Speeding up estimation of spatially varying coefficients models.

Ghislain Geniaux

Speeding up estimation of spatially varying coefficients models

Ghislain Geniaux, INRAE Ecodéveloppement UR 767

Speeding up estimation of spatially varying coefficients models

Extensions of **Geniaux and Martinetti (2018)** "A new method for dealing simultaneously with spatial autocorrelation and spatial heterogeneity in regression models", *RSUE*, vol. 72.

CRAN R package mgwrsar: GWR and MGWR with Spatial depedances

1. Overview of spatially varying coefficients DGPs with spatial dependence,
2. Estimation methods and functionnalities of mgwrsar (0.1)
3. how to reduce the estimation time ?
    1. Methods
    2. Monte Carlo experiments
    3. Real estate data example.

1. Overview of spatially varying coefficients DGPs with spatial dependence

Varying coefficient models in which the linear parameters varying with respect to spatial coordinates, generally named GWR, introduced by Brundson and McMillen (Brunsdon et al., 1996; McMillen, 1996, Cleveland and Devlin, 1988).

$$y_i = \sum_{j=1}^{J} \beta_j(u_i, v_i) x_{ij} + \epsilon_i \quad (GWR)$$

$$y_i = \sum_{k=1}^{K} \beta_k X_k + \sum_{j=1}^{J} \beta_j(u_i, v_i) x_{ij} + \epsilon_i \quad (MGWR)$$

2. Estimation methods and functionnalities of mgwrsar (0.1)

Geniaux and Martinetti (2018) proposes estimation methods for a variety of varying coefficients DGP with spatial dependance :

$$y = \lambda W y + \beta_c X_c + \epsilon_i \qquad (MGWR - SAR(0, k, 0))$$

$$y = \lambda W y + \beta_v(u_i, v_i) X_v + \epsilon_i \qquad (MGWR - SAR(0, 0, k))$$

$$y = \lambda W y + \beta_c X_c + \beta_v(u_i, v_i) X_v + \epsilon_i \qquad (MGWR - SAR(0, k_c, k_v))$$

$$y = \lambda(u_i, v_i) W y + \beta_c X_c + \epsilon_i \qquad (MGWR - SAR(1, k, 0))$$

$$y = \lambda(u_i, v_i) W y + \beta_v(u_i, v_i) X_v + \epsilon_i \qquad (MGWR - SAR(1, 0, k))$$

$$y = \lambda(u_i, v_i) W y + \beta_c X_c + \beta_v(u_i, v_i) X_v + \epsilon_i \qquad (MGWR - SAR(1, k_c, k_v))$$

2. Estimation methods and functionnalities of mgwrsar (0.1)

mgwrsar 0.1 packages allows :

- Estimations with MGWR-SAR like models with **2SLS** and Best 2SLS methods (Kelejian and Prucha 1999),
- to **test** model specifications and spatial stationarity of coefficients using bootstrap
- to deal with **local outliers**,
- Optimization of **bandwidth choice** using cross validation with various kernels,
- Prediction of spatial effects using **BLUP** (Thomas-Agnan et al. 2013)
- to use **General kernel Product functions** (Li and Racine 2010) to increase the dimensions that define the local sample (space + time + other non linear covariate).

3. How to reduce the estimation time ? (mgwrsar 1.0).

Various avenues have been explored in the litterature:

1. Optimization of routines for calculating local regression and spatial weight matrix (lower level language, sparse weight matrix),
2. Avoiding to store the hat matrix,
3. Parallelization of local models and distance calculations,
4. Using adaptive kernels based on K-nearest neighbors,
5. Improvement of bandwidth selection procedures,
6. Using a subset of local models, i.e. reduce the number of target points.

GWR with target points and rough gaussian kernel

Rough gaussian kernel:

Grzesik (2017) shows that the Relative Efficiency of the gaussian kernel, compared to Epanechnikov Kernel, can be optimized by using a truncated gaussian in which the truncation threshold depends on the sample size.

We show the improvement of the rough guassian kernel for speeding up computation time with very small increase in RMSE method using Monte Carlo experiments.

GWR with target points and rough gaussian kernel

Target points:

TP is a subset of size $n_{tp} < n$ of locations.

$W_{tp} = W_{tp}(K(h))$ are matrices based on a $n_{tp} \times n$ matrix given a bandwidth h and a kernel $K()$

$$\hat{\beta}(u_{tp}, v_{tp}) = (X'W_{tp}X)^{-1}X'W_{tp}Y$$

$$\hat{\beta}(u_{\bar{tp}}, v_{\bar{tp}}) = \tilde{W}\,\hat{\beta}(u_{tp}, v_{tp})$$

How to choose the $n \times n_{tp}$ matrix $\tilde{W}$?

- Optimization of $\tilde{W}$ by cross validation ?
- Using a Sheppard kernel with between 12 and 16 neighbors for extrapolating all parameters (Loader 1999, McMillen 2012)
- Relying on cross validated $W(K(h^{\star}))$ to built $\tilde{W}$:
  - Using a new weighting matrix using the same kernel and optimal bandwidth as for fisrt stage estimation, i.e. $K(h^{\star})$.
  - A second alternative, is to rely directly on how tp observations have been weighted in the estimation of $\hat{\beta}(u_{tp}, v_{tp})$

$$\tilde{W} = W'/rowSum(W') = W'/colSum(W)$$

How to choose target points tp?

