Utilisation d'une approche par gradient boosting pour l'estimation de modèles autorégressifs spatiaux non linéaires. Application à des données de surveillance de la jaunisse

Ghislain Geniaux

Utilisation d'une approche par gradient boosting pour l'estimation de modèles autorégressifs spatiaux non linéaires. Application à des données de surveillance de la jaunisse

Ghislain Geniaux, INRAE Ecodéveloppement UR767

PACKAGE R SPBOOST

Ce package vise à permettre d'estimer des modèles autoregressifs spatiaux de type (SAR, SDM, SEM and SARAR) avec des relations non linéaires avec différents estimateurs s'appuyant sur des algorithmes de boosting (Friedman, 2001; Bühlmann et al., 2007).

2 autres méthodes sont aussi proposé pour accéler les estimations en présence de gros échatillons :

- Closed Form Estimator (CFE, Smirnov 2020) pour les modèles SAR et SEM (CFE, Smirnov 2020),
- Flexible Instrumental Variable Approach (FIVA, Marra and Radice 2010) pour les modèles SAR.

Modèles autoregressifs spatiaux non-linéaires

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

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

Modèles autoregressifs spatiaux non-linéaires

$$Y_{it} = \rho \sum_{j} w_{j,s<t} Y_{j,s<t} + \sum_{j=1}^{p} h_j(X_j) + \epsilon \quad (SAR)$$

$$Y = (I - rhoW)^{-1} \sum_{j=1}^{p} h_j(X_j) + \epsilon \quad (SAR)$$

Modèles autoregressifs spatiaux non-linéaires

La logvraissemablance concentré de ce type de modèle s'écrit:

$$lnL(\rho) = C + |I - \rho W| \tag{1}$$

$$- \frac{n}{2} ln\Big(\frac{(Y - \rho WY - \sum h_j(X_j))'(Y - \rho WY - \sum h_j(X_j))}{n} \Big)$$

où $C = -(n/2)ln(2\pi) - (n/2)$.

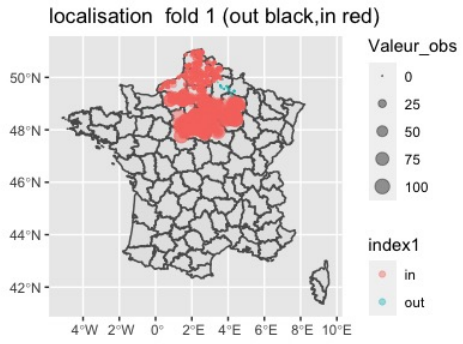
Fonctionalités de spboost

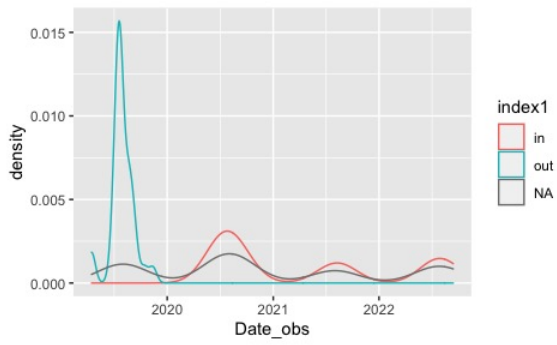
spboost permet d'estimer des modèles SAR, SEM et SARAR :

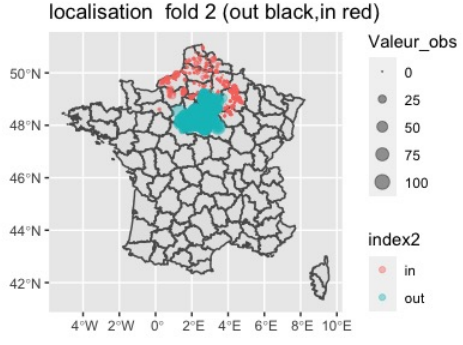
- cas gaussien (Y continu)
- cas probit (Y binomial)
- h(X) ~ fonctions splines (mboost, mgcv/gam)
- h(X) ~ arbres de décisions (xgboost)
- d'identifier la matrice optimale W (in progress)
- Predictions s'appuients sur un estimateur BLUP spatial (Goulard et al. 2017)

