ibr: Iterative Bias Reduction

N.Jégou (nicolas.jegou@uhb.fr)

UseR! 2009

Non parametric regression model

Having n observations $(X_i, Y_i) \in \mathbb{R}^d \times \mathbb{R}$ from the model

$$Y_i = m(X_i) + \epsilon_i, \qquad i = 1 \cdots n$$

where $m(.)$ is an unknown smooth function.

Smoothing

We estimate m non-parametrically (smoothing) :

$$\hat{m} = S_\lambda Y$$

where S_λ is the smoothing matrix and λ is the smoothing parameter (size of the bin, bandwidth, penalty...).

Big picture

Assume λ big, so that the smoother is very smooth ; then

- estimate the bias
- correct the previous smoother

and iterate.

Bias of a linear smoother

We choose a smooth pilot S_λ. The estimation is

$$\hat{m}_1 = S_\lambda Y.$$

The bias is

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

Estimating the bias

We replace m by its estimation :

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

This can be written

$$\hat{b}_1 = -S_\lambda(I - S_\lambda)Y = -S_\lambda R$$

where R denotes the observed residuals.

Bias correction

The corrected estimator is

$$\begin{aligned}\hat{m}_2 &= \hat{m}_1 - \hat{b}_1 \\ &= (S_\lambda + S_\lambda(I - S_\lambda))Y.\end{aligned}$$

After k iterations, the estimator is

$$\hat{m}_k = [S_\lambda + S_\lambda(I - S_\lambda) + \cdots + S_\lambda(I - S_\lambda)\cdots(I - S_\lambda)]Y$$

and can be written

$$\hat{m}_k = [I - (I - S_\lambda)^k]Y.$$

Library IBR

Input parameters :

- Choose the smoother : "tps" or "**k**".
- Choose the criterion : "**gcv**","aic","aicc","bic","rmse","map".
- Choose the pilot's smoothing parameter : df.

```
> res<-ibr(X,Y,smoother="tps",df=1.1,criterion="aic")
```

Which smoother should I use ?
"tps" vs "k"

- smoother="tps"

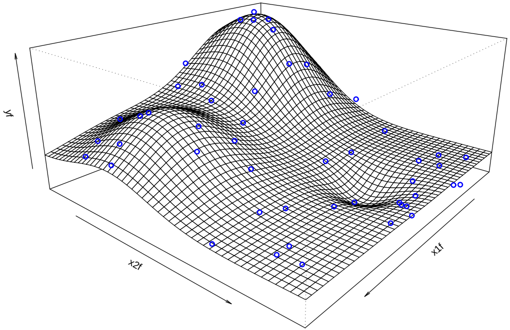
$$\nu_0 = \lfloor d/2 \rfloor + 1 \qquad \#\{e.v = 1\} = \binom{\nu_0 + d - 1}{\nu_0 - 1}$$

d	ν_0	$\#\{e.v = 1\}$
2	2	3
5	3	21
8	5	495

- so we use smoother="k" in high dimensions

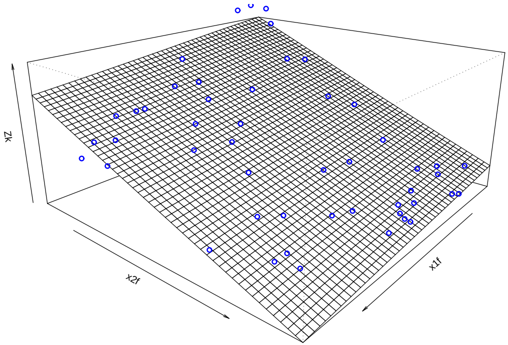
A toy example

True function :



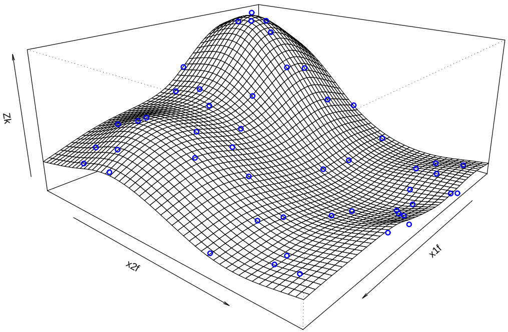
Pilot smoother

```
res.tps<-ibr(X,Y,smoother="tps",df=1.1,iter=1)
Zk<-matrix(predict(res.tps,grid),ncol=ngrid)
res<-persp(x1f,x2f,Zk,...)
```



