ANOVA-DDP harmonic regression on Telecom areal data

Irene Azzini, Michele Bucelli, Elena Morelli
Tutors: Annalisa Cadonna, Andrea Cremaschi

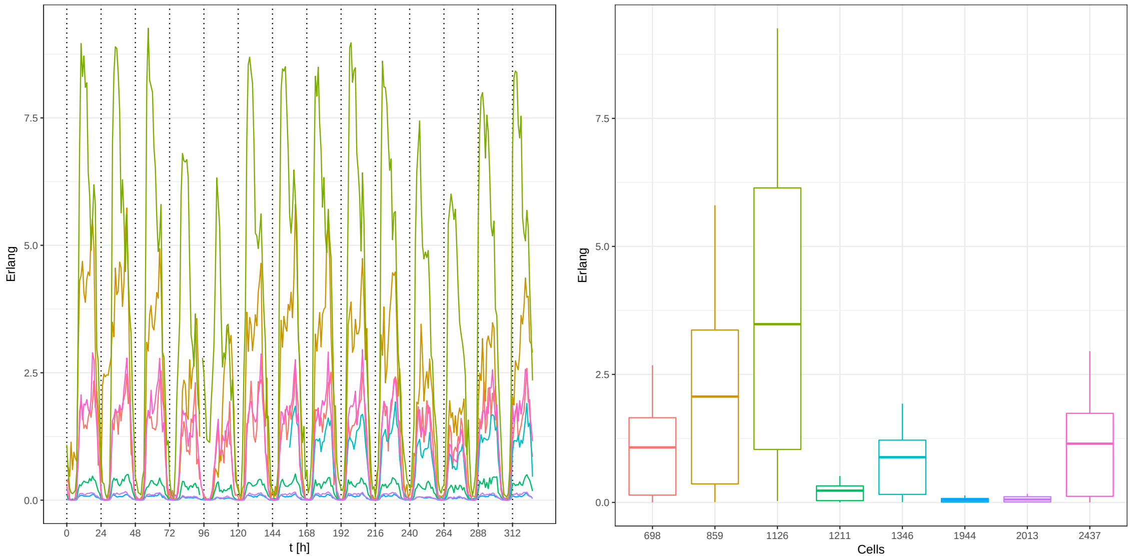
1 February 2019

Contents

1 Introduction

We analyzed a dataset consisting of one time series for each cell of a grid. We considered a non parametric harmonic regression model for the data, involving only correlation in time, without any spatial effects. We aimed at using the posterior of the model, with reference in particular to the infinite mixture representation of the Dirichlet process, to infer a clustering structure for the cells.

We implemented a Gibbs sampler algorithm in C++ to simulate from the posterior, we performed a posterior simulation and analyzed its results.



(a) Erlang number over time for a few cells (b) Erlang number distribution for a few cells

Figure 1

2 Dataset

We analyzed a dataset provided from Telecom Italia, consisting of recordings of the **Erlang number** over a grid of $48 \times 54 = 2592 = n$ cells covering the city of Milan, every hour over the course of two weeks, for a total of $T = 327$ hours. 1448 of the observations are missing.

The Erlang number in a given cell is a positive number that represents the average number of mobile phones that are simultaneously using the network for calling within that cell.

2.1 Preliminary analysis

Plotting the Erlang number over time for some cells, as in figure 1a, a **periodic behavior** can be observed. Obscillating patterns of daily and weekly period appear very clearly. In particular, it appears that there is a **different behavior between weekdays and weekends**, with recordings during weekdays being higher than those during weekends.

Although the periodic structure is common to all cells, there are significant differences in terms of amplitude of the obscillations, as highlighted in figure 1b. These differences appear to be due to the location of the cell. Plotting the average logarithm of the Erlang number for each cell as in figure 2 we can see a significant difference between the central area and the periphery.

3 Statistical model

Since the Erlang number is positive, we first take a logarithmic transformation of it, which is supported on the real line. Denoting with E_{it} the Erlang number recorded at cell i and time t, and with ϵ a small offset (we took $\epsilon = 10^{-8}$) that allows to treat zeros, we define:

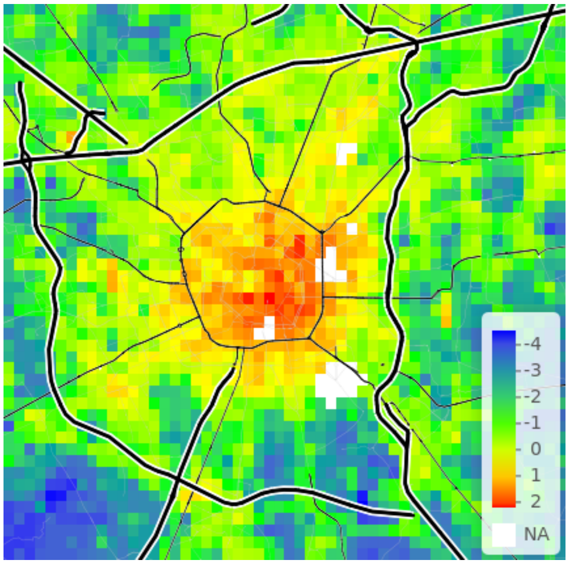


Figure 2: Time-average of the Erlang number for each cell, in logarithmic scale; cells containing NAs are colored in white

