Species Distribution Modeling using Spatial Point Processes: a Case Study of Sloth Occurrence in Costa Rica

by Paula Moraga

Abstract

Species distribution models are widely used in ecology for conservation management of species and their environments. This paper demonstrates how to fit a log-Gaussian Cox process model to predict the intensity of sloth occurrence in Costa Rica, and assess the effect of climatic factors on spatial patterns using the **R-INLA** package. Species occurrence data are retrieved using **spocc**, and spatial climatic variables are obtained with **raster**. Spatial data and results are manipulated and visualized by means of several packages such as **raster** and **tmap**. This paper provides an accessible illustration of spatial point process modeling that can be used to analyze data that arise in a wide range of fields including ecology, epidemiology and the environment.

Introduction

Species distribution models are widely used in ecology to predict and understand spatial patterns, assess the influence of climatic and environmental factors on species occurrence, and identify rare and endangered species. These models are crucial for the development of appropriate strategies that help protect species and the environments where they live. In this paper, we demonstrate how to formulate spatial point processes for species distribution modeling and how to fit them with the **R-INLA** package (Rue et al., 2009) (`http://www.r-inla.org/`).

Point processes are stochastic models that describe locations of events of interest and possibly some additional information such as marks that inform about different types of events (Diggle, 2013; Moraga and Montes, 2011). These models can be used to identify patterns in the distribution of the observed locations, estimate the intensity of events (i.e., mean number of events per unit area), and learn about the correlation between the locations and spatial covariates. The simplest theoretical point process model is the homogeneous Poisson process. This process satisfies two conditions. First, the number of events in any region A follows a Poisson distribution with mean $\lambda|A|$, where λ is a constant value denoting the intensity and $|A|$ is the area of region A. And second, the number of events in disjoint regions are independent. Thus, if a point pattern arises as a realization of an homogeneous Poisson process, an event is equally likely to occur at any location within the study region, regardless of the locations of other events.

In many situations, the homogeneous Poisson process is too restrictive. A more interesting point process model is the log-Gaussian Cox process which is typically used to model phenomena that are environmentally driven (Diggle et al., 2013). A log-Gaussian Cox process is a Poisson process with a varying intensity which is itself a stochastic process of the form $\Lambda(s) = exp(Z(s))$ where $Z = \{Z(s) : s \in \mathbb{R}^2\}$ is a Gaussian process. Then, conditional on $\Lambda(s)$, the point process is a Poisson process with intensity $\Lambda(s)$. This implies that the number of events in any region A is Poisson distributed with mean $\int_A \Lambda(s)ds$, and the locations of events are an independent random sample from the distribution on A with probability density proportional to $\Lambda(s)$. The log-Gaussian Cox process model can also be easily extended to include spatial explanatory variables providing a flexible approach for describing and predicting a wide range of spatial phenomena.

In this paper, we formulate and fit a log-Gaussian Cox process model for sloth occurrence data in Costa Rica that incorporates spatial covariates that can influence the occurrence of sloths, as well as random effects to model unexplained variability. The model allows to estimate the intensity of the process that generates the data, understand the overall spatial distribution, and assess factors that can affect spatial patterns. This information can be used by decision-makers to develop and implement conservation management strategies.

The rest of the paper is organized as follows. First, we show how to retrieve sloth occurrence data using the **spocc** package (Chamberlain, 2018) and spatial climatic variables using the **raster** package (Hijmans, 2019). Then, we detail how to formulate the log-Gaussian Cox process and how to use **R-INLA** to fit the model. Then, we inspect the results and show how to obtain the estimates of the model parameters, and how to create create maps of the intensity of the predicted process. Finally, the conclusions are presented.

Sloth occurrence data

Sloths are tree-living mammals found in the tropical rain forests of Central and South America. They have an exceptionally low metabolic rate and are noted for slowness of movement. There are six sloth species in two families: two-toed and three-toed sloths. Here, we use the R package **spocc** (Chamberlain, 2018) to retrieve occurrence data of the three-toed brown-throated sloth in Costa Rica.

The **spocc** package provides functionality for retrieving and combining species occurrence data from many data sources such as the Global Biodiversity Information Facility (GBIF) (https://www.gbif.org/), and the Atlas of Living Australia (ALA) (https://www.ala.org.au/). We use the `occ()` function from **spocc** to retrieve the locations of brown-throated sloths in Costa Rica recorded between 2000 and 2019 from the GBIF database (GBIF.org, 2020; GBIF: The Global Biodiversity Information Facility, 2020). In the function, we specify arguments `query` with the species scientific name (*Bradypus variegatus*), `from` with the name of the database (GBIF), and `date` with the start and end dates (2000-01-01 to 2019-12-31). We also specify we wish to retrieve occurrences in Costa Rica by setting `gbifopts` to a named list with `country` equal to the 2-letter code of Costa Rica (CR). Moreover, we only retrieve occurrence data that have coordinates by setting `has_coords = TRUE`, and specify `limit` equal to 1000 to retrieve a maximum of 1000 occurrences.

```
library("spocc")
df <- occ(query = "Bradypus variegatus", from = "gbif",
          date = c("2000-01-01", "2019-12-31"),
          gbifopts = list(country = "CR"),
          has_coords = TRUE, limit = 1000)
```

`occ()` returns an object with slots for each of data sources. We can see the slot names by typing `names(df)`.

```
names(df)
```

```
## [1] "gbif"    "bison"   "inat"    "ebird"   "vertnet" "idigbio" "obis"    "ala"
```

In this case, since we only retrieve data from GBIF, the only slot with data is `df$gbif` while the others are empty. `df$gbif` contains information about the species occurrence and also other details about the retrieval process. We can use the `occ2df()` function to combine the output of `occ()` and create a single data frame with the most relevant information for our analysis, namely, the species name, the decimal degree longitude and latitude values, the data provider, and the dates and keys of the occurrence records.

```
d <- occ2df(df)
```

A summary of the data can be seen with `summary(d)`. We observe the data contain 375 locations of sloths occurred between 2000-02-01 and 2019-12-05.

```
summary(d)
```

```
##     name              longitude         latitude          prov
## Length:695         Min.   :-85.51   Min.   : 8.340   Length:695
## Class :character   1st Qu.:-84.15   1st Qu.: 9.391   Class :character
## Mode  :character   Median :-84.01   Median : 9.781   Mode  :character
##                    Mean   :-83.87   Mean   : 9.900
##                    3rd Qu.:-83.51   3rd Qu.:10.450
##                    Max.   :-82.62   Max.   :11.038
##      date                 key
## Min.   :2000-01-24   Length:695
## 1st Qu.:2014-01-13   Class :character
## Median :2017-05-30   Mode  :character
## Mean   :2015-12-31
## 3rd Qu.:2019-01-22
## Max.   :2019-12-30
```

We can visualize the locations of sloths retrieved in Costa Rica using several mapping packages such as **tmap** (Tennekes, 2018), **ggplot2** (Wickham, 2016), **leaflet** (Cheng et al., 2018), and **mapview** (Appelhans et al., 2019). Here, we choose to create maps using **tmap**. First, we use the `SpatialPoints()` function from the **sp** package (Pebesma and Bivand, 2005) to create a `SpatialPoints` object called `dpts` with the coordinates of the sloth locations.

```
library(sp)
dpts <- SpatialPoints(d[, c("longitude", "latitude")])
```

Then we create the map plotting the locations of `dpts`. **tmap** allows to create both static and interactive maps by using `tmap_mode("plot")` and `tmap_mode("view")`, respectively. Here, we create an interactive map using use a basemap given by the OpenStreetmap provider, and plot the sloth locations with `tm_shape(dpts) + tm_dots()`.

```
library(tmap)
tmap_mode("view")
tm_basemap(leaflet::providers$OpenStreetMap) +
  tm_shape(dpts) + tm_dots()
```

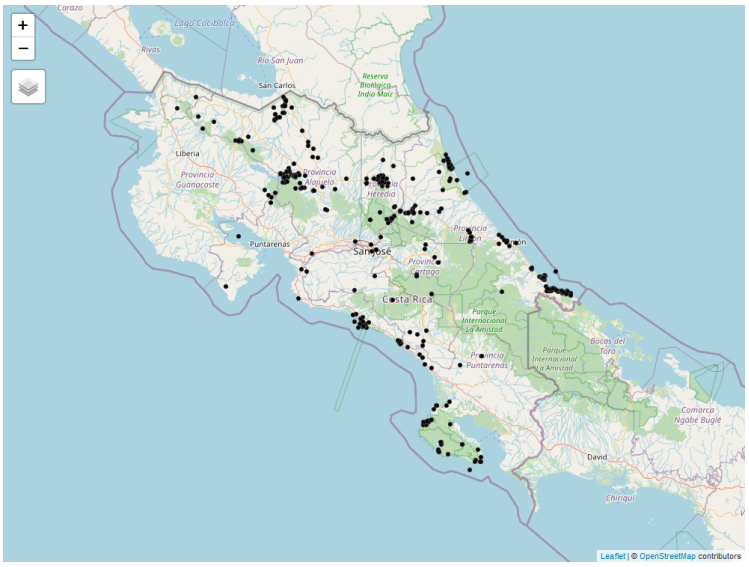

Figure 1: Snapshot of the interactive map depicting sloth locations in Costa Rica. The map shows some areas with no sloths and other areas with sloth aggregations.

