Functional gradient descent boosting for additive non-linear spatial autoregressive model (gaussian and probit)

Ghislain Geniaux

Functional gradient descent boosting for additive non-linear spatial autoregressive model (gaussian and probit)

Ghislain Geniaux, INRAE Ecodéveloppement UR767

ghislain.geniaux@inrae.fr

OUTLINE

1. Semi-parametric autoregressive models (SAR, SDM, SEM, and SARAR) formulated as additive models with smoothing spline or tree-based learners.
2. The SAR Boosting Estimators in *spboost* R package (methods and Monte Carlo Results for gaussian case)
3. Non linear SAR boosting with Multiple spatial weight matrices.
4. Non linear SAR probit boosting using Mendell-Eston approximation from Martinetti and Geniaux 2017

Spatial autogressive non-linear model: a semi-parametri additive estimator

We are considering the following spatial autoregressives models:

$$Y = \rho WY + \sum_{j=1}^{p} h_j(X_j) + \epsilon \;\; (SAR)$$

$$Y = \sum_{j=1}^{p} h_j(X_j) + (I - \lambda M)^{-1}\epsilon \;\; (SEM)$$

$$Y = \rho WY + \sum_{j=1}^{p} h_j(X_j) + (I - \lambda M)^{-1}\epsilon \;\; (SARAR)$$

The estimation was decoupled into two steps or levels of estimation:

1. Nonlinear relationships were handled using Functional Gradient Boosting.
2. The spatial dependence parameter(s) were estimated through:
   i. QML Basile and Gress (2004) and Su and Jin (2010).
   ii. CFE (Smirnov 2020) adapted for nonlinearity.
   iii. Instrumental methods (Mara and Radice 2010, Basile et al. 2014)

I will briefly revisit the foundations of this 'gaussian' version in both ML and CFE.

We can start from the non-linear SAR model:

$$Y = \rho WY + f(X) + \epsilon = \rho WY + \sum_{j=1}^{p} h_j(X_j) + \epsilon \quad (SAR)$$

The spatially filtered model with known ρ is written with $\tilde{Y} = Y - \rho WY$:

$$\tilde{Y} = \sum_{j=1}^{p} h_j(X_j) + \epsilon$$

the functions $h_j()$ can be estimated directly with the existing machine learning libraries (for example mboost, xgboost) and get the residuals :

$$\hat{\epsilon}(\hat{h}(\rho))$$

A first very simple solution to estimate ρ consists indeed in numerically minimizing the concentrated loglik:

$$argmin_\rho \;\; l_c(\rho, \hat{\epsilon}(\hat{h}(\rho)))$$

with

$$l_c(\rho) = C + |I - \rho W| + ln(\frac{\hat{\epsilon}(\hat{h}(\rho))' \hat{\epsilon}(\hat{h}(\rho))}{n})$$

Where $C = -(n/2)ln(2\pi) - (n/2)$.

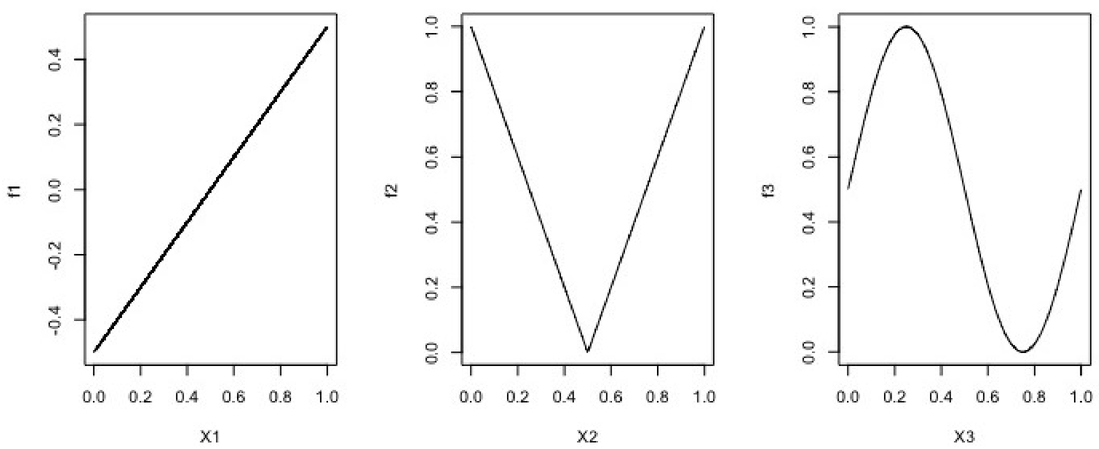
A second more elegant solution consists in redefining the gradients and the loss functions of the boosting algorithms:

Table 1
Gradients, Loss functions and concentrated likelihood

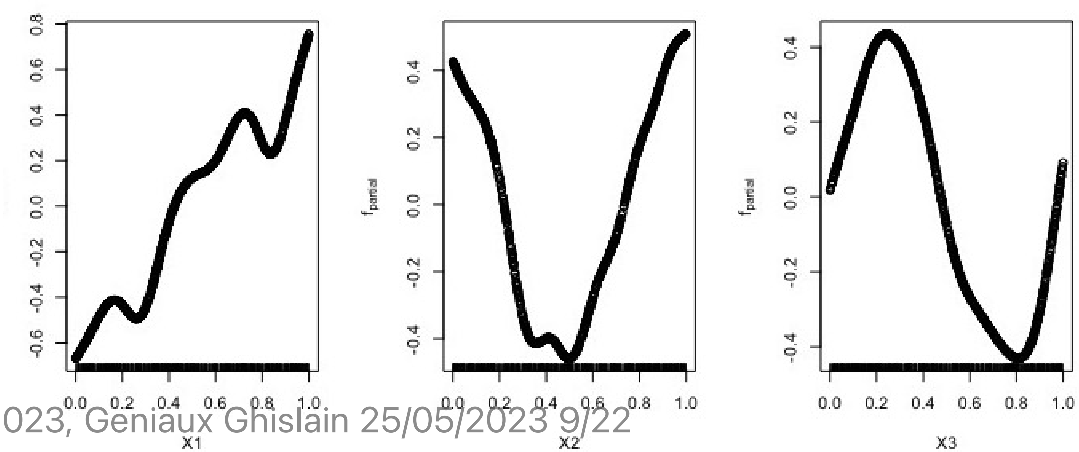
	Aspatial	SAR	SEM	SARAR
Gradient	$(y-f)$	$(y-\rho Wy-f)$	$(I-\lambda W)(y-f)$	$(I-\lambda W)(y-\rho Wy-f)$
Loss Function	$\sum(y-f)^2$	$\sum(y-\rho Wy-f)^2$	$\sum(((I-\lambda W)(y-f))^2)$	$\sum(((I-\lambda W)(y-\rho Wy-f))^2)$

Simulation of a non linear SAR DGP :

- with $\rho = 0.86$
- n=1000
- random location
- W based on the first 4 neighbors
- X=(X1,X2,X3,X4,X5) random normal
- with only f(X1) f(X2) and f(X3) contributing to the DGP.



```
library(spboost)
model1=spgbam(formula=as.formula('Y~X1+X2+X3+X4+X5'),data=mydat
             W=W,DGP='SAR',method='BSPA_SAR_ML')
model1$rho
# 0.8436531
summary(model1)
# Model-based Boosting Call: gamboost(formula=Y~X1+X2+X3+X4+X5,
# Squared Error (Regression) Loss function: (y - rho Wy - f)^2
# Number of boosting iterations: mstop = 1000
# Step size:  0.3
# Offset:  1.141838
# Number of baselearners:  4
# Selection frequencies:
# s(X1)  s(X2)  s(X3)  s(X5)
# 0.443  0.408  0.146  0.003
plot(model1)
```



```
# BSPA_SAR_CFE method

model2=spbgam(formula=as.formula('Y~X1+X2+X3+X4+X5'),data=mydat
             W=W,DGP='SAR',method='BSPA_SAR_CFE')
model2$rho # 0.8436531

summary(model2)
# Model-based Boosting Call: gamboost(formula=Y~X1+X2+X3+X4+X5,
# Squared Error (Regression) Loss function: (y - rho Wy - f)^2

# Number of boosting iterations: mstop = 1000
# Step size:  0.3
# Offset:  1.117332
# Number of baselearners:  4
# Selection frequencies:
# s(X1)  s(X2)  s(X3)  s(X5)
  0.440  0.409  0.147  0.004
```