$$Y_{it} = log(E_{it} + \epsilon) \tag{1}$$

The periodic behaviour induces a **harmonic regression** model, that is a linear model whose regressors are the harmonics of time. Moreover, the weekly periodic pattern suggests to categorize the times according to the weekday/weekend criterion. The model we will consider is based on using an **ANOVA-DDP prior** [1] for the regression coefficients:

$$\begin{aligned}
&\boldsymbol{x}_t = (1, \cos(\omega_1 t), \sin(\omega_1 t), ..., \cos(\omega_p t), \sin(\omega_p t))' \\
&\phi_t = \begin{cases} 0 & \text{on weekdays} \\ 1 & \text{on weekends} \end{cases} \\
&Y_{it} = \boldsymbol{x}_t' \boldsymbol{\beta}_{i\phi_t} + \epsilon_{it} && \epsilon_{it} \mid \sigma^2 \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \\
&\boldsymbol{\beta}_i \mid P \overset{iid}{\sim} P \\
&P \sim \text{ANOVA-DDP}(P_0, \alpha_0) && \alpha_0 > 0 \\
&P_0 \mid \sigma^2_\beta = \mathcal{N}_{2p+1}(\mathbf{0}, \sigma^2_\beta \boldsymbol{I}) \\
&\sigma^2 \sim \text{Inv-Gamma}(a_0, b_0) && a_0 > 0, b_0 > 0 \\
&\sigma^2_\beta \sim \text{Inv-Gamma}(\nu_0, \kappa_0) && \nu_0 > 0, \kappa_0 > 0
\end{aligned} \tag{2}$$

The model can be rewritten in the following way to make explicit the dependence of $\boldsymbol{\beta}$ on the categories ϕ_t and reduce the ANOVA-DDP to a DDP:

$$
\begin{aligned}
& Y_{it} = \begin{cases} \boldsymbol{x}_t'\boldsymbol{\beta}_{i0} + \epsilon_{it} & \text{if } \phi_t = 0 \text{ (weekdays)} \\ \boldsymbol{x}_t'\boldsymbol{\beta}_{i1} + \epsilon_{it} & \text{if } \phi_t = 1 \text{ (weekends)} \end{cases} && \epsilon_{it} \mid \sigma^2 \overset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2) \\
& \boldsymbol{\beta}_i = (\boldsymbol{\beta}_{i0}, \boldsymbol{\beta}_{i1}) && \\
& \boldsymbol{\beta}_i | P \overset{iid}{\sim} P && \\
& P \sim \text{DP}(P_0, \alpha_0) && \alpha_0 > 0 \\
& P_0 \mid \sigma_\beta^2 = \mathcal{N}_{2(2p+1)}(\boldsymbol{0}, \sigma_\beta^2 \boldsymbol{I}) && \\
& \sigma^2 \sim \text{Inv-Gamma}(a_0, b_0) && a_0 > 0, b_0 > 0 \\
& \sigma_\beta^2 \sim \text{Inv-Gamma}(\nu_0, \kappa_0) && \nu_0 > 0, \kappa_0 > 0
\end{aligned}
\tag{3}
$$

Within this context, the coefficients $\boldsymbol{\beta}_{i0}$ and $\boldsymbol{\beta}_{i1}$ are the regression coefficients for weekdays and weekends respectively. Do notice that the dimensionality of the space of the regression coefficients is duplicated in this new model.

A more compact representation of the same model can be obtained as follows:

$$
\begin{aligned}
& \boldsymbol{h}_t = \begin{cases} (\boldsymbol{x}_t, \boldsymbol{0}) & \text{if } \phi_t = 0 \text{ (weekdays)} \\ (\boldsymbol{0}, \boldsymbol{x}_t) & \text{if } \phi_t = 1 \text{ (weekends)} \end{cases} \\
& Y_{it} = \boldsymbol{h}_t'\boldsymbol{\beta}_i + \epsilon_{it}
\end{aligned}
\tag{4}
$$

This way, the regression parameter is the vector $\boldsymbol{\beta}_i$, whose dimension is twice the dimension of the original regression coefficients $\boldsymbol{\beta}_{i\phi_t}$.

The Dirichlet process can be given a representation in terms of **infinite mixture** that is used in the sampling algorithm. We will denote with c_i a label for the component associated to the i-th observation. Observations Y_{it} with the same label c_i come from the same component, and therefore have the same regression coefficients $\boldsymbol{\beta}_i$.

4 Choice of the hyperparameters

The values of a_0, b_0, ν_0 and κ_0 are supposedly positive and fixed a priori. We want the prior distributions of σ^2 and σ_β^2 to be **uninformative**, because such distributions have a significant impact on the result, especially with respect to the clustering structure we have interest in finding.