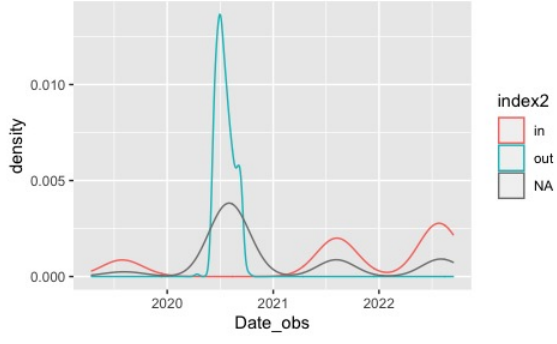
Application aux données *jaunisse*

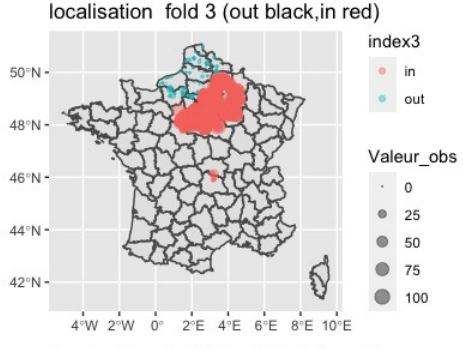
- Stratégie de cross validation : plus de temporel moins de spatial dans l'échatillonage. (reset annuel ?)
- Prise en compte de l'information de début d'année
- horizon de prédiction.

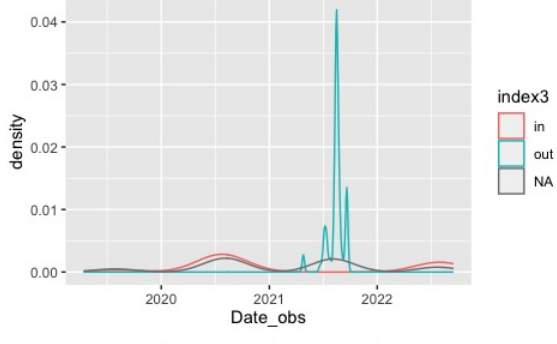

localisation fold 1 (out black,in red)
Valeur_obs
0
25
50
75
100
index1
in
out
50°N
48°N
46°N
44°N
42°N
4°W 2°W 0° 2°E 4°E 6°E 8°E 10°E

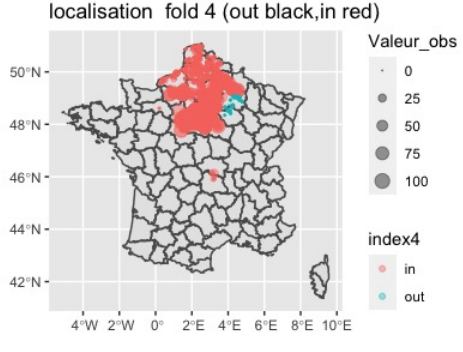

density
0.015
0.010
0.005
0.000
2020
2021
2022
Date_obs
index1
in
out
NA


localisation fold 2 (out black,in red)
Valeur_obs
0
25
50
75
100
index2
in
out
50°N
48°N
46°N
44°N
42°N
4°W 2°W 0° 2°E 4°E 6°E 8°E 10°E


