Bayesian mixture model for environmental application

P. Bogani, P. Botta, S. Caresana, R. Carrara, G. Corbo, L. Mainini

Case study

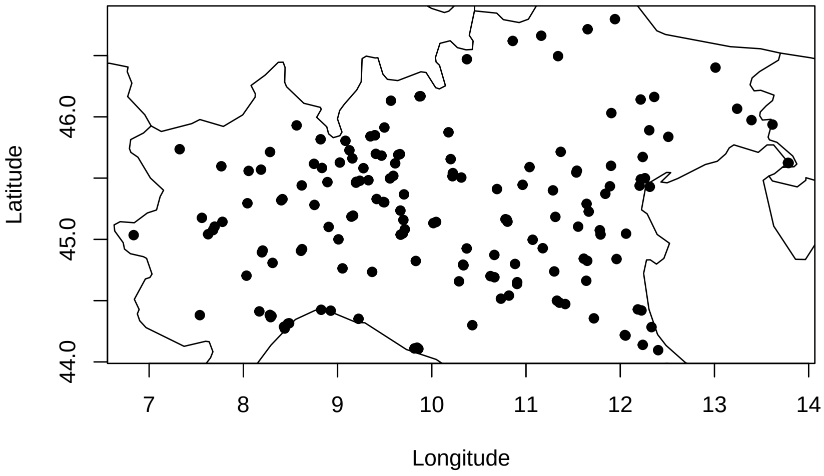
Proposed model

First we import the datasets with the time series and the stations' information, and retrieve the coordinates of the stations.

```
pollutant <- read.csv('code/data/pollutant.csv')
stationInfo <- read.csv('code/data/stationsInfo.csv')
##get latitude and longitude
latitude <- c()
longitude <- c()
region <- c()
for (i in 1:dim(pollutant)[2]) {
  site <- colnames(pollutant)[i]
  latitude <- c(latitude,stationInfo$latitude[which(stationInfo$site==site)])
  longitude <- c(longitude,stationInfo$longitude[which(stationInfo$site==site)])
  region <- c(region,stationInfo$region[which(stationInfo$site==site)])
}
```

The stations are distributed in all the Northern Italy:

```
plot(longitude,latitude,pch=19,cex=0.9,xlab='Longitude',ylab='Latitude')
map("world",add=T)
```



Then we run our code on the dataset. The value of `frequency`, `seasonfreq` and `seasondelay` are set to construct the design matrix $\mathbf{Z}$.

```r
pb=progress_bar$new(total=300)
invisible(pb$tick(0))
tseriesc.out <- tseriesclust(pollutant,maxiter=300,thinning=2,frequency = 365,
                              seasonfreq = 2,seasondelay = 80)
```

Since the computational workload is not negligible and the estimated time to run is approximately 90 minutes, we decided to import the saved R.Data. The code is still replicable from the RMarkdown `Case study.Rmd`, running the above chunk.

```r
load("code/results/result_proposedmodel.RData")
```

We then used the `salso` algorithm developed by Dahl which implements a greedy, stochastic search given the number of desired clusters, in our case 4. We choose the partition minimizing the posterior of the VI loss function:

$$\hat{\mathbf{c}}^* = \underset{\hat{\mathbf{c}}}{\text{argmin}}\mathbb{E}\left(L_{VI}(\mathbf{c},\hat{\mathbf{c}}) \mid \mathcal{D}\right) = \underset{\hat{\mathbf{c}}}{\text{argmin}}\sum_{i=1}^{n} \log_2\left(\sum_{j=1}^{n} \mathbb{I}\left(\hat{c}_j=\hat{c}_i\right)\right) -2\sum_{i=1}^{n} \mathbb{E}\left(\log_2\left(\sum_{j=1}^{n} \mathbb{I}\left(c_j=c_i\right)\mathbb{I}\left(\hat{c}_j=\hat{c}_i\right)\right) \mid \mathcal{D}\right),$$

where $\mathcal{D}$ represents data and $\mathbf{c}$ is the true cluster partition.