It would be therefore appropriate to use small values for all of them (e.g. $a_0 = 0.001$, $b_0 = 0.001$) so that the prior is very dispersed and has a small weight in the posterior. Taking this argument further, we considered to choose the **improper priors** obtained by taking $a_0 = 0$, $b_0 = 0$, $\nu_0 = 0$ and $\kappa_0 = 0$. This yields:

$$\pi(\sigma^2) \propto \frac{1}{\sigma^2} \tag{5}$$

$$\pi(\sigma_\beta^2) \propto \frac{1}{\sigma_\beta^2} \tag{6}$$

The posterior distributions can be still computed thanks to conjugacy and are proper Inverse Gamma distributions.

5 Sampling from the posterior

We simulate from the posterior of model (3) using a **Gibbs sampler** algorithm. The algorithm we implemented is heavily based on algorithm 2 from [2] and on the algorithm described in [1].

5.1 Computation of the full conditionals

The variables whose posterior we have interest in sampling from are the labels c_i for $i = 1, \dots, n$, the coefficients $\boldsymbol{\beta}_c$ for each unique label c, σ^2 and σ^2_β. In order to implement a Gibbs sampler algorithm, we compute the full conditionals for those variables.

We will use the following notation:

- m denotes the number of clusters, that is the number of unique labels
- $\boldsymbol{\beta}$ will be the collection of all the vectors of coefficients (that is, if $c^*_1, c^*_2, \dots, c^*_m$ denote the unique cluster labels, $\boldsymbol{\beta} = \{\boldsymbol{\beta}_{c^*_1}, \boldsymbol{\beta}_{c^*_2}, \dots, \boldsymbol{\beta}_{c^*_m}\}$)
- the pedix $-i$ to a vector denotes all the elements of that vector except the i-th
- n_c denotes the number of observations labelled with c
- $n_{-i,c}$ denotes the number of observations, excluding the i-th, that are labelled with c
- $\boldsymbol{y}_{i\cdot}$ denotes the vector y_{it} for $t = 1, \dots, T$
- $\boldsymbol{y}$ denotes the set of all the observations
- $\boldsymbol{H}$ denotes the matrix whose rows are the covariates $\boldsymbol{h}'_t$

Full conditional for cluster labels The full conditional for the labels can be obtained from a finite mixture model by integrating out the probabilities components weights and then taking the limit as the number of components tends to infinity (as for example in [2]), or equivalently by exploiting the Pólya urn representation of the Dirichlet process. Either way, the result is:

$$P(c_i = c \mid \boldsymbol{c}_{-i}, \boldsymbol{y}_{i\cdot}, \boldsymbol{\beta}, \sigma^2, \sigma^2_\beta) \propto n_{-i,c} F(\boldsymbol{y}_{i\cdot} | \boldsymbol{\beta}_c, \sigma^2), \quad \text{if } c = c_j \text{ for some } j \neq i \tag{7}$$

$$P(c_i \neq c_j \ \forall j \neq i | \boldsymbol{c}_{-i}, \boldsymbol{y}_i, \boldsymbol{\beta}, \sigma^2, \sigma^2_\beta) \propto \alpha_0 \int F(\boldsymbol{y}_{i\cdot} | \boldsymbol{\beta}, \sigma^2) dP_0(\boldsymbol{\beta} \mid \sigma^2_\beta) \tag{8}$$

Note that the previous relations do not identify a probability distribution on the integers (or any discrete set). However, since the value of c_i has no meaning in itself, but only in that it is equal to or different from the value of some other c_j, the two conditional probabilities of above are enough for the purpose of the algorithm. In practice, the actual value for c_i can be chosen for programming convenience.

In our case, the likelihood $F(\boldsymbol{y}_{i\cdot} | \boldsymbol{\beta}_c, \sigma^2)$ is the density of a T-dimensional Gaussian vector of mean $\boldsymbol{H}\boldsymbol{\beta}_c$ and variance $\sigma^2 \boldsymbol{I}$, and the base distribution P_0 of the Dirichlet process is the density of a p-dimensional Gaussian of mean $\boldsymbol{0}$ and variance $\sigma^2_\beta \boldsymbol{I}$. Therefore, thanks to conjugacy, we can explicitly rewrite the previous probabilities as:

$$P(c_i = c \mid \boldsymbol{c}_{-i}, \boldsymbol{y}_{i\cdot}, \boldsymbol{\beta}, \sigma^2, \sigma^2_\beta) \propto n_{-i,c} \exp\Big(- \frac{||\boldsymbol{y}_{i\cdot} - \boldsymbol{H}\boldsymbol{\beta}_c||^2}{2\sigma^2}\Big) \quad \text{if } c = c_j \text{ for some } j \neq i \tag{9}$$

$$P(c_i \neq c_j \; \forall j \neq i | \boldsymbol{c}_{-i}, \boldsymbol{y}_{i\cdot}, \boldsymbol{\beta}, \sigma^2, \sigma^2_\beta) \propto \alpha_0 \sqrt{\frac{|\boldsymbol{\Sigma}|}{(\sigma^2_\beta)^{2(2p+1)}}} \exp\Big(- \frac{||\boldsymbol{y}_{i\cdot}||^2}{2\sigma^2} + \frac{1}{2}\bar{\boldsymbol{\beta}}'\boldsymbol{\Sigma}^{-1}\bar{\boldsymbol{\beta}}\Big) \tag{10}$$

$$\boldsymbol{\Sigma} = \Big(\frac{\boldsymbol{H}'\boldsymbol{H}}{\sigma^2} + \frac{\boldsymbol{I}}{\sigma^2_\beta}\Big)^{-1} \tag{11}$$

$$\bar{\boldsymbol{\beta}} = \frac{\boldsymbol{\Sigma}\boldsymbol{H}'\boldsymbol{y}_{i\cdot}}{\sigma^2} \tag{12}$$