The map created is shown in Figure 1. The map shows an inhomogeneous pattern of sloths with concentrations in several locations of Costa Rica. We will use a log-Gaussian Cox point process model to predict the intensity of the process that generates the sloth locations and assess the potential effect of climatic variables on the occurrence pattern.

Spatial climatic covariates

In the model, we include a spatial explanatory variable that can potentially affect sloth occurrence. Specifically, we include a variable that denotes annual minimum temperature observed in the study region. This variable can be obtained using the **raster** package (Hijmans, 2019) from the WorldClim database (`http://www.worldclim.org/bioclim`). We use the `getData()` function of the **raster** package by specifying the name of the database (`"worldclim"`), the variable name (`"tmin"`), and a resolution of 10 minutes of a degree (`"10"`). `getData()` returns a `RasterStack` with minimum temperature observations with degree Celsius x 10 units for each month. We average the values of the `RasterStack` and compute a raster that represents annual average minimum temperature.

```
library(raster)
rmonth <- getData(name = "worldclim", var = "tmin", res = 10)
rcov <- mean(rmonth)
```

Implementing and fitting the spatial point process model

Log-Gaussian Cox process model

We assume that the spatial point pattern of sloth locations in Costa Rica, $\{x_i : i = 1, \ldots, n\}$, has been generated as a realization of a log-Gaussian Cox process with intensity given by $\Lambda(s) = exp(\eta(s))$.

This model can be easily fitted by approximating it by a latent Gaussian model by means of a gridding approach (Illian et al., 2012). First, we discretize the study region into a grid with $n_1 \times n_2 = N$ cells $\{s_{ij}\}$, $i = 1, \ldots, n_1$, $j = 1, \ldots, n_2$. In the log-Gaussian Cox process, the mean number of events in cell s_{ij} is given by the integral of the intensity over the cell, $\Lambda_{ij} = \int_{s_{ij}} exp(\eta(s))ds$, and this integral can be approximated by $\Lambda_{ij} \approx |s_{ij}|exp(\eta_{ij})$, where $|s_{ij}|$ is the area of the cell s_{ij}. Then, conditional on the latent field η_{ij}, the observed number of locations in grid cell s_{ij}, y_{ij}, are independent and Poisson distributed as follows,

$$y_{ij}|\eta_{ij} \sim Poisson(|s_{ij}|exp(\eta_{ij})).$$

In our example, we model the log-intensity of the Poisson process as

$$\eta_{ij} = \beta_0 + \beta_1 \times cov(s_{ij}) + f_s(s_{ij}) + f_u(s_{ij}).$$

Here, β_0 is the intercept, $cov(s_{ij})$ is the covariate value at s_{ij}, and β_1 is the coefficient of $cov(s_{ij})$. `f_s()` is a spatially structured random effect reflecting unexplained variability that can be specified as a second-order two-dimensional CAR-model on a regular lattice. `f_u()` is an unstructured random effect reflecting independent variability in cell s_{ij}.

Computational grid

In order to fit the model, we create a regular grid that covers the region of Costa Rica. First, we obtain a map of Costa Rica using the `ne_countries()` function of the **rnaturalearth** package (South, 2017). In the function we set `type = "countries"`, `country = "Costa Rica"` and `scale = "medium"` (`scale` denotes the scale of map to return and possible options are small, medium and large).

```
library(rnaturalearth)
map <- ne_countries(type = "countries", country = "Costa Rica", scale = "medium")
```

Then, we create a raster tha covers Costa Rica using `raster()` where we provide the map Costa Rica and set `resolution = 0.1` to create cells with size of 0.1 decimal degrees. This creates a raster with $31 \times 33 = 1023$ cells, each having an area equal to 0.1^2 decimal degrees^2 (or 11.132 Km^2 at the equator).

```
resolution <- 0.1
r <- raster(map, resolution = resolution)
(nrow <- nrow(r))
## [1] 31
(ncol <- ncol(r))
## [1] 33
nrow*ncol
## [1] 1023
```

We initially set to 0 the values of all the raster cells by using `r[] <-0`. Then, we use `cellFromXY()` to obtain the number of sloths in each of the cells, and assign these counts to each of the cells of the raster.

```
r[] <- 0
tab <- table(cellFromXY(r, dpts))
r[as.numeric(names(tab))] <- tab
```

Finally, we convert the raster `r` to a `SpatialPolygonsDataFrame` object called `grid` using `rasterToPolygons()`. This grid will be used to fit the model with the **R-INLA** package.

```
grid <- rasterToPolygons(r)
```

Data

Now, we add to `grid` the data needed for modeling. Specifically, we add variables `id` with the id of the cells, `Y` with the number of sloths, and `cellarea` with the cell areas.

```
grid$id <- 1:nrow(grid)
grid$Y <- grid$layer
grid$cellarea <- resolution*resolution
```

We also add a variable `cov` with the value of the minimum temperature covariate in each of the cells obtained with the `extract()` function of **raster**.

```
grid$cov <- extract(rcov, coordinates(grid))
```

Finally, we delete the cells of `grid` that lie outside Costa Rica. First, we use `raster::intersect()` to know which cells lie within the map, and then subset these cells in the `grid` object.

```
gridmap <- raster::intersect(grid, map)
grid <- grid[grid$id %in% gridmap$id, ]
```

A summary of the data can be seen as follows,

```
summary(grid)
```

```
## Object of class SpatialPolygonsDataFrame
## Coordinates:
##          min       max
## x -85.908008 -82.60801
## y   8.089453  11.18945
## Is projected: FALSE
## proj4string :
## [+proj=longlat +datum=WGS84 +no_defs +ellps=WGS84 +towgs84=0,0,0]
## Data attributes:
##      layer              id               Y             cellarea         cov
##  Min.   : 0.000   Min.   :   3.0   Min.   : 0.000   Min.   :0.01   Min.   : 78.0
##  1st Qu.: 0.000   1st Qu.: 233.5   1st Qu.: 0.000   1st Qu.:0.01   1st Qu.:176.7
##  Median : 0.000   Median : 387.0   Median : 0.000   Median :0.01   Median :202.4
##  Mean   : 1.356   Mean   : 415.5   Mean   : 1.356   Mean   :0.01   Mean   :189.0
##  3rd Qu.: 0.000   3rd Qu.: 557.5   3rd Qu.: 0.000   3rd Qu.:0.01   3rd Qu.:211.2
##  Max.   :93.000   Max.   :1021.0   Max.   :93.000   Max.   :0.01   Max.   :223.2
##                                                                    NA's   :2
```

We observe that the minimum temperature covariate has 2 missing values. We decide to impute these missing values with a simple approach where we set these values equal to the values of the cells next to them.

```
indNA <- which(is.na(grid$cov))
indNA
```

```
## [1] 206 388
```

```
grid$cov[indNA] <- grid$cov[indNA+1]
```

We use **tmap** to create maps of the number of sloths (`Y`) and the covariate values (`cov`). In the maps, we plot the border of `grid` that we obtain with the `gUnaryUnion()` function of the **rgeos** package (Bivand and Rundel, 2019).

```
library(rgeos)
gridborder <- gUnaryUnion(grid)
```

We use `tm_facets(ncol = 2)` to plot maps in the same row and two columns, and `tm_legend()` to put the legends in the left-bottom corner of the plots (Figure 2).

```
tmap_mode("plot")
tm_shape(grid) +
  tm_polygons(col = c("Y", "cov"), border.col = "transparent") +
  tm_shape(gridborder) + tm_borders() +
  tm_facets(ncol = 2) + tm_legend(legend.position = c("left", "bottom"))
```

Fitting the model using R-INLA

We fit the log-Gaussian Cox process model to the sloths data using the **R-INLA** package. This package implements the integrated nested Laplace approximation (INLA) approach that permits to perform approximate Bayesian inference in latent Gaussian models (Rue et al., 2009; Moraga., 2019). **R-INLA** is not on CRAN because it uses some external C libraries that make difficult to build the binaries. Therefore, when installing the package, we need to specify the URL of the R-INLA repository. We also need to add the `https://cloud.r-project.org` repository to enable the installation of CRAN dependencies as follows,

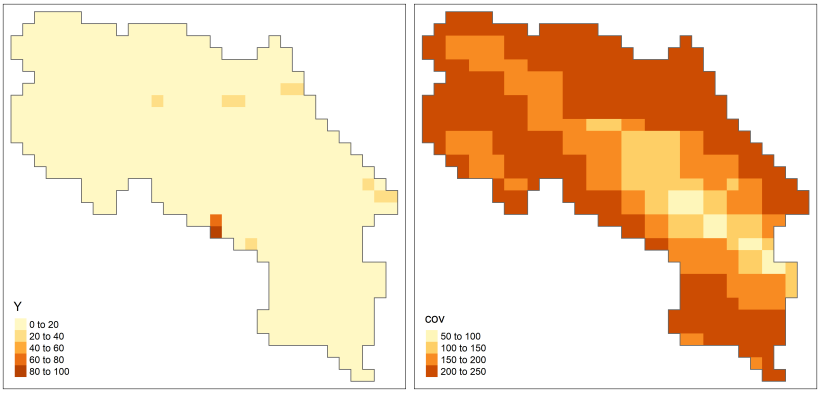

Figure 2: Maps with the number of sloths (left) and minimum temperature values (right) in Costa Rica. The intensity of sloths occurrence is modeled using minimum temperature as a covariate.

```
install.packages("INLA", repos = c("https://inla.r-inla-download.org/R/stable",
                                   "https://cloud.r-project.org"), dep = TRUE)
```

Note that the **R-INLA** package is large and its installation may take a few minutes. Moreover, **R-INLA** suggests the **graph** and **Rgraphviz** packages which are part of the Bioconductor project. These packages have to be installed by using their tools, for example, by using `BiocManager::install(c("graph","Rgraphviz"),dep = TRUE))`.

To fit the model in INLA we need to specify a formula with the linear predictor, and then call the `inla()` function providing the formula, the family, the data, and other options. The formula is written by writing the outcome variable, then the $\sim$ symbol, and then the fixed and random effects separated by `+` symbols. By default, the formula includes an intercept. The outcome variable is `Y` (the number of occurrences in each cell) and the covariate is `cov`. The random effects are specified with the `f()` function where the first argument is an index vector specifying which elements of the random effect apply to each observation, and the other arguments are the model name and other options. In the formula, different random effects need to have different indices vectors. We use `grid$id` for the spatially structured effect, and create an index vector `grid$id2` with the same values as `grid$id` for the unstructured random effect. The spatially structured random effect is specified with the index vector `id`, the model name that corresponds to ICAR(2) (`"rw2d"`), and the number of rows (`nrow`) and columns (`ncol`) of the regular lattice. The unstructured random effect is specified with the index vector `id2` and the model name `"iid"`.

```
library(INLA)

grid$id2 <- grid$id

formula <- Y ~ 1 + cov +
  f(id, model="rw2d", nrow = nrow, ncol = ncol) +
  f(id2, model="iid")
```

Finally, we call `inla()` where we provide the formula, the family (`"poisson"`) and the data (`grid@data`). We write `E = cellarea` to denote that the expected values in each of the cells are in variable `cellarea` of the data. We also write `control.predictor = list(compute = TRUE)` to compute the marginal densities for the linear predictor.

```
res <- inla(formula, family = "poisson", data = grid@data,
            E = cellarea, control.predictor = list(compute = TRUE))
```

Results

The execution of `inla()` returns an object `res` that contains information about the fitted model including the posterior marginals of the parameters and the intensity values of the spatial process. We can see a summary of the results as follows,

```
summary(res)
```

```
## Fixed effects:
##               mean    sd 0.025quant 0.5quant 0.975quant   mode kld
## (Intercept) -2.810 1.477     -5.877   -2.752     -0.071 -2.638   0
## cov          0.022 0.007      0.009    0.022      0.035  0.021   0
##
## Random effects:
##   Name          Model
##     id Random walk 2D
##    id2 IID model
##
## Model hyperparameters:
##                     mean    sd 0.025quant 0.5quant 0.975quant  mode
## Precision for id  0.423 0.278       0.13    0.348      1.152 0.250
## Precision for id2 0.246 0.050       0.16    0.241      0.358 0.233
##
## Expected number of effective parameters(stdev): 221.05(9.55)
## Number of equivalent replicates : 2.31
##
## Marginal log-Likelihood:  -1648.63
## Posterior marginals for the linear predictor and the fitted values are computed
```

The intercept $\hat{\beta}_0 = -2.810$ with 95% credible interval $(-5.877, -0.071)$, the minimum temperature covariate has a positive effect on the intensity of the process with a posterior mean $\hat{\beta}_1 = 0.022$ and 95% credible interval $(0.009, 0.035)$. We can plot the posterior distribution of the coefficient of the covariate $\hat{\beta}_1$ with **ggplot2** (Figure 3). First, we calculate a smoothing of the marginal distribution of the coefficient with `inla.smarginal()` and then call `ggplot()` specifying the data frame with the marginal values.

```
library(ggplot2)
marginal <- inla.smarginal(res$marginals.fixed$cov)
marginal <- data.frame(marginal)
ggplot(marginal, aes(x = x, y = y)) + geom_line() +
  labs(x = expression(beta[1]), y = "Density") +
  geom_vline(xintercept = 0, col = "black") + theme_bw()
```

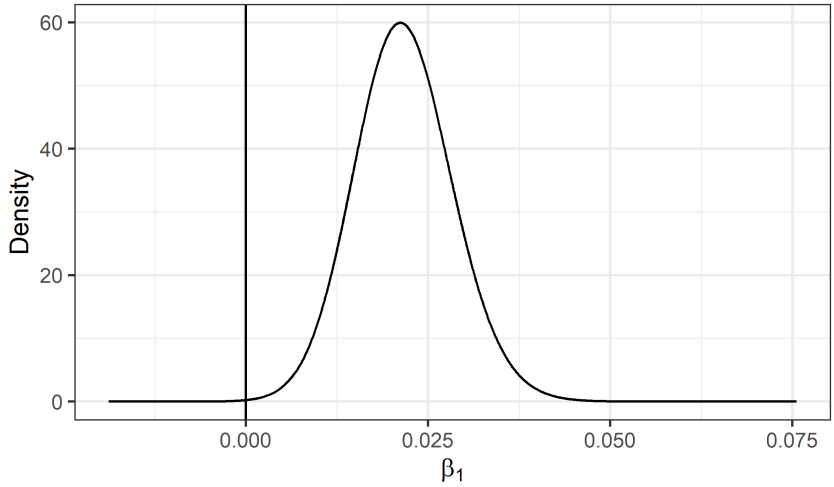

Figure 3: Posterior distribution of the coefficient of covariate minimum temperature. A vertical line at 0 is depicted and it can be seen that the posterior mean is positive and the 95% credible interval does not include 0.

The estimated spatially structured effect is in `res$summary.random$id`. This object contains 1023 elements that correspond to the number of cells in the regular lattice. We can add to the `grid` object the posterior mean of the spatial effect corresponding to each of the cells in Costa Rica as follows,

```
grid$respa <- res$summary.random$id[grid$id, "mean"]
```

We can also obtain the posterior mean of the unstructured random effect as follows,

```
grid$reiid <- res$summary.random$id2[, "mean"]
```

Then we can create maps of the random effects with **tmap**.

```
tm_shape(grid) +
  tm_polygons(col = c("respa", "reiid"), style = "cont", border.col = "transparent")  +
  tm_shape(gridborder) + tm_borders() +
  tm_facets(ncol = 2) + tm_legend(legend.position = c("left", "bottom"))
```

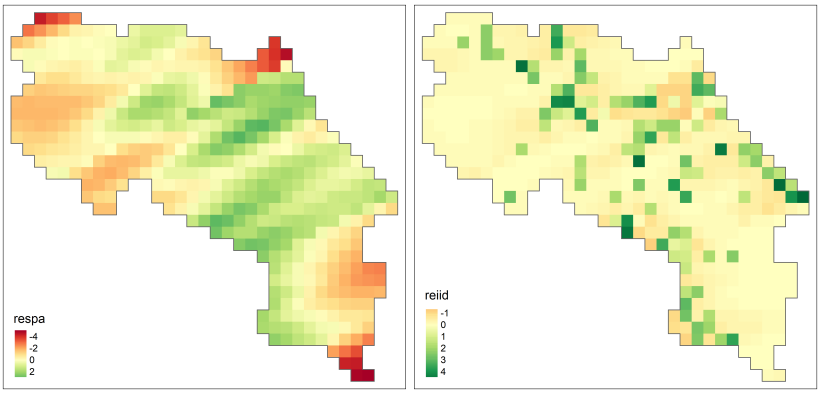

Figure 4: Maps with the values of the spatially structured (left) and unstructured (right) random effects. Maps show there is spatially structured and unstructured residual variation.

Figure 4 shows the maps of the spatially structured and unstructured random effects. We observe a non-constant pattern of the spatially structured random effect suggesting that the intensity of the process that generates the sloth locations may be affected by other spatial factors that have not been considered in the model. Morevoer, the unstructured random effect shows several locations with high values that modify the intensity of the process in individual cells independently from the rest.