density
0.010
0.005
0.000
2020
2021
2022
Date_obs
index2
in
out
NA

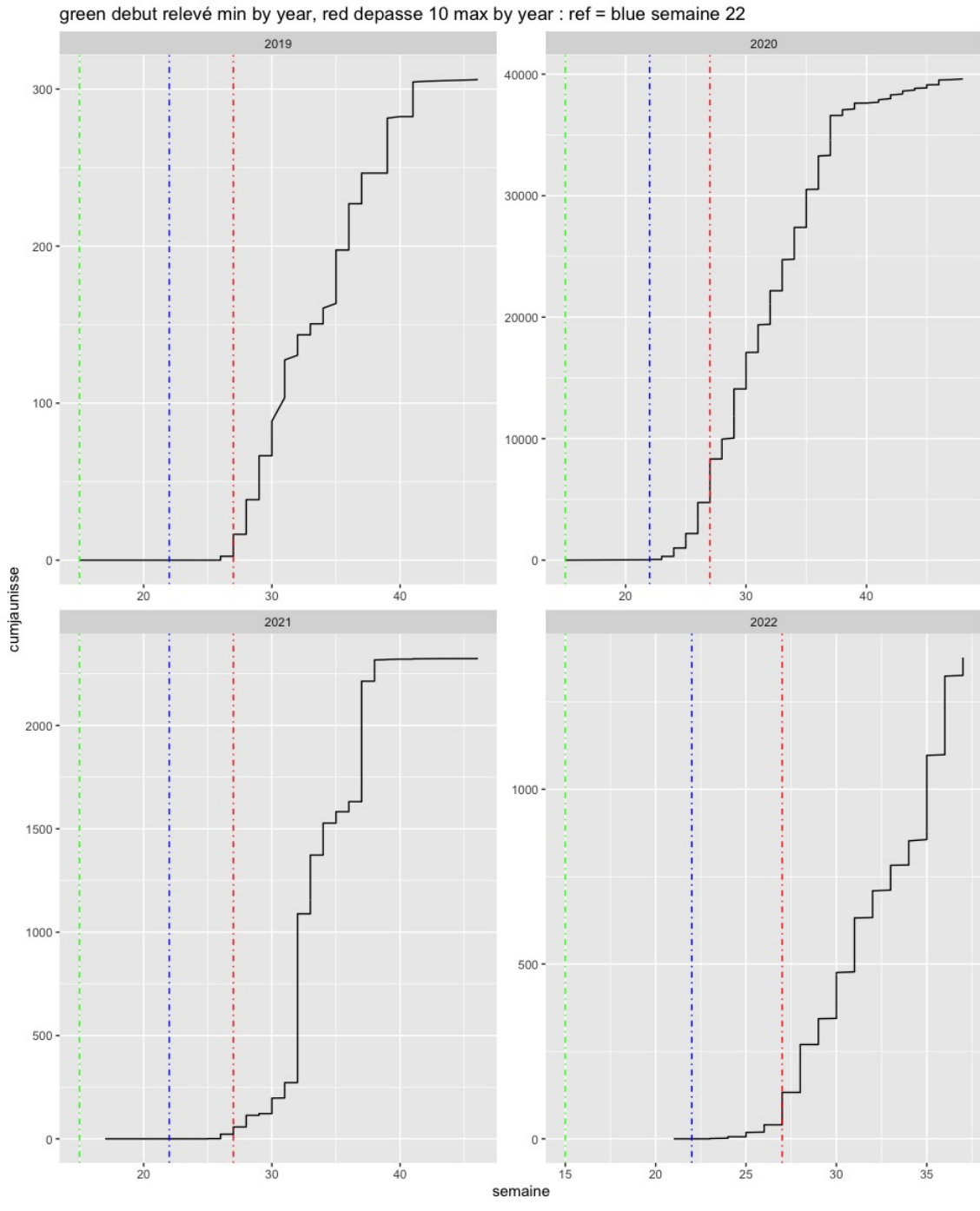

localisation fold 3 (out black,in red)
index3
in
out
Valeur_obs
0
25
50
75
100
50°N
48°N
46°N
44°N
42°N
4°W 2°W 0° 2°E 4°E 6°E 8°E 10°E