For local linear smoother, the choice of target points in univariate case can be based:

1. On the density of observations (McMillen 2012 for GWR)
2. equidistributed along the support of the function,
3. or with respect to the curvature of the function.

The last alternative is the most efficient but is tricky for 2D space.

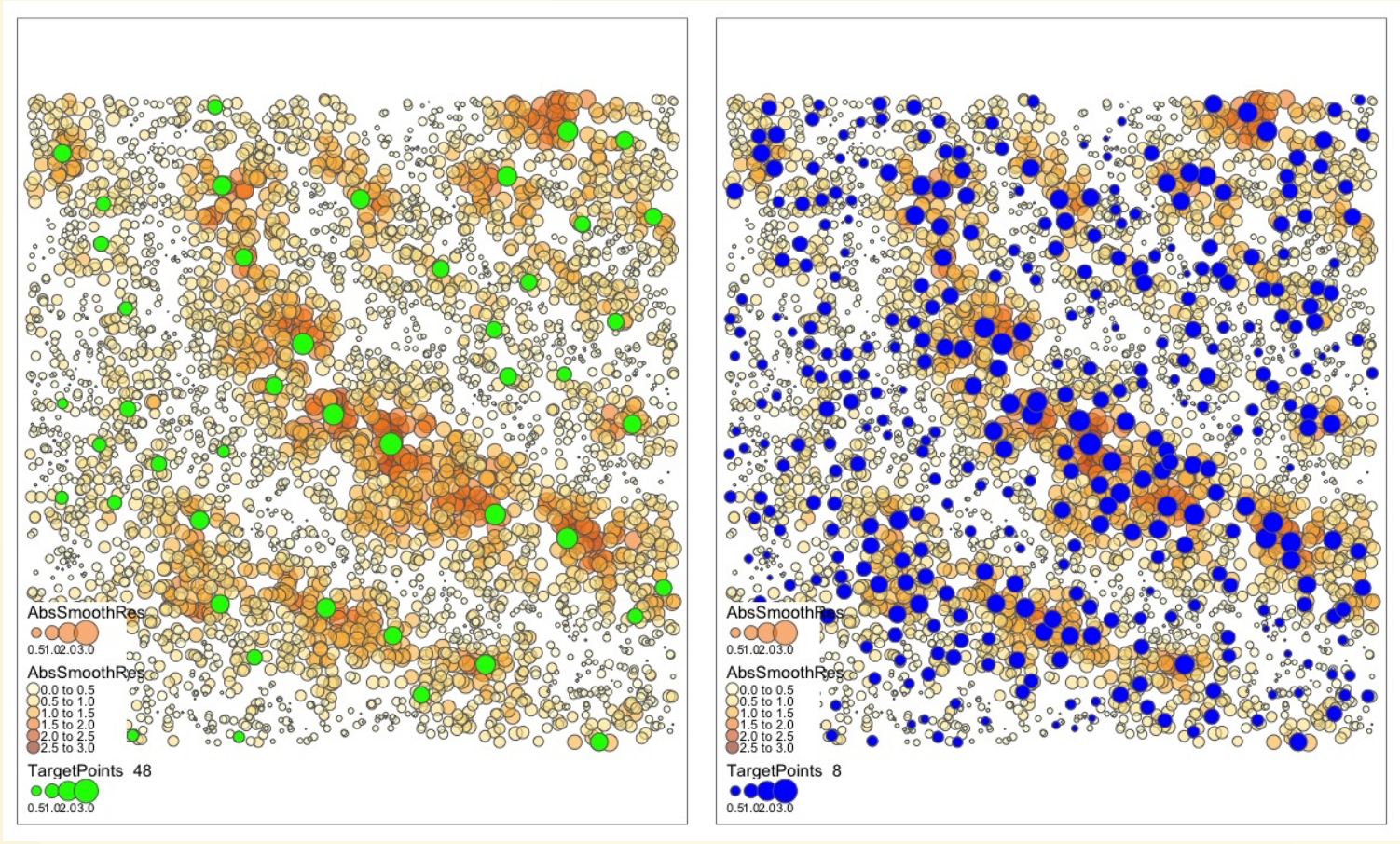
Our proposition to choose target points tp?

Our proposition to deal with 2D curvature of local heterogeneity uses 3 steps:

1. In a first step, we fit an **initial linear model** without taking into account spatial heterogeneity, using OLS, and get the residuals.
2. In a second step, we **smooth the first stage residuals** using an adaptive rectangle kernel based on ***ks*** nearest neighbors,
3. In a third step, the selection of the target points is done by selecting the points where the absolute value of the **smoothed residuals is locally the highest among the *kt* nearest neighbors***.

Example of target points choice using GWR_TP2S method:

$(ks = 16, kt = 48)$ $(ks = 16, kt = 8)$



Monte Carlo Design

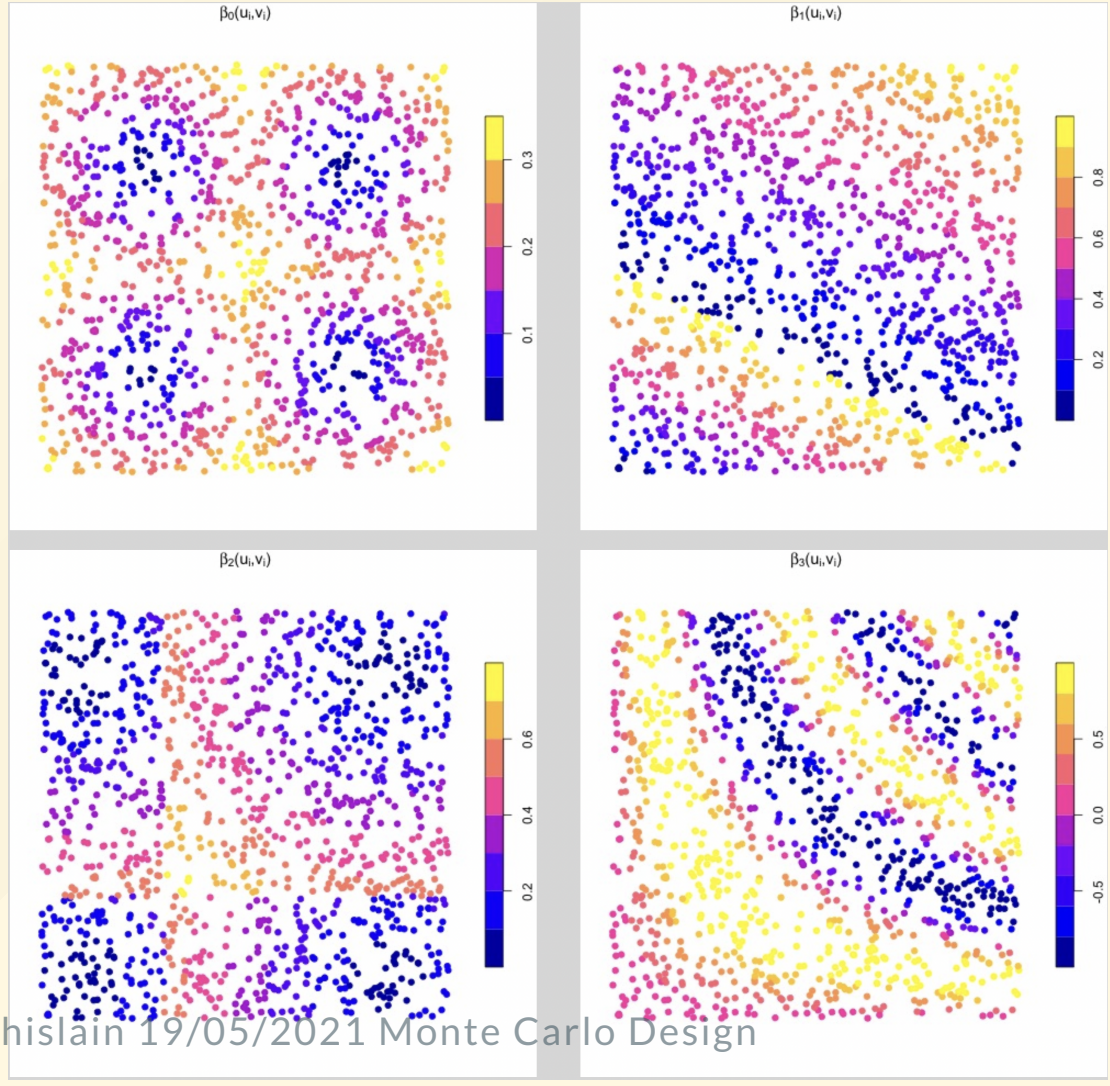
The **first set of experiments (E1)** focuses only on the use of truncated gaussian kernels.

In the **second set of experiments (E2)**, we compare three way of selecting target points for GWR :

- Random selection of target points (GWR_TPR)
- using a quadcell algorithm: space is recursively divided in rectangles with comparable number of observations (Loader 1999) (GWR_TPQ)
- Our proposition (GWR_TP2S)