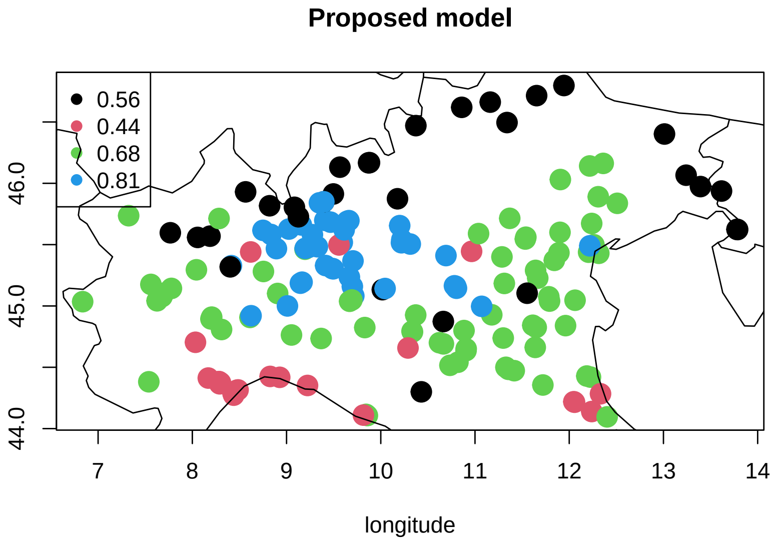
```r
gnstar <- as.numeric(salso(tseriesc.out$memorygn, maxNClusters=4,
                            loss=VI(a=0.5), nRuns=50, nCores=2))
```

We computed the mean value of ρ in each cluster:

```r
#compute mean rho
d <- dim(tseriesc.out$rhosample)[1]
rho <- tseriesc.out$rhosample[d,]
rho_m <- c()
for (i in 1:4) {
  rho_m <- c(rho_m, round(mean(rho[which(gnstar==i)]),2) )
}
```

and plotted the cluster obtained on the map:

```r
plot(longitude,latitude,pch=19,cex=2,main = "Proposed model"
     , ylab="",col=gnstar)
legend('topleft',legend=rho_m,col=1:4,pch=19)
map("world",add=T)
```



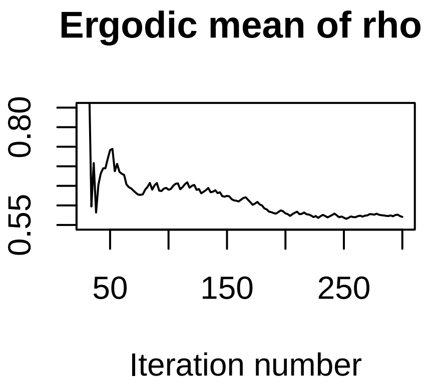
We can see that the four clusters obtained can be divided into 4 distinct natural regions.

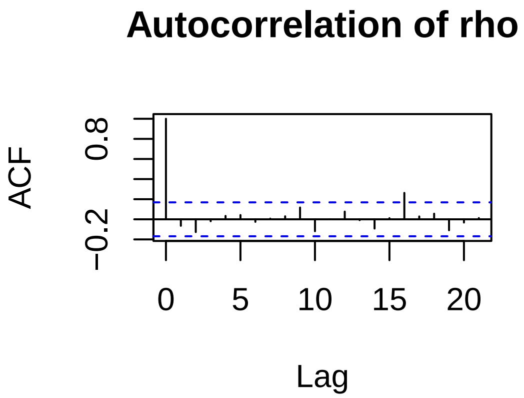
The Milan area has the highest persistence cluster, and the cities of the Padan Pianury are home to another significant cluster. The other two clusters, with less persistence, are in the Genoa or marine cities zone and in the stations of the Alps.

In fact, a search of the literature revealed studies analysing PM10 concentration that confirmed that the persistence is often higher in the proximity of urban areas and dry zones while it is lower in greener areas and more breezy regions.

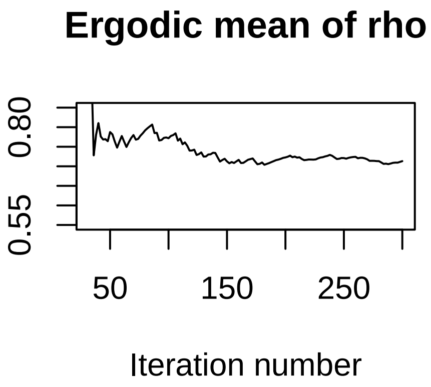
We've then decided to compare the convergence of the ergodic mean of ρ of two stations belonging to different clusters.

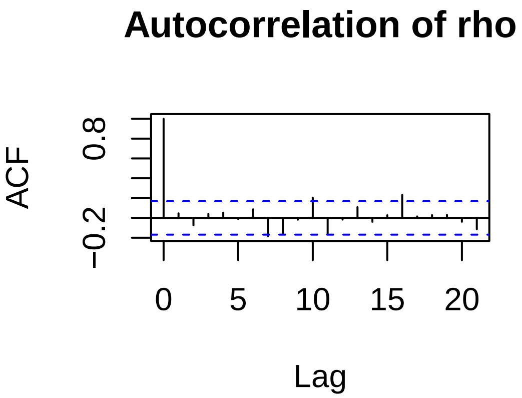
```
x <- 30:300                          #x is defined just for visual purposes
x <- x[which(x %% 2 == 0)]
x <- x[-1]
par(mfrow=c(2,1))
station <- 112
rhosample <- tseriesc.out$rhosample[,station]
plot(x,cumsum(rhosample)/1:d,type = "l",main = "Ergodic mean of rho "
     ,xlab = "Iteration number",ylab = "",ylim=c(0.55,0.85))
acf(rhosample, main='Autocorrelation of rho')
```





```r
gnstar[112]      #2nd cluster, "red" one
par(mfrow=c(2,1))
station <- 102
rhosample <- tseriesc.out$rhosample[,station]
plot(x,cumsum(rhosample)/1:d,type = "l",main = "Ergodic mean of rho"
     ,xlab = "Iteration number",ylab = "",ylim=c(0.55,0.85))
acf(rhosample, main='Autocorrelation of rho')
```



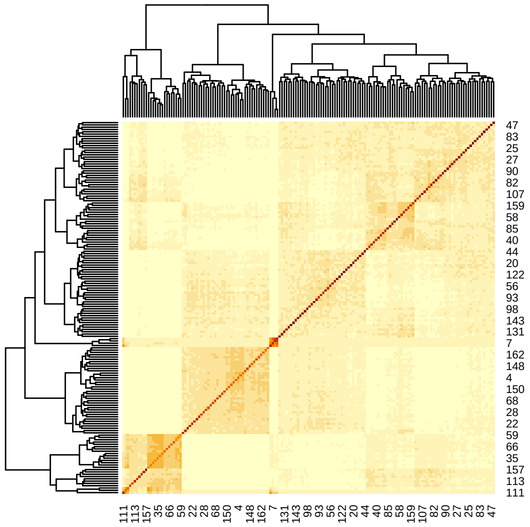


```
gnstar[102]      #4th cluster, "blue" one
```

We can note how indeed the two ergodic mean converge to two different values, with the Milan area station persistence higher than the one in Genoa zone.

Finally, we plotted the similarity matrix of the cluster obtained:

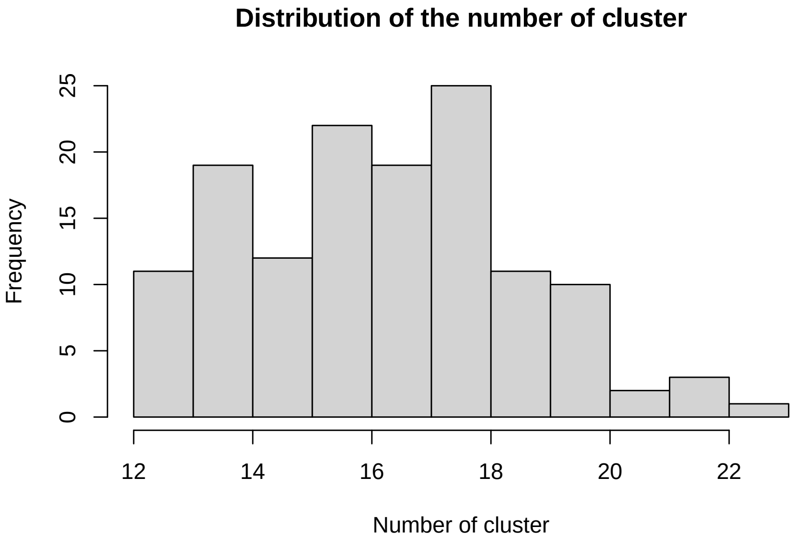
```
simm <- comp.psm(tseriesc.out$memorygn)
heatmap(as.matrix(simm))
```



We can indeed denote from the dendrogram the formation of the 4 clusters.

The following figure shows the distribution of the number of clusters obtained by the algorithm after 300 iterations.

```
hist(tseriesc.out$msample, main="Distribution of the number of cluster"
     ,xlab = "Number of cluster")
```



We can see that even if the number of stations is high (162), the minimum number of clusters obtained by the algorithm is still high (12). Deeper analysis and tuning of the hyper-parameters, number of iterations or the introduction of an acceleration step (in case we assume different priors) could be useful to have a histogram of the number of clusters more centered in a small value than the one obtained.

Starting model

Using the same dataset its reproducible also the clustering using the starting model.

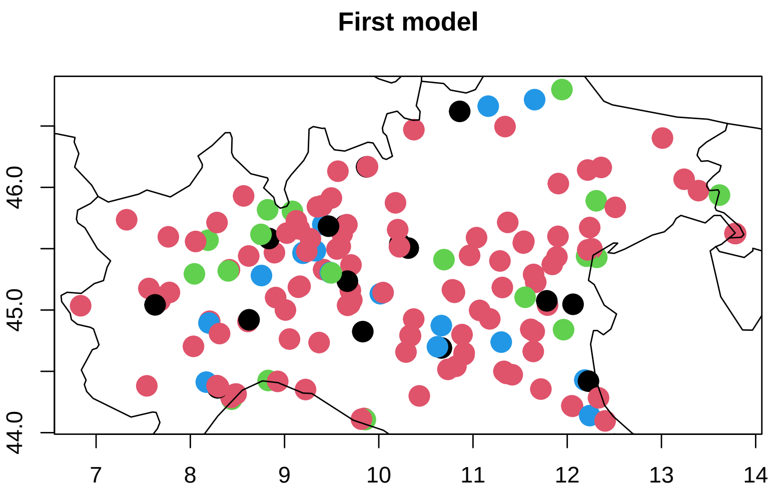
```
tseriesc.out <- tseriesclust_first(pollutant,maxiter=300,
                  thinning=2,frequency = 365, seasonfreq = 2,seasondelay = 80)
```

Same as before we load the results, but all the code is reproducible in the file `Case study.Rmd`, running the above chunk.

```
load("code/results/result_firstmodel.RData")
```

```
gnstar_f <- as.numeric(salso(tseriesc.out.first$memorygn, maxNClusters=4,
                              loss=VI(a=0.5), nRuns=50, nCores=2))
```

```
plot(longitude,latitude,pch=19,cex=2,main = "First model", xlab="", ylab="",col=gnstar_f)
map("world",add=T)
```



We could have expected a poor outcome because there isn't much of a difference in level or trend between the stations, as can be seen in the matplot of the PM10 concentration in Northern Italy in the data exploration section.