density
0.04
0.03
0.02
0.01
0.00
2020
2021
2022
Date_obs
index3
in
out
NA

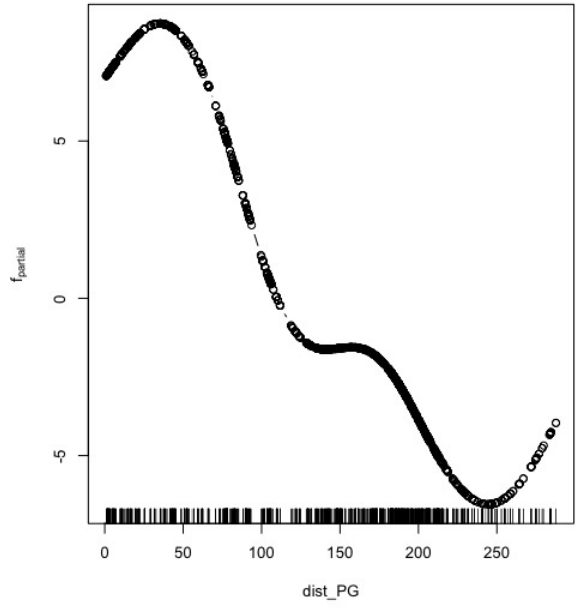

localisation fold 4 (out black,in red)
Valeur_obs
0
25
50
75
100
index4
in
out
50°N
48°N
46°N
44°N
42°N
4°W 2°W 0° 2°E 4°E 6°E 8°E 10°E

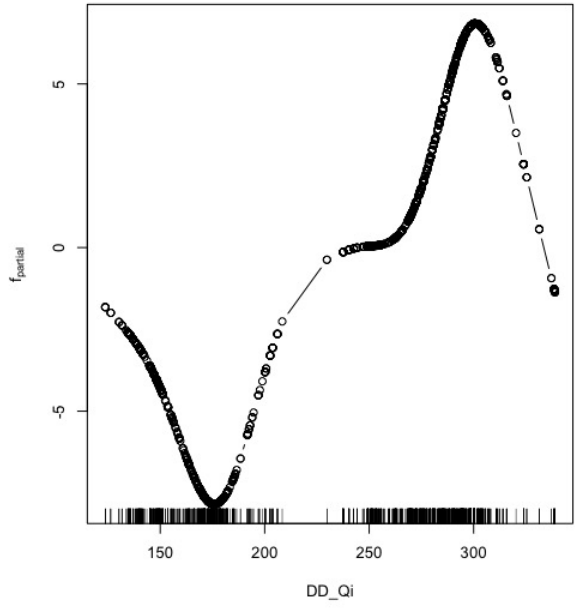

density
0.010
0.005
0.000
2020
2021
2022
Date_obs
index4
in
out
NA

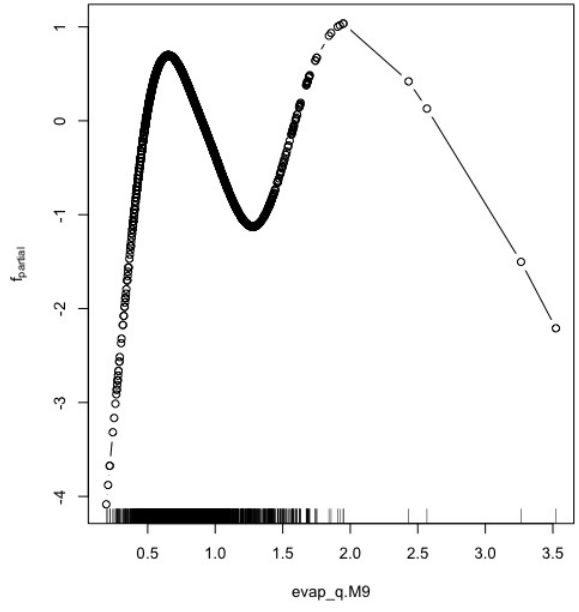
```
> apply(dist_train,2,function(x) quantile(x,c(0.2,0.8)))
    [,1] [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]    [,9]    [,10]
20%    0    0  0.00000    0.000    0.000    0.000    0.000    0.000    0.00    0.000
80%    0    0 51.00253 1063.318 2536.319 3549.165 4798.226 6430.166 7448.54 8395.293
> apply(dist_test_train,2,function(x) quantile(x,c(0.2,0.8)))
         [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]      [,8]      [,9]     [,10]
20%  392.1433  400.7523  414.8903  458.1708  544.3642  578.9649  634.1451  786.2675  786.2675  832.5567
80% 2848.7532 3581.4461 4050.9302 4493.7231 4559.7512 4745.8794 6833.6161 7573.0305 8753.5954 9728.5111
```

