 1). A smoother function can be obtained by selecting directly "by hand" the number of iterations, which improve the MAE:

```
R> res.ibr2 <- ibr(X,Zc,df=1.2,smoother="tps",iter=250)
R> mean(abs(predict(res.ibr2,grid)-as.vector(zf)))
[1] 0.04714154
```

We can make a comparison to the usual thin plate splines with λ chosen by GCV provided by the function Tps of package **fields**:

```
R> res.tps <- Tps(X, Zc, scale.type="unscaled")
```

and calculates the resulting MAE:

```
R> mean(abs(predict(res.tps,grid)-as.vector(zf)))
[1]  0.05255563
```

On this toy example, the simple use of ibr outperform the usual thin plate splines smoother. Other stopping rules are available and if we use another stopping rules, such as AICc, we simply issue the following command

```
R> res.ibr <- ibr(X,Zc,df=1.2,smoother="tps",criterion="aicc")
```

and we improve our MAE to 0.04530328 with a lower number of iterations ($\hat{k}_{AICc} = 160$). This improvement is consistent with the visual examination of the previous fit of figure 2. This toy example shows the ability of iterated bias reduction smoothing in a small dataset. Let us evaluate the procedure on a real dataset in the next section.

4. Real example: Los Angeles Ozone Data

We consider the classical data set of ozone concentration in the Los Angeles basin which has been previously considered by many authors Breiman (1996); Bühlmann and Yu (2003, 2006). The sample size of the data is $n = 330$ and the number of explanatory variables $d = 8$.

If we want to use thin plate splines here, the order ν_0 have to be greater than $d/2$, that is $\nu_0 = 5$. Thus the minimum degree of freedom of S_λ is $M_0 = 495$ which is greater than n. The thin plate splines smoother is impossible to use here. Recall that the method needs a really smooth base smoother S_λ, with a low degree of freedom compared to n. Thus even if n was 500, the thin plate splines will be unsatisfactory base smoother (recall that in the preceding section, for $d = 2$ we started at 3.3 df with 100 observations).

Let us use the (default) Gaussian kernel smoother. As we do not have any idea of choosing bandwidth for each of the 8 explanatory variables, we fix the degree of freedom of each univariate kernel smoothing matrix at 1.1. Every covariate is implicitly thought as having the same influence and smoothness. The grid of number of bias correction iterations k considered by the model selection procedure for selecting the optimal number of iterations is chosen to be the integer sequence from 1 to 500:

```
R> data(ozone)
R> res.ibr <- ibr(ozone[,-1],ozone[,1],df=1.1,K=1:500)
R> summary(res.ibr)
```

The following summary is obtained

```
Residuals:
      Min       1Q   Median       3Q       Max
-13.5581  -2.0566  -0.3481   1.9816  12.6049
Residual standard error: 3.946 on 309.6 degrees of freedom

Initial df: 2.06 ; Final df: 20.42
  gcv
2.873

Number of iterations: 64 chosen by gcv
Base smoother: gaussian kernel (with 2.06 df)
```

The number of iterations is $\hat{k}_{GCV} = 64$ which can be thought as quite low (recall that in the previous example it was around 150-300). That suggests that we could choose an initial df per variable less than 1.1 (for instance 1.05). In that case, the number of iterations will increase obviously. A little gain of performance is usually expected.

A plot method is also available and gives the index plot of residuals and the evolution of model selection criterion used for choosing k (if sensible).

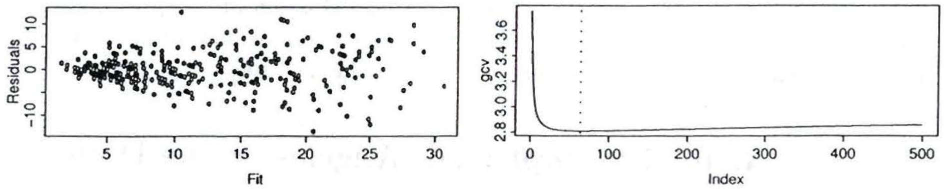

Figure 3: Index plot of residuals and evolution of GCV with k. The vertical dotted line is at the selected $\hat{k}_{GCV} = 64$.

If we want to evaluate the prediction on 50 random splits of 33 observations in test set and 297 in training (see Bühlmann and Yu 2003) use the following commands

```
R> XX <- ozone[,-1]
R> Y <- ozone[,1]
R> erreur1.5 <- rep(0,33*50)
R> aa <- c(1,945095059,162152953)
R> for(i in 1:50){
+    set.seed(aa+i)
+    ind <- sample(1:330,33)
+    XXA <- XX[-ind,]
+    YA <- Y[-ind]
+    XXT <- XX[ind,]
+    YT <- Y[ind]
+    res.ibr <- ibr(XXA,YA,df=1.1,K=1:500)
+    erreur1.5[(33*(i-1)+1):(33*i)] <- YT-predict(res.ibr,XXT)
+ }
R> print(mean(erreur1.5^2))
```

getting an error less than 14.9, which compare favourably with MARS (**mda**), projection pursuit (ppr) or boosting (package **mboost**, (?)) which are around 17 (see Cornillon *et al.* 2009).

5. Conclusion

This new method of smoothing multivariate dataset seems to be promising especially on real dataset. But one limitation of this smoothing method is the use of matrix $n \times n$, where n is the number of observations. Moreover, the longest operation is an eigen decomposition of an $n \times n$ matrix which limits the size of dataset to be used.

References

Breiman L (1996). "Bagging predictors." *Machine Learning*, **24**, 123–140.

Bühlmann P, Yu B (2003). "Boosting with the l_2 loss: Regression and classification." *J. Amer. Statist. Assoc.*, **98**, 324–339.

Bühlmann P, Yu B (2006). "Sparse boosting." *J. Machine Learning Research*, **7**, 1001–1024.

Cornillon PA, Hengartner N, Matzner-Løber E (2009). "Recursive Bias Estimation for high dimensional regression smoothers." *submitted*.

Gu C (2002). *Smoothing spline ANOVA models*. Springer, New-York.

Hastie T, Tibshirani R (1995). *Generalized Additive Models*. Chapman & Hall.

Linton O, Nielsen J (1995). "A kernel method of estimating structured nonparametric regression based on marginal integration." *Biometrika*, **82**, 93–100.

Reinhard Furrer DN, Sain S (2009). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Boulder, USA. URL http://www.image.ucar.edu/Software/Fields.

Utreras F (1988). "Convergence rates for multivariate smoothing spline functions." *Journal of Approximation Theory*, pp. 1–27.

Wendelberger J (1982). "Smoothing Noisy Data with Multivariate Splines and Generalized Cross-Validation." *Ph.D thesis, University of Wisconsin.*

Affiliation:

Pierre-André Cornillon
UMR ASB, SupAgro INRA
2, Place P. Viala
34060 Montpellier Cedex, France
E-mail: pierre-andre.cornillon@supagro.inra.fr
URL: http://www1.montpellier.inra.fr/umr_asb/umr.php?page=cornillon

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd