SpatialRegimes package: a brief introduction to spatial clusterwise regression with a focus on SkaterF function

F. Vidoli and R. Benedetti

October 30, 2020

Spatial regimes

The SpatialRegimes package contains functions for the estimation of spatial regimes ... but what it means, more closely, the term ”spatial regime”? Spatial regime is an aggregation of neighboring units that are homogeneous in functional terms or that ”share” the same relationship between a dependent variable and some covariates.

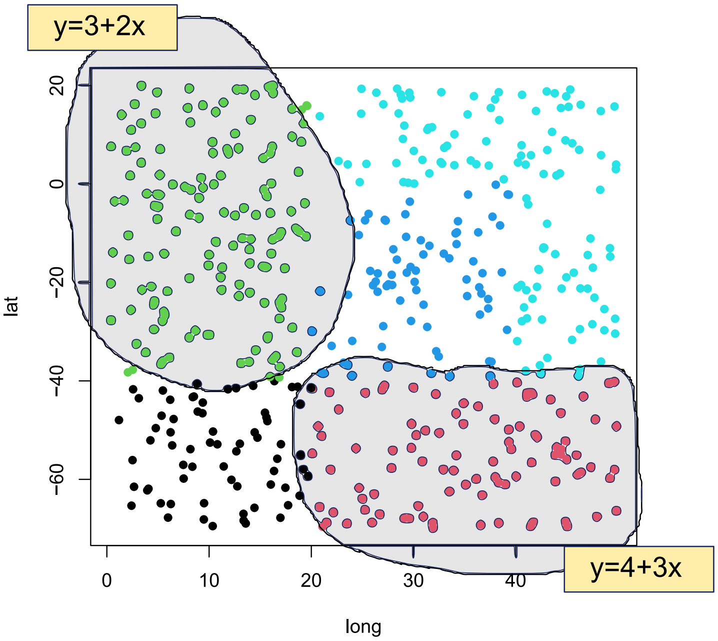


Figure 1: Just a stylized example

Figure 1 shows a stylized case in which neighboring units show a different relationship between a variable y and a covariate x; in the case of subjects in green the relation $y = f(x)$ assumes, in fact, estimated value equal to $y = 3+2x$, while in the case of subjects in red $y = 4 + 3x$.

It is a tool, therefore, extremely useful when - for example - social and economic phenomena show heterogeneous behaviors depending on the territories in which they occur, territories that most often does not coincide with mere administrative boundaries, but instead have to be identified **at the same time** as the functional estimate.

The identified territories are maximally homogeneous in functional terms and therefore help - for example - the researcher to control the **real fixed regional effects** where the term regional is here to be understood not in an administrative sense, but as a territorial aggregation where it is maximally homogeneous the relationship you want to study.

The term ”spatial regime”, therefore, should not be understood as a synonym for ”cluster”. More precisely, the term ”cluster” does not presuppose any functional relationship between the variables considered, while the term ”regime” is linked to a regressive relationship underlying the spatial process. Identifying different spatial regimes, therefore, is equivalent to estimating different functional production regimes.

In order to illustrate the package more simply and/or interactively, a Shiny application has been designed and it is available here:

```
https://fvidoli.shinyapps.io/SpatialRegimes_app/
```

In the second part of the vignette, we will focus our attention on a specific function called SkaterF.

SkaterF function

SkaterF function allows to estimate territorially defined areas in which the production units are maximally homogeneous in functional terms - defining a spatial regimes - and at the same time inhomogeneous with the other ones.

These areas, which are the hierarchical subdivision of the territory, are identified as spatially bound areas or as areas in which adjacent units (intended as neighboring, bordering, contiguous) are constrained to be part of the same cluster; this constraint, therefore, allows to obtain homogeneous closed areas or in which all units belong to the same cluster.

So let's begin with a practical example!