The mean and quantiles of the predicted intensity (mean number of events per unit area) in each of the grid cells are in `res$summary.fitted.values`. In the object `grid`, we add a variable `NE` with the mean number of events of each cell by assigning the predicted intensity multiplied by the cell areas. We also add variables `LL` and `UL` with the lower and upper limits of 95% credible intervals for the number of events by assigning quantiles 0.025 and 0.975 multiplied by the cell areas.

```
cellarea <- resolution*resolution
grid$NE <- res$summary.fitted.values[, "mean"] * cellarea
grid$LL <- res$summary.fitted.values[, "0.025quant"] * cellarea
grid$UL <- res$summary.fitted.values[, "0.975quant"] * cellarea
```

We use **tmap** to create maps with the mean and lower and upper limits of 95% credible intervals for the number of sloths in each of the cells. We plot the three maps with a common legend that has breaks from 0 to the maximum number of cases in `grid$UL`.

```
tm_shape(grid) +
  tm_polygons(col = c("NE", "LL", "UL"),
              style = 'fixed', border.col = "transparent",
              breaks = seq(0, ceiling(max(grid$UL)), 10)) +
  tm_shape(gridborder) + tm_borders() +
  tm_facets(ncol = 3) + tm_legend(legend.position = c("left", "bottom"))
```

Maps created are shown in Figure 5. We observe that overall, the intensity of sloth occurrence is low, with less than 10 sloths in each of the cells. We also see there are some locations of high sloth intensity in the west and east coasts and the north of Costa Rica. The maps with the lower and upper limits of 95% credible intervals denote the uncertainty of these predictions. The maps created inform about the spatial patterns in the period where the data were collected. In addition, maps of the sloth numbers over time can also be produced using spatio-temporal point process models and this would help understand spatio-temporal patterns. The modeling results can be useful for decision-makers to identify areas of interest for conservation management strategies.

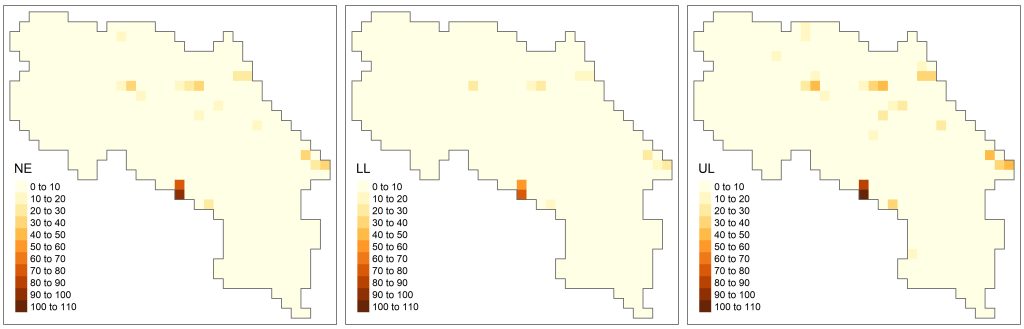

Figure 5: Maps with the predicted mean number of sloths (left), and lower (center) and upper (right) limits of 95% credible intervals. Maps show low intensity of sloth occurrence overall, and some specific locations with high intensity.

Summary

Species distribution models are widely used in ecology for conservation management of species and their environments. In this paper, we have described how to develop and fit a log-Gaussian Cox process model using the **R-INLA** package to predict the intensity of species occurrence, and assess the effect of spatial explanatory variables. We have illustrated the modeling approach using sloth occurrence data in Costa Rica retrieved from the Global Biodiversity Information Facility database (GBIF) using **spocc**, and a spatial climatic variable obtained with **raster**. We have also shown how to examine and interpret the results including the estimates of the parameters and the intensity of the process, and how to create maps of variables of interest using **tmap**.

Statistical packages such as Stan (Carpenter et al., 2017) or JAGS (Plummer, 2003) could have been used instead of **R-INLA** to fit our data. However, these packages use Markov chain Monte Carlo (MCMC) algorithms and may be high computationally demanding and become infeasable in large spatial data problems. In contrast, INLA produces faster inferences which allows us to fit large spatial datasets and explore alternative models.

The objective of this paper is to illustrate how to analyze species occurrence data using spatial point process models and cutting-edge statistical techniques in R. Therefore, we have ignored the data collection methods and have assumed that the spatial pattern analyzed is a realization of the true underlying process that generates the data. In real investigations, however, it is important to understand the sampling mechanisms, and assess potential biases in the data such as overrepresentation of certain areas that can invalidate inferences. Ideally, we would analyze data that have been obtained using well-defined sampling schemes. Alternatively, we would need to develop models that adjust for biases in the data to produce meaningful results (Giraud et al., 2015; Dorazio, 2014; Fithian et al., 2015). Moreover, expert knowledge is crucial to be able to develop appropriate models that include important predictive covariates and random effects that account for different types of variability.

To conclude, this paper provides an accessible illustration of spatial point process models and computational approaches that can help non-statisticians analyze spatial point patterns using R. We have shown how to use these approaches in the context of species distribution modeling, but they are also useful to analyze spatial data that arise in many other fields such as epidemiology and the environment.