Full conditional for the $\boldsymbol{\beta}$ coefficients For each unique label c, the full conditional of $\boldsymbol{\beta}_c$ depends only on the observations $\boldsymbol{y}_{i\cdot}$ that are labelled with c. In our conjugate context it can be computed analytically, and there holds:

$$\boldsymbol{\beta}_c \mid \text{all else} \sim \mathcal{N}_p(\boldsymbol{\beta}^*, \boldsymbol{\Sigma}^*) \tag{13}$$

$$\boldsymbol{\Sigma}^* = \Big(\frac{n_c}{2\sigma^2}\boldsymbol{H}'\boldsymbol{H} + \frac{\boldsymbol{I}}{\sigma^2_\beta}\Big)^{-1} \tag{14}$$

$$\boldsymbol{\beta}^* = \frac{\boldsymbol{\Sigma}^*\boldsymbol{H}'\boldsymbol{S}(\boldsymbol{y}, c)}{\sigma^2} \tag{15}$$

$$\boldsymbol{S}(\boldsymbol{y}, c) = \sum_{i:c_i = c} \boldsymbol{y}_{i\cdot} \tag{16}$$

Full conditional for σ^2 The Inverse Gamma prior for σ^2 is conjugate to the model. Its full conditional is:

$$\sigma^2 \mid \text{all else} \sim \text{Inv-Gamma}(a_n, b_n) \tag{17}$$

$$a_n = a_0 + \frac{nT}{2} \tag{18}$$

$$b_n = b_0 + \frac{\sum_{i=1}^n ||\boldsymbol{y}_{i\cdot} - \boldsymbol{H}\boldsymbol{\beta}_{c_i}||^2}{2} \tag{19}$$

This result still holds true if we use the improper prior obtained by setting $a_0 = 0$, $b_0 = 0$.

Full conditional for σ^2_β The Inverse Gamma prior for σ^2_β is conjugate to the model. Its full conditional is:

$$\sigma^2_\beta \mid \text{all else} \sim \text{Inv-Gamma}(\nu_n, \kappa_n) \tag{20}$$

$$\nu_n = \nu_0 + \frac{m \cdot 2(2p+1)}{2} \tag{21}$$

$$\kappa_n = \kappa_0 + \frac{1}{2}\sum_{c=1}^m ||\boldsymbol{\beta}_c||^2 \tag{22}$$

This result still holds true if we use the improper prior obtained by setting $\nu_0 = 0$, $\kappa_0 = 0$.

Full conditional for the missing data To deal with missing data we can treat them as parameters and sample them within the Gibbs sampler loop. The full conditional of a single observation is:

$$y_{it} \mid \text{all else} \sim \mathcal{N}(h_t'\boldsymbol{\beta}_{c_i}, \sigma^2) \tag{23}$$

Therefore, at every iteration we will resample according to this distribution all the observations that are missing in the initial dataset.

5.2 The posterior similarity matrix

After every iteration $b \in \{1, \ldots, B\}$ in the Gibbs sampler, the observations are grouped into clusters by their labels $c_i^{(b)}$. This information can be encoded in a **similarity matrix**, that is a n-by-n symmetric matrix defined as:

$$\boldsymbol{S}_{ij} = \begin{cases} 1, & c_i = c_j \\ 0, & c_i \neq c_j \end{cases} \tag{24}$$

Taking the ergodic mean of the similarity matrices $\boldsymbol{S}_{ij}^{(b)}$ yields an estimate of the **posterior similarity matrix**, that is a matrix whose entries are the posterior probabilities of two observations belonging to the same cluster:

$$\boldsymbol{S}_{ij}^{post} = P(c_i = c_j \mid \boldsymbol{y}) \approx \frac{1}{B} \sum_{b=1}^{B} \boldsymbol{S}_{ij}^{(b)} \tag{25}$$

The posterior similarity matrix can be used to make inference on the clustering structure of the cells. This approach allows to deal quite naturally with label switching without fixing a priori the number of clusters. To infer a partition of the observations from the posterior similarity matrix, we minimize the posterior expectation of the following loss function (**Binder's loss function** [3], also discussed in [4]): denoting with $\boldsymbol{S}$ the "true" similarity matrix and with $\hat{\boldsymbol{S}}$ its estimate,

$$L(\boldsymbol{S}, \hat{\boldsymbol{S}}) = \sum_{i<j} \Big((1 - \hat{\boldsymbol{S}}_{ij})\boldsymbol{S}_{ij} + \hat{\boldsymbol{S}}_{ij}(1 - \boldsymbol{S}_{ij}) \Big) \tag{26}$$

The function penalizes differences between the "true" and the estimated posterior similarity matrices. The clustering structure we will deduce will be the one that minimizes the posterior loss, that is:

$$\boldsymbol{S}^* = \arg\min_{\hat{\boldsymbol{S}}} \mathbb{E}[L(\boldsymbol{S}, \hat{\boldsymbol{S}}) \mid \boldsymbol{y}] \tag{27}$$

5.3 Implementation of the algorithm

The Gibbs sampling algorithm can be summarized as follows: at each iteration,

```
for each cell $i = 1, \ldots, n$ do
    resample the label $c_i$ according to (9) and (10)
    if $n_{c_i,-i} = 0$ then
        resample the coefficient $\boldsymbol{\beta}_c$ according to (13) using as sample the sole observation $\boldsymbol{y}_i$.
    end if
end for
```

```
for each unique cluster label $c$ do
    resample the coefficient $\boldsymbol{\beta}_c$ according to (13)
end for
resample $\sigma^2$ according to (17)
resample $\sigma^2_\beta$ according to (20)
for each missing observation at cell $i$ and time $t$ do
    resample $y_{it}$ according to (23)
end for
```

The procedure is quite computationally intensive due to the dimension of the dataset and the high number of matrix operations involved. Because of this, we chose to implement the algorithm in C++, binding it to R by means of the R C++ API and the RCpp package [5]. Matrix operations in the C++ code are delegated to the Eigen library [6], that can be interfaced with RCpp using the package RCppEigen [7].

The C++ code exposes a single function, `fitAnovaDDP`, with the following signature and arguments:

```
fitAnovaDDP (y, h, posteriorDraws, burnInIterations, trim = 1, alpha0 = 1, a0 = 1,
             b0 = 1, nu0 = 1, kappa0 = 1)
```

- `y`: a n-by-T matrix with, on the i-th row, the vector of the observations for all times at cell i
- `h`: the n-by-2$(2p+1)$ design matrix
- `ndraws`, `burnin`, `trim`: the iteration parameters; the Gibbs sampler will run for a total of `burnin + ndraws` iterations, the first `burnin` iterations will be discarded and the remaining `ndraws` will be considered for the output, subsampling one iteration every `trim` to remove autocorrelation
- `alpha0`: the mass parameter for the Dirichlet process
- `a0,b0`: the prior hyperparameters for the distribution of σ^2
- `nu0,kappa0`: the prior hyperparameters for the distribution of σ^2_β

The function outputs a R list with the following members:

- `psm`: the posterior similarity matrix
- `beta`: a list that for each iteration has a matrix whose rows are the vectors $\boldsymbol{\beta}_c$ for each unique cluster label c
- `labels`: a matrix that has a row for each iteration, and a column for each observation, and in each entry has the label assigned to the corresponding observation at the corresponding iteration
- `sigma`: the vector of the values sampled from the posterior of σ^2
- `tauBeta`: the vector of the values sampled from the posterior of $\tau_\beta = \frac{1}{\sigma^2_\beta}$

The minimization of the Binder loss function was done using the function `minbinder` of the R package mcclust [8].

Drawing from a random multivariate normal Within the Gibbs sampler it is necessary to draw from multivariate normals of a given mean $\boldsymbol{\mu}$ and variance-covariance matrix Σ. In our case the variance-covariance matrix is defined in terms of the inverse of another matrix. Resorting to the R package mvtnorm for the generation of samples is therefore inefficient, because the `rmvnorm` function takes as argument the variance matrix, requiring the explicit computation of an inverse, which is computationally expensive.

Because of this, we implemented the following algorithm to sample from a multivariate normal given its mean $\boldsymbol{\mu}$ and the inverse of its variance matrix $Q = \Sigma^{-1}$:

```
draw $z_1,\ z_2\ \ldots\ z_n \overset{iid}{\sim} \mathcal{N}(0,1)$
define $\boldsymbol{z} = (z_1, z_2, \ldots, z_n)'$
compute the Cholesky factorization $Q = LL'$
set $\boldsymbol{x} = \boldsymbol{\mu} + L^{-1}\boldsymbol{z}$
```

This way, the generated vector $\boldsymbol{x}$ is multivariate normal of mean $\boldsymbol{\mu}$ and variance:

$$Var[\boldsymbol{X}] = (L^{-1})'Var[\boldsymbol{Z}]L^{-1} = (L^{-1})'L^{-1} = (LL')^{-1} = Q^{-1} = \Sigma \tag{28}$$

This approach spares the computation of an inverse and it only requires to compute a Cholesky factorization and solve the triangular system $L^{-1}\boldsymbol{z}$. Moreover, since the same matrix Q has to be used more than once, it is often possible to compute a single factorization for multiple samples, and this results in a significant improvement in efficiency.

The procedure has been implemented in the function `VectorXd rmvnorm2 ( const VectorXd &mu, const Eigen::LLT<MatrixXd> &SigmaInvLLT )`, that makes use of the Eigen library's classes both for vectors and for Cholesky factorization.

6 Results

We ran a simulation with 5000 iterations, with 500 burn-in iterations and trimming every 5 iterations to remove autocorrelation.

6.1 Goodness of fit

To verify that our simulation's results are consistent with the data, we estimated the predictive distribution of y_{it}. We report the results in figure 3 for three cells, coming from the periphery, the area around the center and the center respectively. We can see that the posterior expectation and the posterior credibility bands are consistent with the available data.

6.2 Posterior distribution of σ^2 and σ^2_β

In figures 4 and 5 we report the results of the posterior draws from σ^2 and σ^2_β. We can see that the algorithm appears to have reached convergence, and we can appreciate a sufficiently small autocorrelation between subsequent draws.

The plots also show histograms from the posterior distributions. We report in the following table the posterior mean and quantiles for such distributions:

	mean	$q_{0.25}$	$q_{0.50}$	$q_{0.75}$
σ^2	1.945554	1.944241	1.945293	1.946495
σ^2_β	0.000325514	0.000304638	0.000324085	0.000343622

6.3 Clustering of the cells

Finally, in figure 6 we reported the results of minimizing the Binder loss function given the posterior similarity matrix produced by the Gibbs sampler. The algorithm identifies a total of 5 clusters, four of which are of a significant size.

The three largest clusters (displayed in red, yellow and blue respectively in figure 6) correspond to the center, the near periphery (along with some of the principal roads) and the far periphery respectively. This is in accordance with what we expected from the exploratory analysis, as it appeared how the amplitude of the obscillation is correlated to the distance from the center. This observation is confirmed by 6b, in which the evolution over time of the Erlang number is plotted, colored according to the cluster attribution of each cell.

In general, we can appreciate how the clusters follow the features of the city (with reference in particular to the city center and some of the principal roads).

7 Conclusions and further developments

During our work we simulated the posterior of a non parametric harmonic regression model, used to explain the dependence on time of the Erlang number. We used the infinite mixture representation of the Dirichlet process to produce a posterior similarity matrix, thanks to which we could infer a clustering structure on the cells composing our dataset. We found such clustering structure to be coherent with the features of the city.

One possible extension to this work could be to take into account spatial effects in the model, that is to include correlation between nearby cells. We expect that such an addition could regularize the shape of the resulting clusters.

Moreover, it could be possible to experiment with different choices of priors for the hyperparameters, assuming for example a more general form for the variance-covariance matrix of the regression coefficients (which we assumed to be in the form $\sigma^2_\beta \boldsymbol{I}$).