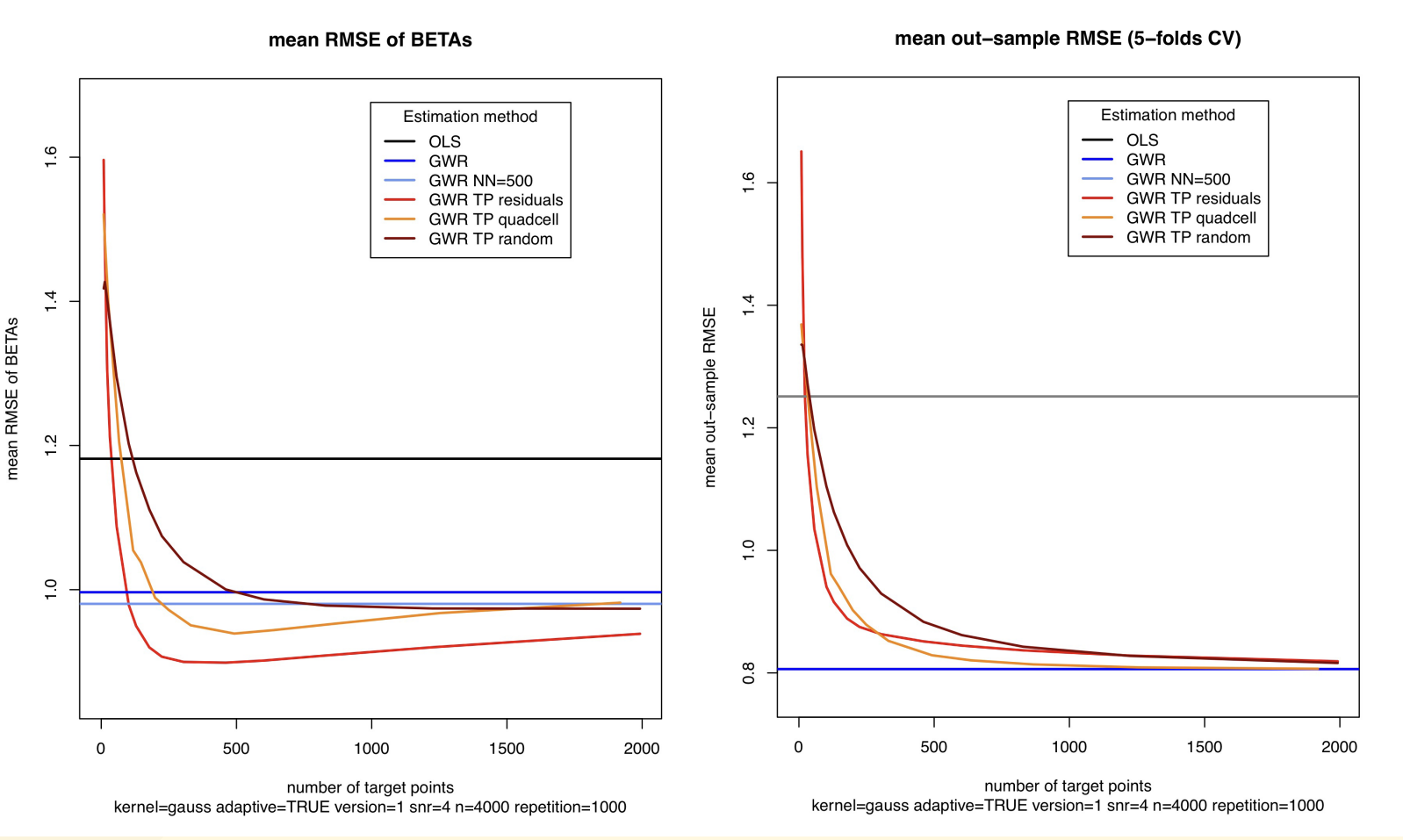
The **last experiment (E3)** focuses on scalability issues by varying the sample size and the cnumber of ovariate.

$$y_i = \beta_0(u_i, v_i) + \sum_{j=1}^{k-1} \beta_j(u_i, v_i) X_j + \epsilon \quad (GWR\ DGP)$$



Monte carlo Results (Exp. 2)

RMSE with respect to the number of target points (kt)