A quick set-up

Let's start by loading the library SpatialRegimes and the simulated dataset SimData; in this dataset 500 units, their respective coordinates, three covariates called A, L and K and a variable that identifies the membership to a specific area (clu) have been simulated. The dependent variable Y has been generated as:

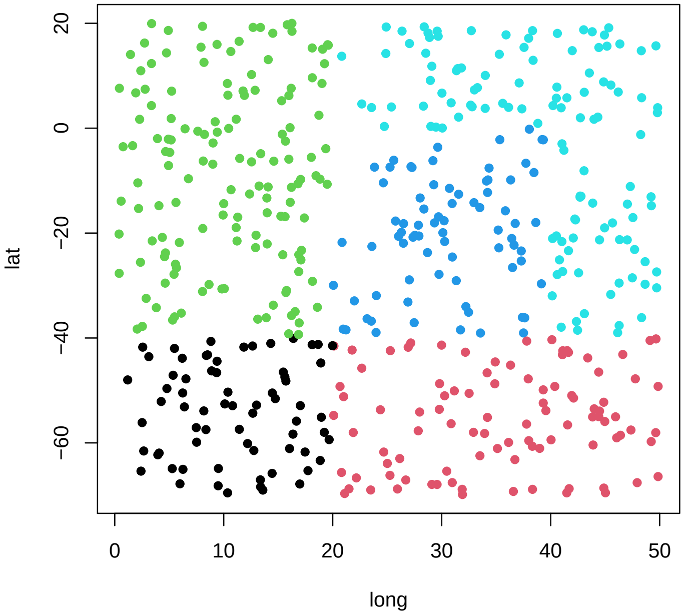
$$Y = \begin{cases} 13 + 0.5 * A + 0.3 * L + 0.2 * K + \epsilon, \text{if } i \in \text{cluster } 1 \\ 11 + 0.8 * A + 0.1 * L + 0.1 * K + \epsilon, \text{if } i \in \text{cluster } 2 \\ 9 + 0.3 * A + 0.2 * L + 0.5 * K + \epsilon, \text{if } i \in \text{cluster } 3 \\ 7 + 0.4 * A + 0.3 * L + 0.3 * K + \epsilon, \text{if } i \in \text{cluster } 4 \\ 5 + 0.2 * A + 0.6 * L + 0.2 * K + \epsilon, \text{if } i \in \text{cluster } 5 \end{cases} \quad (1)$$

where ϵ is a normally distributed error term $\in \mathcal{N}(0, 1)$.

```
> library(SpatialRegimes)
> data(SimData)
> coords = cbind(SimData$long, SimData$lat)
```

Our aim is, therefore, to use SkaterF function to estimate the adherence to the simulated functional cluster as plotted below.

```
> plot(lat~long,SimData,col=clu,pch=16)
```



The first step is the construction of the neighborhood that can be carried out according to several choices; please refer to the `https://cran.r-project.org/web/packages/spdep/vignettes/nb.pdf` for an exhaustive review in R.

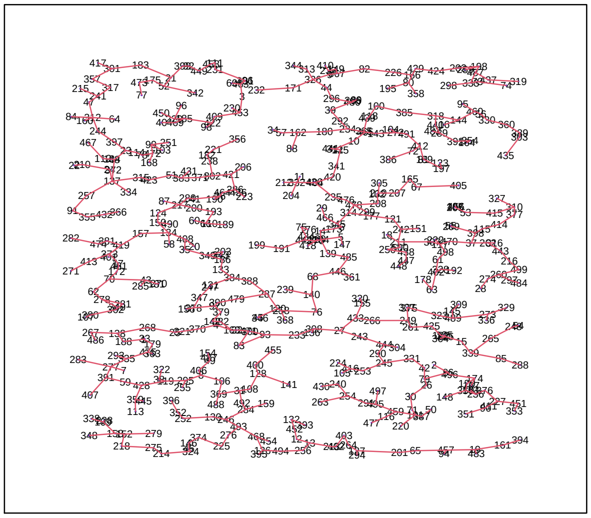
```
> neighbours = tri2nb(coords, row.names = NULL)
> bh.nb <-  neighbours
> lcosts <- nbcosts(bh.nb, SimData)
> nb <- nb2listw(bh.nb, lcosts, style="B")
```

After identifying the typology of neighborhood between units, function mstree from spdep package is used to identify the minimal spanning tree as the smaller class of possible partitions of a graph by pruning edges with high dissimilarity.

```
> mst.bh <- mstree(nb,5)
> edges1 = mst.bh[,1:2]
```

The figure of the minimal spanning tree is shown below.

```
> plot(mst.bh, coords, col=2,
+       cex.lab=.5, cex.circles=0.035, fg="blue")
```



Therefore, all the input are in place to provide to our function to identify - according to the chosen neighborhood and the functional specification chosen - the spatial regimes.