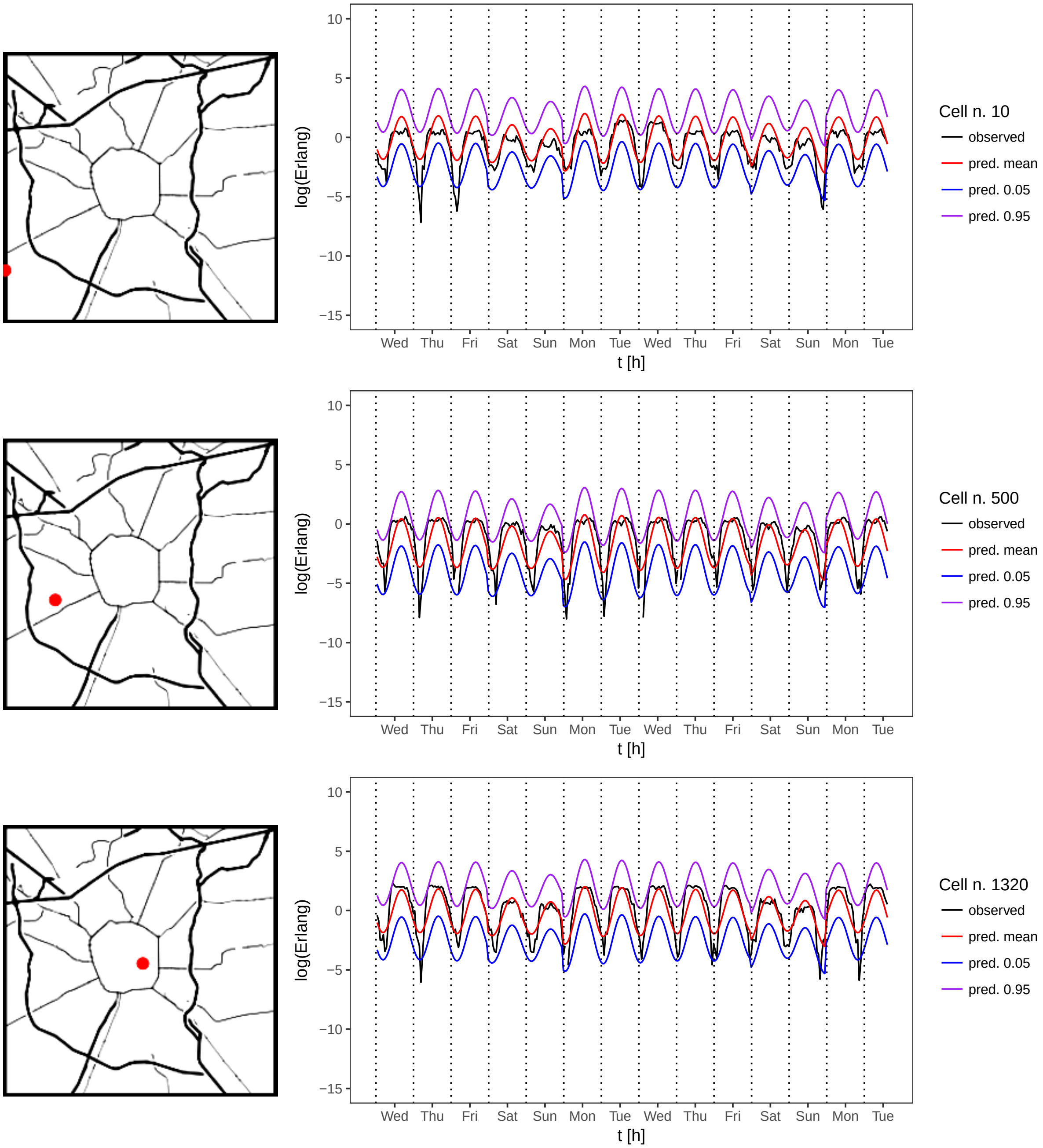


Figure 3: Predictive distribution of Y_{it} for three cells (whose location is displayed on the left)