Selected smoother by ibr

```
> res.tps<-ibr(X,Y,smoother="tps",df=1.1)
> Zk<-matrix(predict(res.tps,grid),ncol=ngrid)
> res<-persp(x1f,x2f,Zk,...)
```



Summary

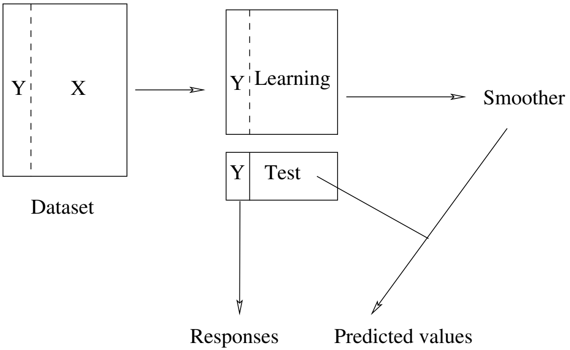
```
> summary(res.tps)

Residuals:
      Min         1Q     Median         3Q        Max
-0.094546 -0.022644 -0.002132  0.023794  0.093353
Residual standard error: 0.05373 on 26.1 degrees of freedom

Initial df: 3.3 ; Final df: 23.86
   gcv
-4.551

Number of iterations: 482 chosen by gcv
Base smoother: Thin plate spline of order 2 (with 3.3 df)
```

Simulation study

We will compare ibr with :

- Additive models (Hastie & Tibshirani, 1995),
- MARS (Friedman, 1991),
- Projection pursuit (Breiman & Friedman, 1985),
- L2-Boosting (Bühlmann & Yu, 2003).

Data splitting



Simulation results

$$f(x) = a\sin(\pi x_1 x_2 x_3) + b(x_4 - 0.5)^2 + c x_5$$

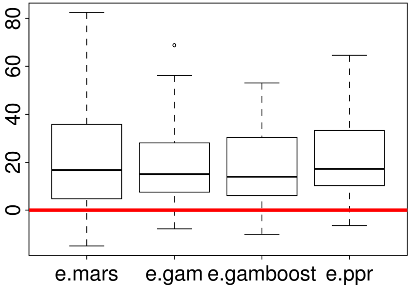
$(a, b, c) =$	IBR (k,df=1.1)	GAM	MARS	PPR	gamboost
(0.2, 10, 5)	29%	0.14	21%	379%	7%
(1, 10, 5)	11%	5%	11%	139%	0.18
(10, 4, 5)	1.38	301%	117%	53%	111%
(10, 1, 1)	1.19	350%	134%	21%	124%

A real example : Los Angeles Ozone Data

The sample size is $n = 330$ and $d = 8$ explanatory variables :

```
"Pressure.Vand"  "Wind"  "Humidity"
"Temp.Sand"  "Inv.Base.height" "Pressure.Grad"
"Inv.Base.Temp" "Visilibity"
```

Comparing relative prediction mean square errors

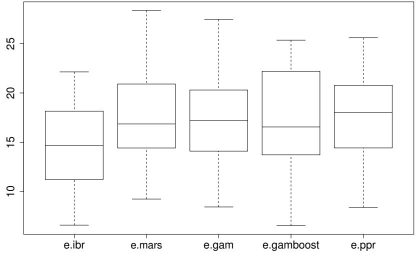


Download

Google :

- ibr + matzner
- ibr + cornillon

Comparing spread



Predictive smoothers

Recall

$$\hat{m}_k = [S_\lambda + S_\lambda(I - S_\lambda) + \cdots + S_\lambda(I - S_\lambda)\cdots(I - S_\lambda)]Y$$

which can be written :

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

At an arbitrary location $x \in \mathbb{R}^d$

$$\hat{m}_k(x) = S(x)^t \hat{\beta}_k.$$