Bibliography

T. Appelhans, F. Detsch, C. Reudenbach, and S. Woellauer. *mapview: Interactive Viewing of Spatial Data in R*, 2019. URL https://CRAN.R-project.org/package=mapview. R package version 2.7.0. [p2]

R. Bivand and C. Rundel. *rgeos: Interface to Geometry Engine - Open Source ('GEOS')*, 2019. URL https://CRAN.R-project.org/package=rgeos. R package version 0.4-3. [p5]

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 76 (1), 2017. doi: https://doi.org/10.18637/jss.v076.i01. [p9]

S. Chamberlain. *spocc: Interface to Species Occurrence Data Sources*, 2018. URL https://CRAN.R-project.org/package=spocc. R package version 0.9.0. [p1, 2]

J. Cheng, B. Karambelkar, and Y. Xie. *leaflet: Create Interactive Web Maps with the JavaScript 'Leaflet' Library*, 2018. URL https://CRAN.R-project.org/package=leaflet. R package version 2.0.2. [p2]

P. J. Diggle. *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. Chapman & Hall/CRC, 2013. [p1]

P. J. Diggle, P. Moraga, B. Rowlingson, and B. M. Taylor. Spatial and Spatio-Temporal Log-Gaussian Cox Processes: Extending the Geostatistical Paradigm. *Statistical Science*, 28(4):542–563, 2013. URL https://doi.org/10.1214/13-STS441. [p1]

R. M. Dorazio. Accounting for imperfect detection and survey bias in statistical analysis of presence-only data. *Global Ecology and Biogeography*, 23, 2014. doi: https://doi.org/10.1111/geb.12216. [p9]

W. Fithian, J. Elith, T. Hastie, and D. A. Keith. Bias correction in species distribution models: pooling survey and collection data for multiple species. *Methods in Ecology and Evolution*, 6(4):428–438, 2015. doi: https://doi.org/10.1111/2041-210X.12242. [p9]

GBIF: The Global Biodiversity Information Facility. What is GBIF?, 2020. URL https://www.gbif.org/what-is-gbif. Accessed on 2 October 2020. [p2]

GBIF.org. GBIF Home Page, 2020. URL https://www.gbif.org. Accessed on 2 October 2020. [p2]

C. Giraud, C. Calenge, C. Coron, and R. Julliard. Capitalizing on opportunistic data for monitoring relative abundances of species. *Biometrics*, 72, 2015. doi: https://doi.org/10.1111/biom.12431. [p9]

R. J. Hijmans. *raster: Geographic Data Analysis and Modeling*, 2019. URL https://CRAN.R-project.org/package=raster. R package version 2.9-5. [p1, 3]

J. B. Illian, S. H. Sorbye, H. Rue, and D. Hendrichsen. Using INLA To Fit A Complex Point Process Model With Temporally Varying Effects - A Case Study. *Journal of Environmental Statistics*, 3, 2012. [p4]

P. Moraga. *Geospatial Health Data: Modeling and Visualization with R-INLA and Shiny*. Chapman & Hall/CRC Biostatistics Series, 2019. [p5]

P. Moraga and F. Montes. Detection of spatial disease clusters with LISA functions. *Statistics in Medicine*, 30:1057–1071, 2011. URL https://doi.org/10.1002/sim.4160. [p1]

E. J. Pebesma and R. S. Bivand. Classes and methods for spatial data in R. *R News*, 5, 2005. URL https://cran.r-project.org/doc/Rnews/. [p2]

M. Plummer. JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, 2003. [p9]

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian Inference for Latent Gaussian Models Using Integrated Nested Laplace Approximations (with discussion). *Journal of the Royal Statistical Society B*, 71:319–392, 2009. URL https://doi.org/10.1111/j.1467-9868.2008.00700.x. [p1, 5]

A. South. *rnaturalearth: World Map Data from Natural Earth*, 2017. URL https://CRAN.R-project.org/package=rnaturalearth. R package version 0.1.0. [p4]

M. Tennekes. tmap: Thematic Maps in R. *Journal of Statistical Software*, 84(6):1–39, 2018. doi: 10.18637/jss.v084.i06. [p2]

H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL https://ggplot2.tidyverse.org. [p2]

Paula Moraga
Computer, Electrical and Mathematical Sciences and Engineering Division
King Abdullah University of Science and Technology (KAUST)
Thuwal, 23955-6900
Saudi Arabia
ORCiD: 0000-0001-5266-0201
Webpage: http://www.paulamoraga.com/
paula.moraga@kaust.edu.sa