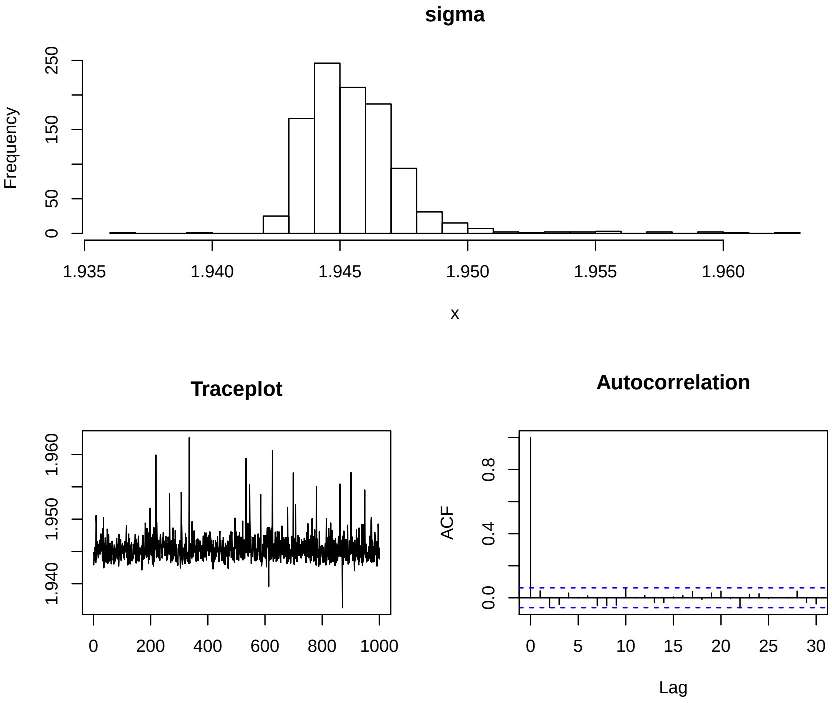


Figure 4: Summary of the posterior distribution for σ^2

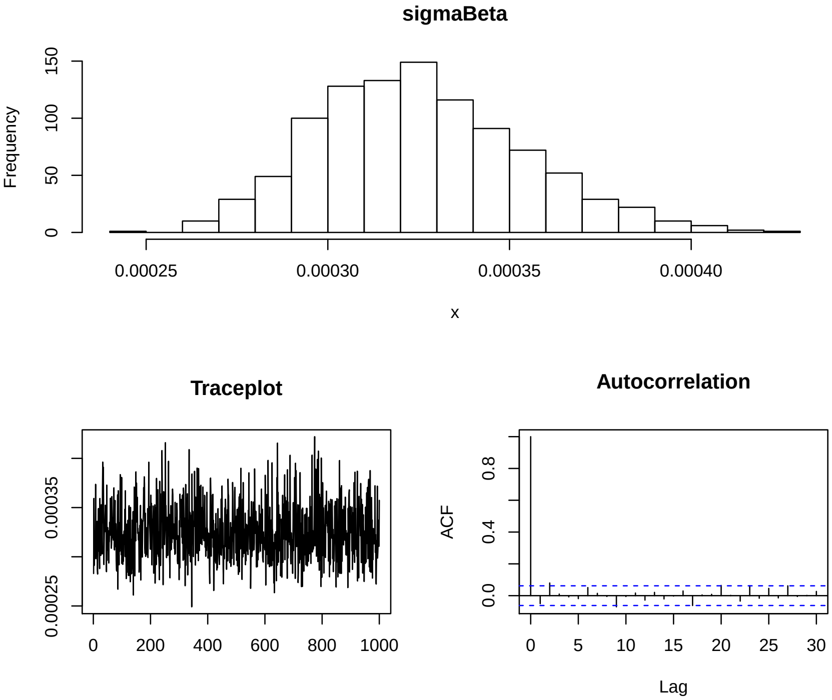
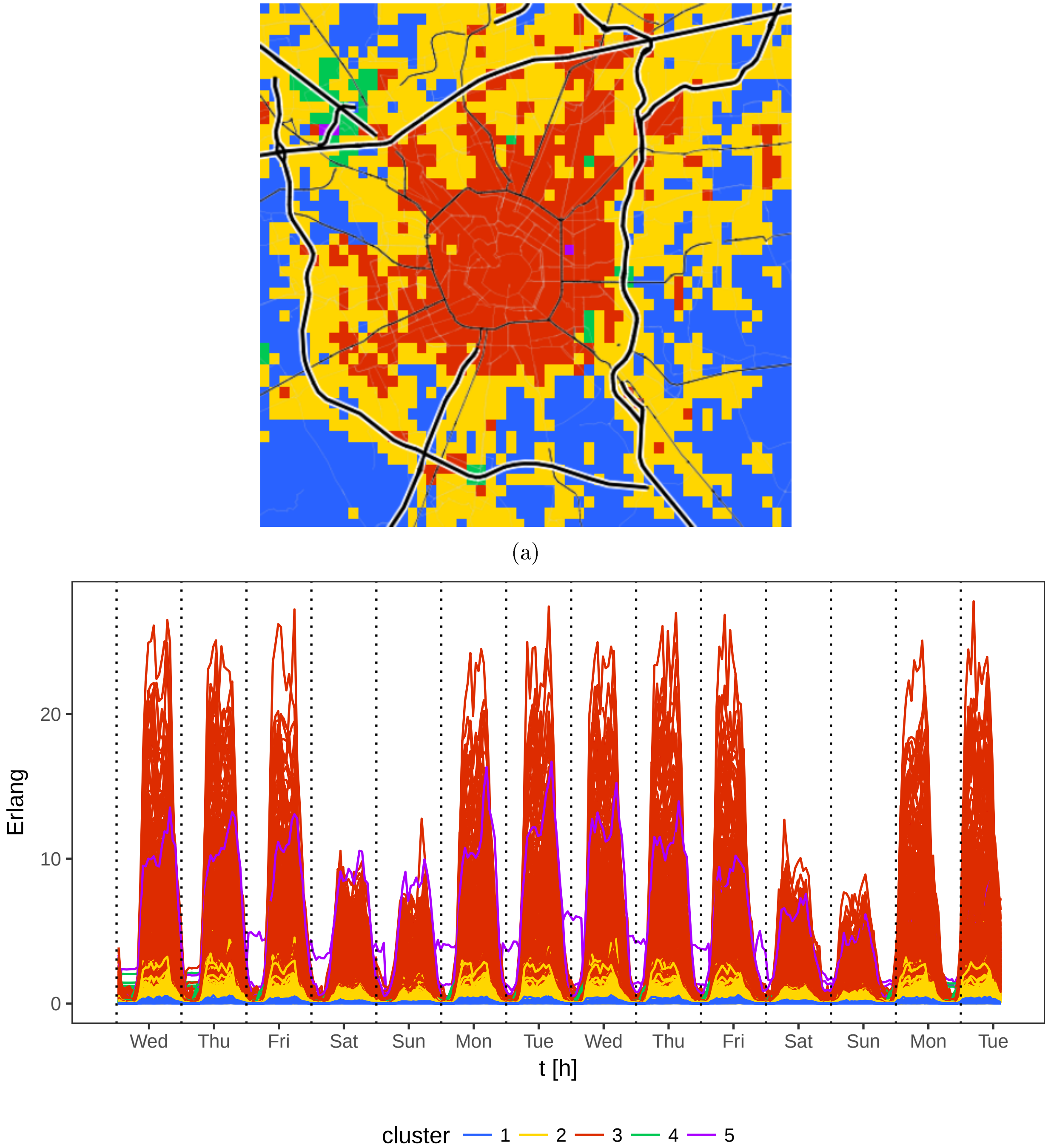


Figure 5: Summary of the posterior distribution for σ^2_β


Erlang
20
10
0
Wed
Thu
Fri
Sat
Sun
Mon
Tue
Wed
Thu
Fri
Sat
Sun
Mon
Tue
t [h]
cluster
1
2
3
4
5

(a)

(b)

Figure 6

8 References

[1] Maria De Iorio, Peter Müller, Gary L. Rosner, and Steven N. MacEachern. An ANOVA model for dependent random measures. *Journal of the American Statistical Association*, 99:205–215, 2004.

[2] Radford M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265, 2000.

[3] D. A. Binder. Bayesian cluster analysis. *Biometrika*, 65:31–38, 1978.

[4] A. Fritsch and K. Ickstadt. Improved criteria for clustering based on the posterior similarity matrix. *Bayesian Analysis*, 4:367–391, 2009.

[5] R package RCpp, `https://cran.r-project.org/web/packages/Rcpp/index.html`.

[6] Eigen, `http://eigen.tuxfamily.org/index.php`.

[7] R package RCppEigen, `https://cran.r-project.org/web/packages/RcppEigen/index.html`.

[8] R package mcclust, `https://cran.r-project.org/web/packages/mcclust/index.html`.