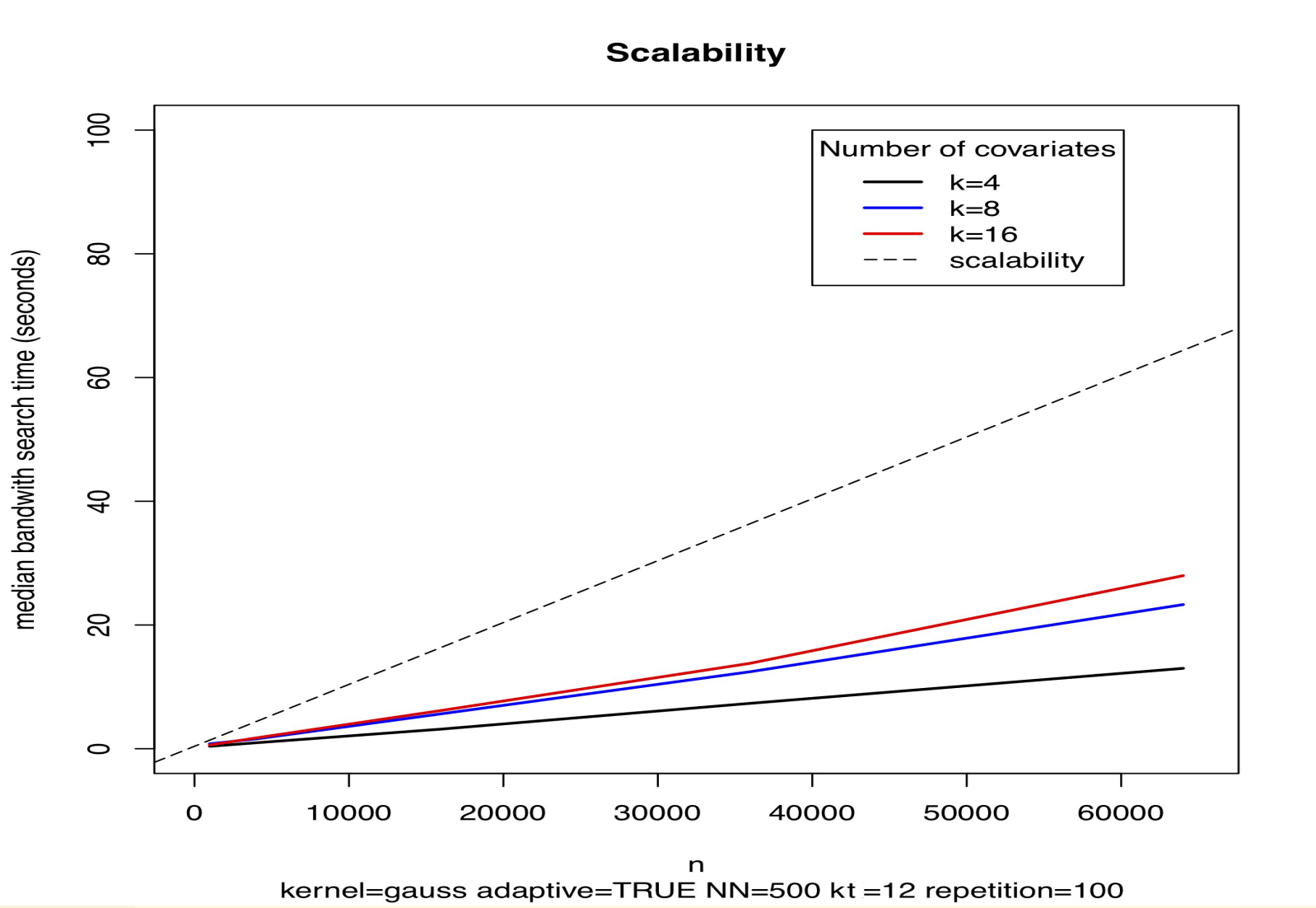


RMSE with respect to the number of target points (ks)

method	kt	ntp	bandwidth time	estimation time	β_k mean RMSE	β_k mean bias	Y Prediction out-sample RMSE
OLS				0.06	1.18	1.01	1.25
GWR		4000	114.40	8.60	1.00	0.77	0.81
GWR_NN		4000	14.83	2.93	0.98	0.76	0.81
GWR_TP2S	2	1992	7.50	1.45	0.94	0.72	0.82
GWR_TP2S	3	1224	4.88	0.90	0.92	0.71	0.83
GWR_TP2S	4	829	3.45	0.62	0.91	0.70	0.84
GWR_TP2S	5	602	2.57	0.46	0.90	0.70	0.84
GWR_TP2S	6	460	2.00	0.37	0.90	0.69	0.85
GWR_TP2S	8	304	1.42	0.26	0.90	0.69	0.86
GWR_TP2S	10	224	1.12	0.21	0.91	0.70	0.88
GWR_TP2S	12	178	0.95	0.18	0.92	0.71	0.89
GWR_TP2S	16	129	0.76	0.14	0.95	0.74	0.92
GWR_TP2S	20	102	0.66	0.12	0.98	0.76	0.94
GWR_TP2S	40	57	0.49	0.09	1.09	0.85	1.03
GWR_TP2S	80	32	0.39	0.08	1.21	0.96	1.16
GWR_TP2S	120	22	0.33	0.07	1.31	1.05	1.25
GWR_TP2S	200	13	0.30	0.06	1.48	1.22	1.48
GWR_TP2S	400	9	0.28	0.06	1.60	1.34	1.65

Table 3
Computation time and RMSE with and without target points based on OLS residuals, n=4000, 1000 repetitions, k=4, version=1, adaptive Gaussian kernel.

Scalability of GWR_TP2S method



Real Estate DATA [work in progress]

- French Single House Sales 2014-2019 (n=2052127)
- k= 40 candidates variables
- 4 regression methods compared:
  - geo-additive SLX model with xgboost,
  - geo-additive SLX model with gamboost,
  - mgwr SLX model
  - mgwr SLX model with target points (kt=8)

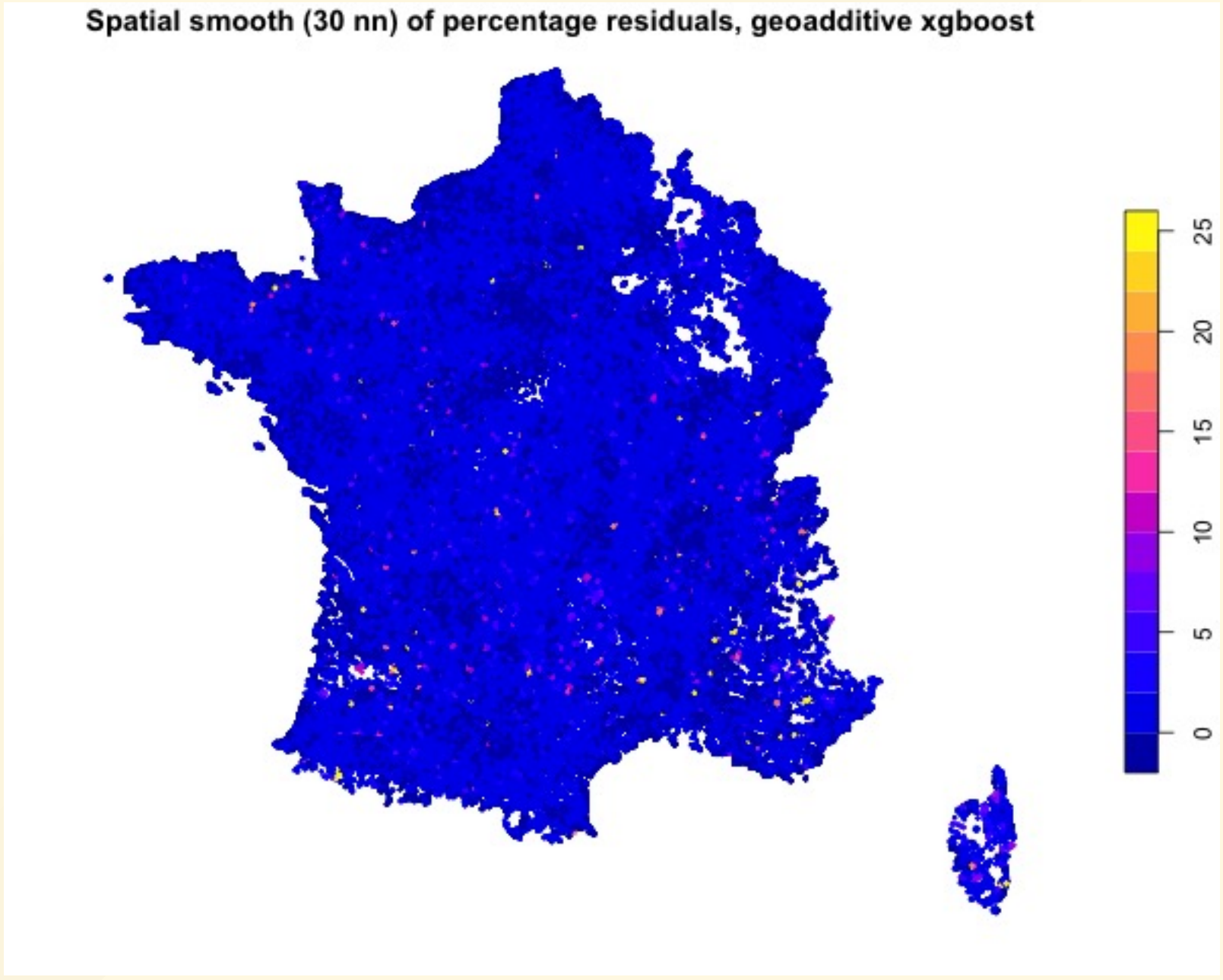
$$y = \beta_c X_c + \beta_v(u_i, v_i) X_v + \epsilon_i \qquad (MGWR)$$