Monte Carlo results

Result for experiments E1, nonlinear SAR, SNR=0.9 n=2000

		gamboost		gam		lagsarlm		BSPA_SAR_ML		BSPA_SAR_CFE		GAM_SAR_ML		GAM_SAR_CFE	
ρ	terms	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE	Bias	RMSE
0.0	$\hat{\rho}$					-0.0020	*0.0306*	-0.0021	*0.0276*	-0.0021	*0.0276*	-0.0022	*0.0276*	-0.0021	*0.0276*
	$\hat{f}_1$		*0.0172*		*0.0142*		*0.5355*		*0.0172*		*0.0172*		*0.0142*		*0.0141*
	$\hat{f}_2$		*0.0459*		*0.0343*		*0.2885*		*0.0459*		*0.0459*		*0.0343*		*0.0343*
	$\hat{f}_3$		*0.0391*		*0.0202*		*0.5384*		*0.0391*		*0.0391*		*0.0202*		*0.0202*
0.2	$\hat{\rho}$					-0.0022	*0.0275*	-0.0022	*0.0253*	-0.0020	*0.0255*	-0.0023	*0.0253*	-0.0020	*0.0255*
	$\hat{f}_1$		*0.0164*		*0.0182*		*0.5376*		*0.0174*		*0.0174*		*0.0154*		*0.0154*
	$\hat{f}_2$		*0.0460*		*0.0371*		*0.2888*		*0.0466*		*0.0466*		*0.0366*		*0.0366*
	$\hat{f}_3$		*0.0379*		*0.0209*		*0.5385*		*0.0393*		*0.0393*		*0.0215*		*0.0216*
0.6	$\hat{\rho}$					-0.0019	*0.0177*	-0.0019	*0.0169*	-0.0016	*0.0180*	-0.0020	*0.0169*	-0.0016	*0.0180*
	$\hat{f}_1$		*0.0375*		*0.0605*		*0.5448*		*0.0187*		*0.0187*		*0.0193*		*0.0193*
	$\hat{f}_2$		*0.0533*		*0.0615*		*0.2896*		*0.0494*		*0.0495*		*0.0438*		*0.0438*
	$\hat{f}_3$		*0.0443*		*0.0444*		*0.5390*		*0.0399*		*0.0399*		*0.0258*		*0.0259*
0.9	$\hat{\rho}$					-0.0009	*0.0060*	-0.0008	*0.0059*	-0.0007	*0.0076*	-0.0009	*0.0059*	-0.0007	*0.0076*
	$\hat{f}_1$		*0.2243*		*0.2621*		*0.5549*		*0.0226*		*0.0226*		*0.0249*		*0.0250*
	$\hat{f}_2$		*0.2061*		*0.2188*		*0.2912*		*0.0542*		*0.0542*		*0.0532*		*0.0533*
	$\hat{f}_3$		*0.2319*		*0.2453*		*0.5396*		*0.0421*		*0.0421*		*0.0316*		*0.0317*

To resume *spboost* package allows to :

- Estimate SAR, SEM and SARAR models for Gaussian case (continuous Y) using boosting algorithm
- Consider h(X) as spline functions (mboost, mgcv/gam)
- Consider h(X) as decision trees (xgboost)
- Plot, summary, predict methods for spbgam models,
- Predict uses a spatial BLUP estimator for spatial autoregressive model (Goulard et al. 2017)
- Identify a suitable matrix W from data (linear combination of W candidates).
- Estimate SAR model for probit case (binomial Y) using boosting algorithm

Identify a suitable matrix W from data

We consider a set of candidate spatial weight matrices $WW = (W_1, W_2, W_3, ..., W_q)$ all rownormalized and start form estimating the following model using QML approach :

$$Y = \sum_{l \in q'} \rho_l W_l Y + f(X) + \epsilon$$

The estimation of the ρ_l and the choice of q' is done sequentially.