SkaterF function needs (like the *k*-means procedure) to indicate a number of cuts (ncuts)[1] and it allows to indicate a minimum number of subjects within each cluster (crit). OLS (method=1) is currently the only estimator implemented.

[1] Please note that the desired number of spatial regimes is equal to ncuts +1.

```
> ncuts1 = 4
> crit1 = 10
> coly1 = c("y_ols")
> colx1 = c("A","L","K")
> sk = SkaterF(edges = edges1,
+               data= SimData,
+               coly = coly1,
+               colx= colx1,
+               ncuts=ncuts1,
+               crit=crit1,
+               method=1)
> SimData$regimes = sk$groups
```

Results can be finally appreciated both through a visual spatial analysis and by implementing regressive analysis, one for each group obtained.

```
> plot(lat~long,SimData,col=regimes,pch=16)
```

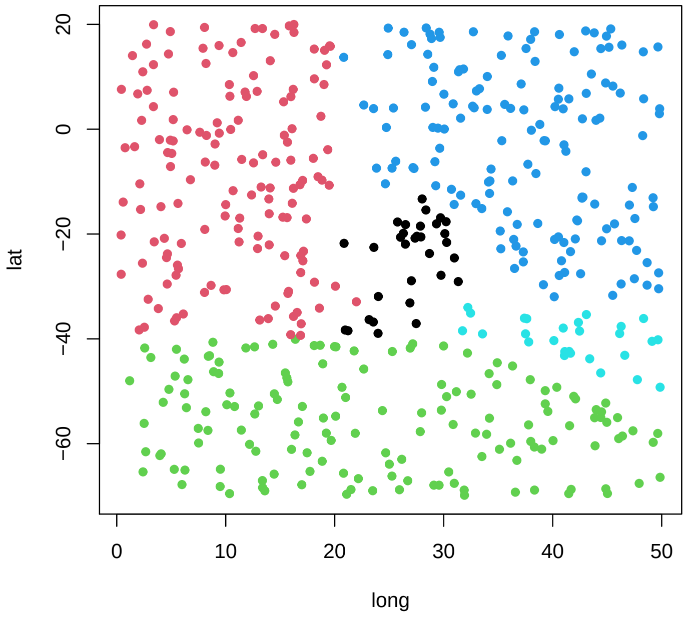


Table 1

	Dependent variable:				
	Y				
	(1)	(2)	(3)	(4)	(5)
A	0.485^{*}	0.406^{***}	0.552^{***}	0.200	0.343
	(0.264)	(0.115)	(0.150)	(0.142)	(0.628)
L	0.245	0.328^{***}	0.288^{**}	0.541^{***}	−0.145
	(0.193)	(0.117)	(0.133)	(0.150)	(0.825)
K	0.496^{**}	0.771^{***}	0.326^{**}	0.589^{***}	1.701^{**}
	(0.207)	(0.112)	(0.142)	(0.153)	(0.728)
Constant	6.640^{***}	7.597^{***}	11.384^{***}	4.427^{***}	5.634
	(0.994)	(0.559)	(0.664)	(0.699)	(3.387)
Observations	31	145	151	146	27
R^2	0.300	0.332	0.138	0.183	0.198
Adjusted R^2	0.222	0.317	0.120	0.166	0.093
Residual Std. Error	0.903 (df = 27)	0.986 (df = 141)	1.254 (df = 147)	1.297 (df = 142)	2.508 (df = 23)
F Statistic	3.852^{**} (df = 3; 27)	23.326^{***} (df = 3; 141)	7.840^{***} (df = 3; 147)	10.629^{***} (df = 3; 142)	1.893 (df = 3; 23)
Note:					$^{*}p<0.1$; $^{**}p<0.05$; $^{***}p<0.01$