geoadditive XGBOOST : 10 minutes for computation (n=2052127, 6 cores)

5 folds outsample RMSE : 0.44

Spatial Autocorrelation tests (PACA region only):

W2 : Moran I statistic standard deviate = 51.757, p-value < 2.2e-16
W4 : Moran I statistic standard deviate = 50.809, p-value < 2.2e-16
W10: Moran I statistic standard deviate = 75.961, p-value < 2.2e-16
W60: Moran I statistic standard deviate = 130.04, p-value < 2.2e-16


Spatial smooth (30 nn) of percentage residuals, geoadditive xgboost
25
20
15
10
5
0

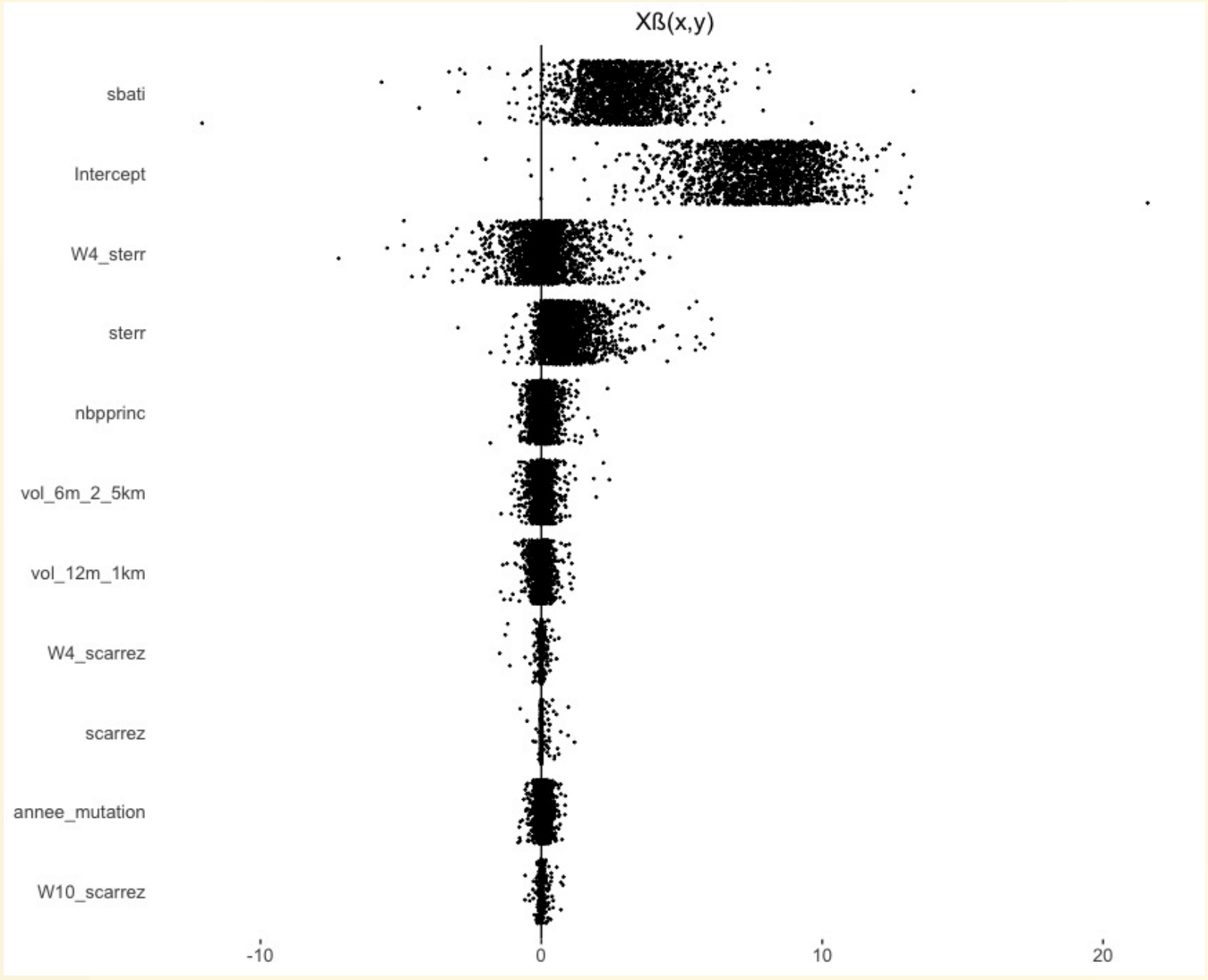
MGWR with TP : 6 minutes for computation with kt=12 (TP= 123 192 obs, 6 cores)