Example:

Suppose that W_1 is the best candidate for a "single" SAR model :

$$Y = \rho_1 W_1 Y + f(X) + \epsilon \quad (1)$$

We estimate $\hat{\rho}_1$ and consider a second model :

$$Y - \hat{\rho}_1 W_1 Y = \rho_{j*} W_{j*} Y + f(X) + \epsilon \quad (2)$$

And we look for $j^* \neq 1 \in q$ minimizing the loglik of (2).

If $j^* = 3$, then we estimate $\hat{\rho}_3 = \hat{\rho}_3(\hat{\rho}_1)$ and introduce a third model to reestimate $\rho_1(\hat{\rho}_3)$:

$$Y - \hat{\rho}_3 W_3 Y = \rho_1(\hat{\rho}_3) W_j Y + f(X) + \epsilon \quad (3)$$

we repeat reestimation of (2) and (3) until the values of $\hat{\rho}_1$ and $\hat{\rho}_3$ converge.

Once $\hat{\rho}_1$ and $\hat{\rho}_3$ have converged, we consider the introduction of a third candidate matrix $j^* \neq (1,3) \in q$ using:

$$Y - \hat{\rho}_1 W_1 Y - \hat{\rho}_3 W_3 Y = \rho_{j*} W_{j*} Y + f(X) + \epsilon \quad (4)$$

And so onâ€¦

We repeat this procedure as long as the Akaike criterion is improved.

```
model3=spbgam(formula=as.formula('Y~X1+X2+X3+X4+X5'),data=mydat
             W=WW,DGP='SARW',method='BSPA_SAR_ML')
## W=WW is a list of weight matrice candidates and DGP='SARW' s
```

Monte Carlo results

DGP: Same design +

- 3 different W_l based on the 2, 4 and 8 nearest neighbors
- $\rho_l \in (0.25, 0.15, 0.1, 0, -0.25)$
- linear SAR with boosting (BLA_SAR_ML).

1000 replications, $\beta = (0.5, 1, -2, 1)$

n	True ρ_j			Mean Bias ρ_j			Mean Bias β_k			Mean Bias β_k direct			Mean Bias β_k indirect		
1000	0.10	-0.25	0.15	-0.0129	-0.0980	0.1208	-0.0063	0.0030	0e+00	-0.0062	0.0077	0.0030	-0.0156	0.0001	0.0078
1000	0.00	0.25	0.15	-0.0027	-0.0143	0.0258	-0.0076	0.0031	-1e-04	-0.0070	0.0138	0.0015	-0.0350	0.0007	0.0184
1000	0.10	0.25	0.15	-0.0009	-0.0017	0.0101	-0.0084	0.0036	0e+00	-0.0077	0.0143	0.0016	-0.0410	0.0010	0.0220
1000	0.25	0.25	0.15	-0.0034	-0.0126	0.0224	-0.0102	0.0047	-3e-04	-0.0090	0.0210	0.0008	-0.0681	0.0018	0.0373
1000	0.40	0.25	0.15	-0.0023	-0.0105	0.0173	-0.0129	0.0060	-5e-04	-0.0118	0.0322	-0.0012	-0.1344	0.0037	0.0760
2000	0.10	-0.25	0.15	0.0005	-0.0130	0.0299	0.0057	0.0021	3e-04	0.0055	0.0160	0.0026	-0.0321	0.0001	0.0161
2000	0.00	0.25	0.15	-0.0001	0.0015	0.0058	0.0068	0.0030	3e-04	0.0072	0.0188	0.0025	-0.0267	0.0006	0.0148
2000	0.10	0.25	0.15	0.0009	-0.0020	0.0083	0.0074	0.0035	3e-04	0.0082	0.0278	0.0025	-0.0378	0.0008	0.0214
2000	0.25	0.25	0.15	0.0010	0.0004	0.0036	0.0086	0.0045	2e-04	0.0106	0.0419	0.0026	-0.0456	0.0014	0.0276
2000	0.40	0.25	0.15	0.0016	-0.0003	0.0021	0.0108	0.0062	0e+00	0.0160	0.0935	0.0025	-0.0844	0.0026	0.0552
4000	0.10	-0.25	0.15	0.0002	0.0018	0.0000	0.0008	-0.0029	0e+00	0.0006	0.0006	-0.0026	-0.0009	-0.0002	0.0005
4000	0.00	0.25	0.15	0.0002	0.0022	-0.0010	0.0008	-0.0034	-2e-04	0.0008	0.0001	-0.0034	-0.0006	-0.0002	-0.0008
4000	0.10	0.25	0.15	0.0004	0.0019	-0.0004	0.0009	-0.0037	-2e-04	0.0010	-0.0010	-0.0039	0.0009	-0.0002	-0.0022
4000	0.25	0.25	0.15	0.0008	0.0019	0.0039	0.0010	-0.0044	-3e-04	0.0014	-0.0140	-0.0052	0.0253	0.0000	-0.0164
4000	0.40	0.25	0.15	0.0019	0.0040	0.0208	0.0017	-0.0062	1e-04	0.0023	-0.0972	-0.0080	0.1872	0.0004	-0.1022