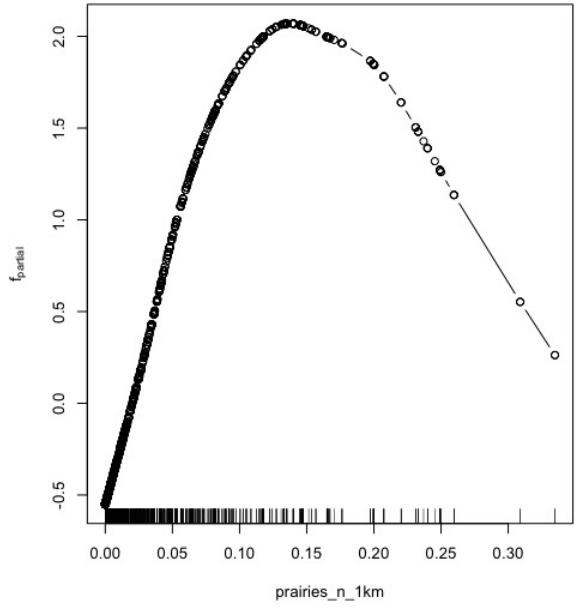

green debut relevé min by year, red depasse 10 max by year : ref = blue semaine 22
2019
2020
2021
2022
cumjaunisse
semaine

```
s1<-sampling_st(jj,2021,22,36,0)
s2<-sampling_st(jj,2021,22,36,1)
s3<-sampling_st(jj,2021,22,36,2)
s4<-sampling_st(jj,2021,22,36,4)

W1km<-matWpast(coords,TI,k=100,radius=100000
,H=1000,kernels='gauss')
WT5<-matTIME(TI,H=5,kernels='gauss')
## + croisement

model_spboost=spgam(formula=myformula,data=train,W=W_tr
DGP='SAR',method='gamboost_ML',
control=list(control_gamboost=boost_control(mstop=1000,

pred<-predict_spboost(model_spboost, test,train,
W = W_TRAIN_TEST, type = "BPN")
pred[pred<0]<-0
```


f_partial
5
0
-5
0
50
100
150
200
250
dist_PG


f_partial
5
0
-5
150
200
250
300
DD_Qi


f_partial
1
0
-1
-2
-3
-4
0.5
1.0
1.5
2.0
2.5
3.0
3.5
evap_q.M9


f_partial
2.0
1.5
1.0
0.5
0.0
-0.5
0.00
0.05
0.10
0.15
0.20
0.25
0.30
prairies_n_1km

Comparaisons : rf, xgboost et spboost sur 2020 2021 et 2022 avec depart du test en semaine 15,20, 22, 24, 26, 28 avec 0,1,2,4 semaine de delais.

- si semaine >=22 spboost > rf > xgboost et particulièrement si la l'horizon de prediction est plus lointain.
- 10 à 20 % d'amélioration des RMSE (max en semaine 30)
- proportional logit/ dirilichet
- W optimal ?