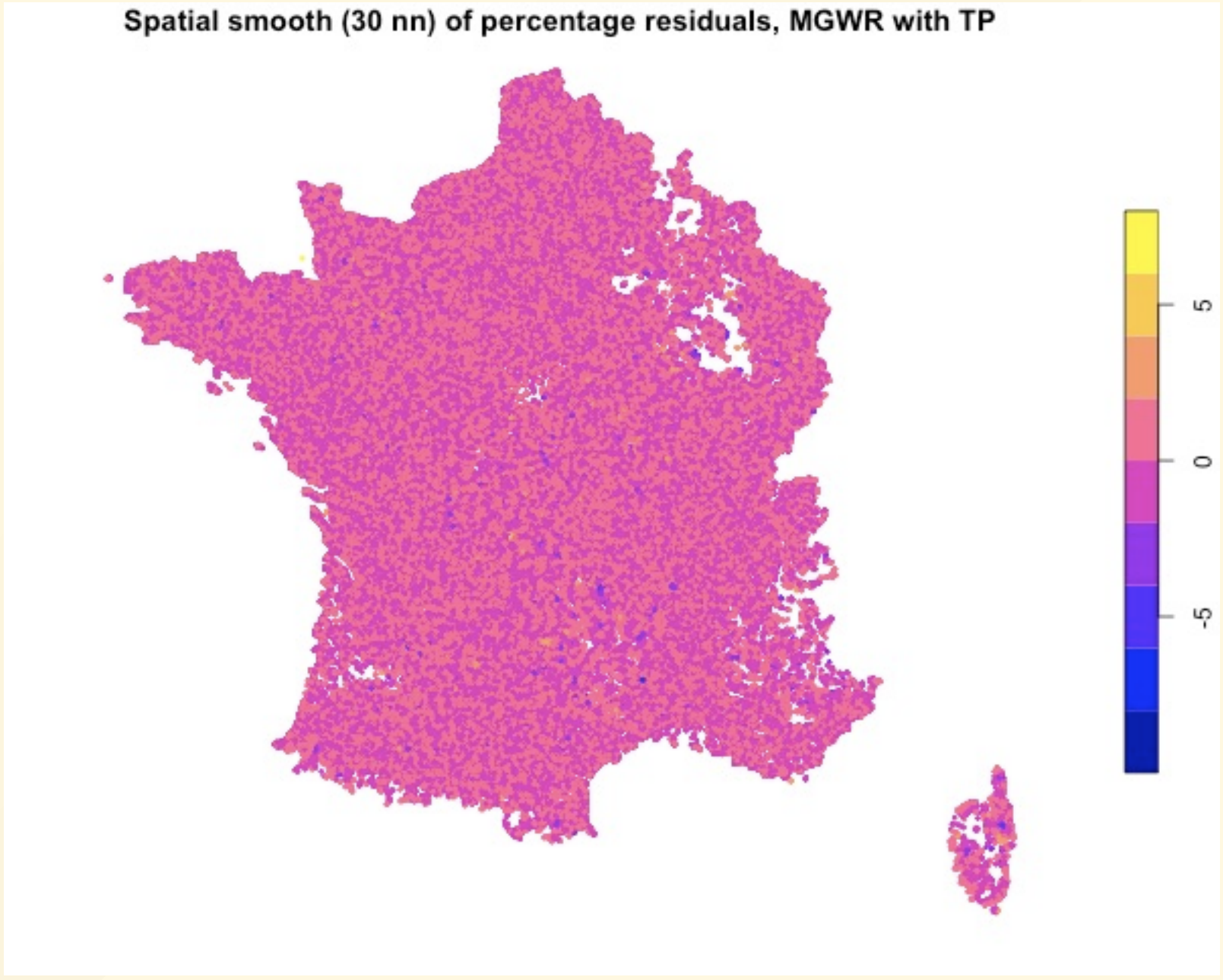
```
prix_mutation ~ nbpprinc + sbati + sterr + scarrez + annee_mutation +
W30_nbpprinc + W10_scarrez + W4_sterr + W4_scarrez + vol_12m_1km + vol_6m_2_5k
d_ecole_ma + d_hop_court_s + d_hop_long_s + d_CBD13+ IQualLife+W30_scarrez

fixed_vars=c('d_ecole_ma','d_hop_court_s','d_hop_long_s','d_CBD13','W30_nbppri
'W30_scarrez','IQualLife')
```

leave one out outsample RMSE : 0.412

Spatial Autocorrelation tests (PACA region only):

W2 : Moran I statistic standard deviate = -0.74538 p-value = 0.772
W4 : Moran I statistic standard deviate = -1.2421 , p-value= 0.8929
W10: Moran I statistic standard deviate = -6.332, p-value = 1
W60: Moran I statistic standard deviate = -13.912 , p-value = 1