Estimate SAR model for probit case (binomial Y) using boosting algorithm

Regular non linear PROBIT

$$\hat{l}\left(f\right)=\frac{1}{n}\sum_{i=1}^{n}\left[Y_i\log\Phi\left(f\left(\mathbf{x}_i\right)\right)+\left(1-Y_i\right)\log\Phi\left(-f\left(\mathbf{x}_i\right)\right)\right]$$

The gradient is calculated as:

$$D(f)=\mathbf{E}\left[\frac{\varphi(f)(Y-\Phi(f))}{\Phi(f)\Phi(-f)}\mid\mathbf{x}\right]$$

Non linear SAR PROBIT:

$$y = (I - \rho W)^{-1}(f(x) + \epsilon)$$

The probability of $Y = (y > 0)$ given x has the form:

$$P(Y = 1 \mid \mathbf{x}) = \Phi(\frac{(I - \rho W)^{-1}}{v} f(\mathbf{x})),$$

where

$$v = \left(\sqrt{(I - \rho W)^{-1}(I - \rho W)^{-1}}\right)_{ii}$$

The gradient and the loss function for boosting can be formulated as:

$$D(f) = \frac{(I-\rho W)^{-1}}{v} \frac{\varphi(\frac{(I-\rho W)^{-1}}{v} f)(Y - \Phi(\frac{(I-\rho W)^{-1}}{v} f))}{\Phi(\frac{(I-\rho W)^{-1}}{v} f)\Phi(-\frac{(I-\rho W)^{-1}}{v} f)},$$

$$loss(f) = Y \log \Phi(\frac{(I-\rho W)^{-1}}{v} f(\mathbf{x}))$$

$$+(1-Y) \log \Phi(-\frac{(I-\rho W)^{-1}}{v} f(\mathbf{x})),$$

Again, we separate the estimation of ρ and $f(x)$ with the particularity here of using a Mendell-Elston ML approximation method for probit model to estimate ρ as proposed by Martinetti and Geniaux (2017); see also SpatialProbit R package.

Does-it works ?

Several estimation modes were tested for the boosting/nonlinear part of the probit model:

- Direct approaches using the previous gradient adpated to non linear SAR probit with gamboost or penalized spline (RMLE) for f and SpatialProbit library (ML approximation for ρ).
- Indirect approaches in which we first construct the decomposition of the k cubic splines into a matrix of size $n \times (k + s(k))$ which can be fed in linear form into the spatial probit mode estimation with the SpatialProbit R function.

Surprisingly simulation results show that the indirect approach gives better results in general for estimating ρ, even in the case where confusing explanatory variables are present,

And the direct boosting approach using ρ estimated with the indirect approach provides better estimation of non linear relationship $f(X)$

Perspectives

In progress :

- Towards a spatial variability of the autoregressive structure.
- Assess spatial cross validation strategy (regular, spatial block, environmental block)

Longer term:

- Improve compatibility with other Machine Learning algorithms.

Grants: SEPIM, BEYOND, URBANSIMUL.