Xß(x,y)
sbati
Intercept
W4_sterr
sterr
nbpprinc
vol_6m_2_5km
vol_12m_1km
W4_scarrez
scarrez
annee_mutation
W10_scarrez
-10
0
10
20

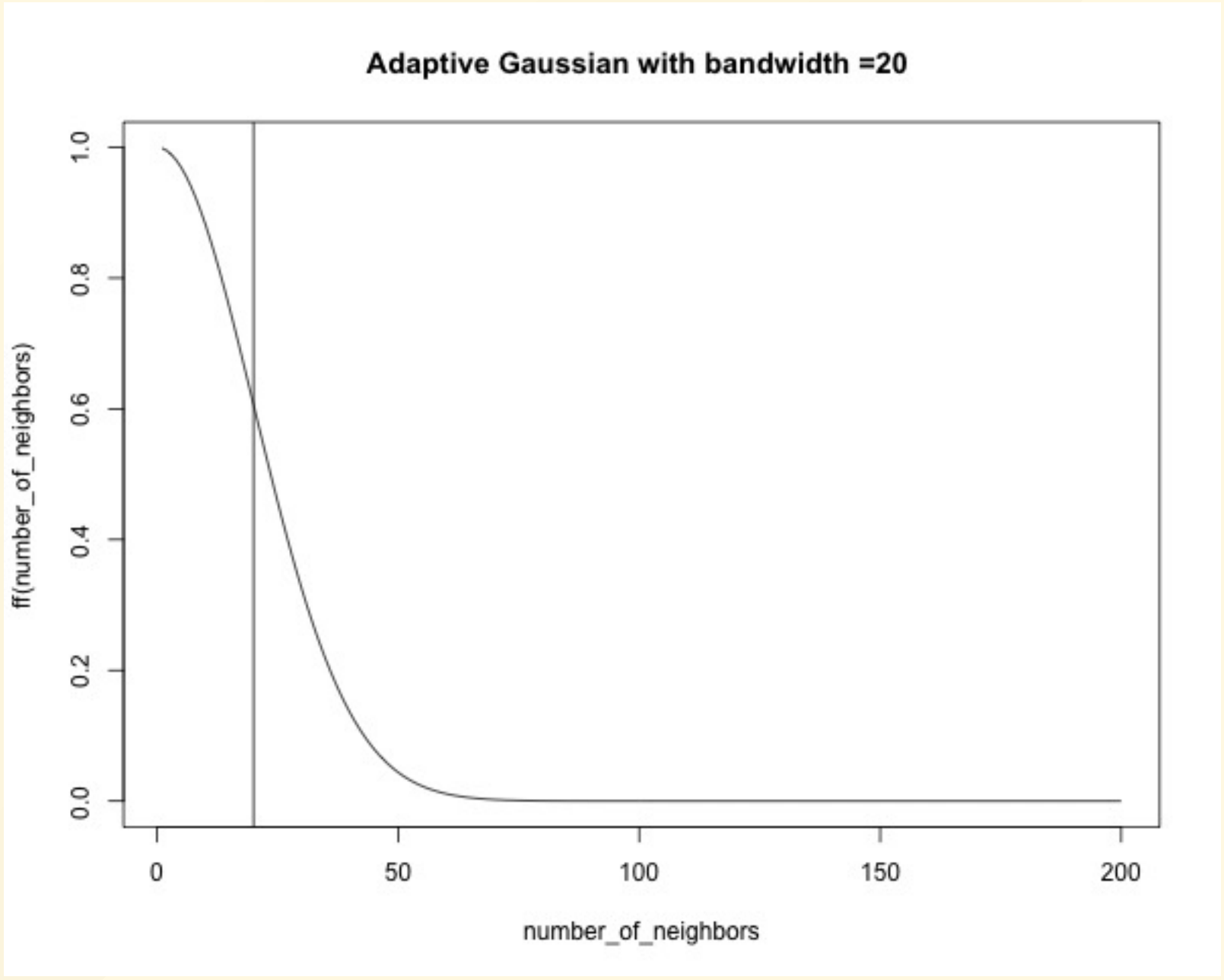

Spatial smooth (30 nn) of percentage residuals, MGWR with TP
5
0
-5

Perspectives

- local bandwidth size (in progress)
- Better control of colinearity issues,
- Studying the interest of stationarity versus non stationarity of autoregressive parameter $\lambda(u_i, v_i)$ when using target points.

---> switching to multiscale GWRSAR (a bandwidth by covariate) framework using a backfitting algorithm.

Thank you for your attention


Adaptive Gaussian with bandwidth =20
ff(number_of_neighbors)
0.0
0.2
0.4
0.6
0.8
1.0
0
50
100
150
200
number_of_neighbors

Our proposition to choose target points tp?

More precisely, in a first step, we compute for all i the smooth of OLS residuals using the ks neighbors of i, including i. If we note $v_{i,ks}$ the set of the ks first neighbors of i, we compute $\forall i$:

$$\tilde{\epsilon}_i = \frac{\sum_{j \in v_{i,ks}} \epsilon_j}{ks}$$

Then, we identify the set of target points TPC as follows:

$$i \in TPC \; if \; |\tilde{\epsilon}_i| \geq |\tilde{\epsilon}_j| \;\; \forall j \in v_{j,kt}$$