Example of a spatial item response model

Edgar Santos-Fernandez[*] Kerrie Mengersen [†]

4/16/2021

In this simple example, we illustrate the fit of the spatial item response model proposed in the paper: **"Understanding the reliability of citizen science observational data using item response models"**.

This is done using the `ir_spat` function from the R package `staircase`. We use the gold standard dataset, but with a reduced number of participants and images for demonstration purposes only. We should keep in mind that these item response models require a large number of classifications per item to obtain suitable estimates of the parameters.

Loading the libraries we will use:

```r
library('Rcpp')
library('dplyr')
library('ggplot2')
library('rstan')
library('bayesplot')
library('viridis')
library('ggrepel')
library('RColorBrewer')
library('coda')
library('rgeos')
library('dismo')
library('ggvoronoi')
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

Let us start downloading and reading the gold standard dataset:

```r
download.file("https://raw.github.com/EdgarSantos-Fernandez/hakuna/main/serengety_gs.rds", "seren.rds")
data <- readRDS("seren.rds")
```

We select only 30 participants to fit the model so that it does not take too much running time.

```r
seed <- 202105
set.seed(seed)
```

```r
# NB: remove the all the lines in this chunk if you want to run the model using the whole dataset
data <- data[data$CaptureEventID %in% names(table(data$CaptureEventID)[(table(data$CaptureEventID) > 30
data <- data[data$user %in% names(table(data$user)[(table(data$user) > 30) ]),]
data <- data[data$user %in% unique(data$user)[1:30],]
```

[*] Queensland University of Technology, edgar.santosfdez@gmail.com, santosfe@qut.edu.au

[†] Queensland University of Technology, k.mengersen@qut.edu.au

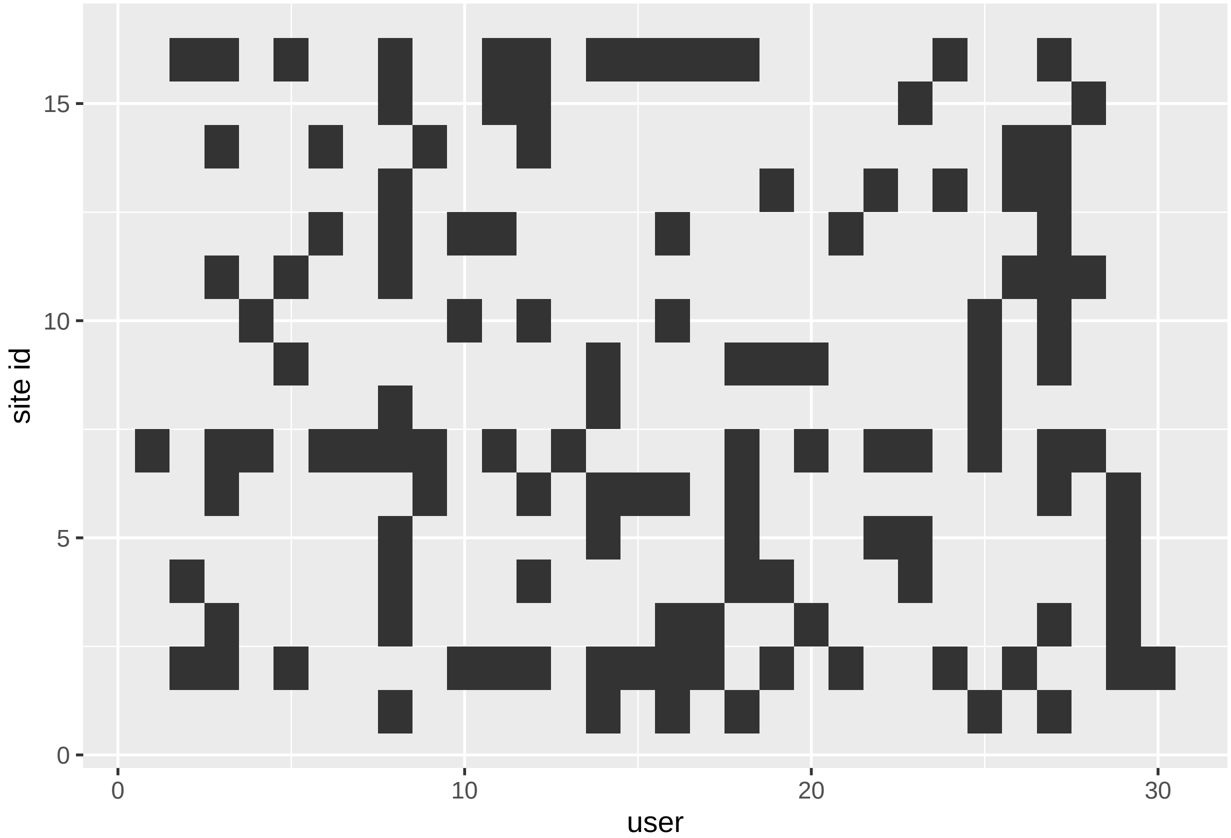
```
images <- sort(table(data$CaptureEventID), decreasing = T)
images <- names(images[images > 10])
data <- data[data$CaptureEventID %in% images,]
```

We then create unique ids for the sites and users:

```
# creating a unique site id
data$id <- as.numeric(factor(data$id))
# creating a unique user id
data$user <- as.numeric(factor(data$user))
```

Depicting the working dataset composed of 16 sites and 30 users:

```
ggplot(data) + geom_raster(aes(y= id, x = user)) + ylab('site id')
```



Then using the function `ir_spat`, we fit a one-parameter spatial item response model (1PLUS), which involves abilities and difficulties associated with the species and spatial locations. However, it does not includes guessing and slope parameters. See, more details on the `ir_spat` function help.
We use a exponential covariance matrix (spat_model = 'exp').

We consider two covariates: animal moving and the presence of babies on images, which are expected to affect the difficulty of the task.

```
# To run this chunk use eval = T.

```

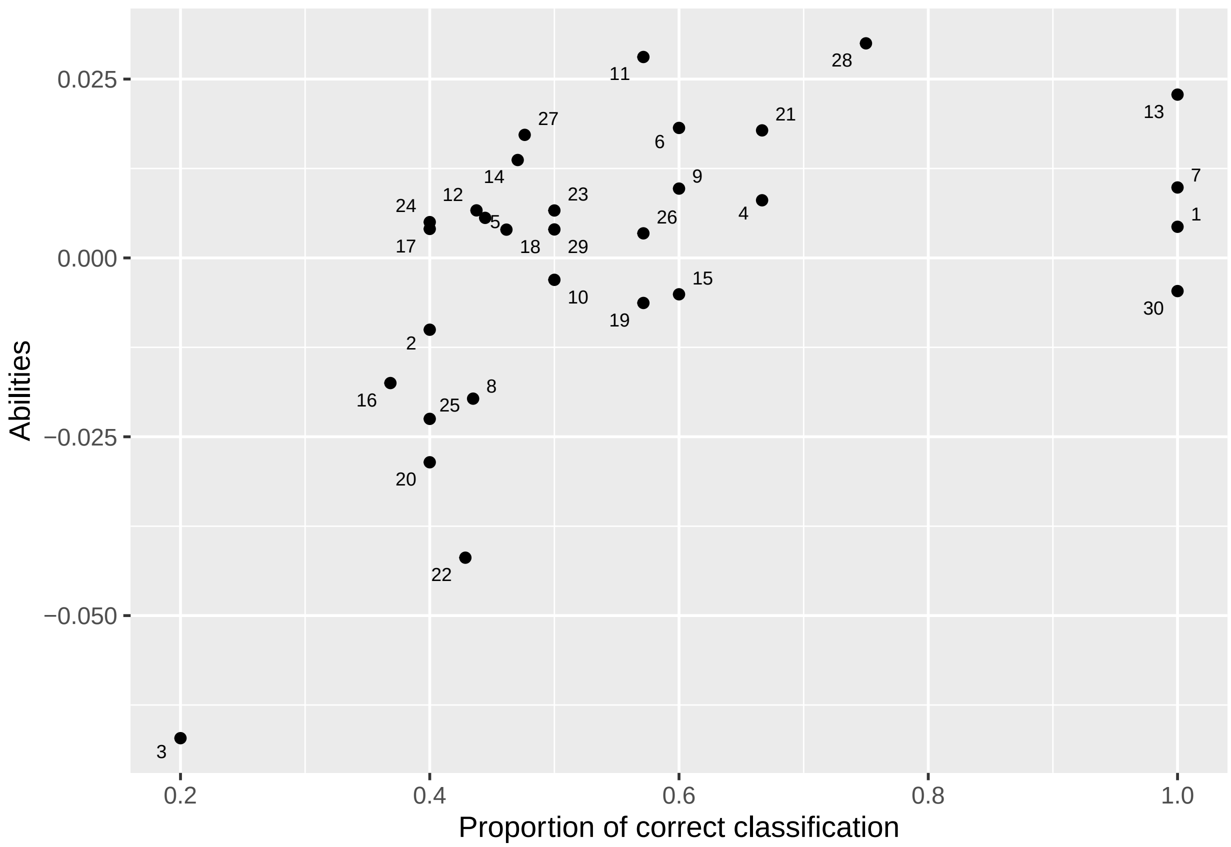
```
fit <- ir_spat(formula = site ~ -1 + Moving + Babies, # covariates affecting the difficulty
               data = data, # a data frame
               spat_model = 'exp', # spatial covariance matrix
               itemtype = '1PLUS', # item response model
               abil = 'user', # participants ids
               diff = 'id', # location id
               y = 'correct', # binary response variable
               coords = c("LocationX", "LocationY"), # coordinates
               iter = 8000,
               warmup = 4000,
               chains = 3,
               refresh = 100,
               seed = seed
)
```

Computing a summary statistics:

```
stats <- summary(fit)
stats <- stats$summary
array <- as.array(fit)
```

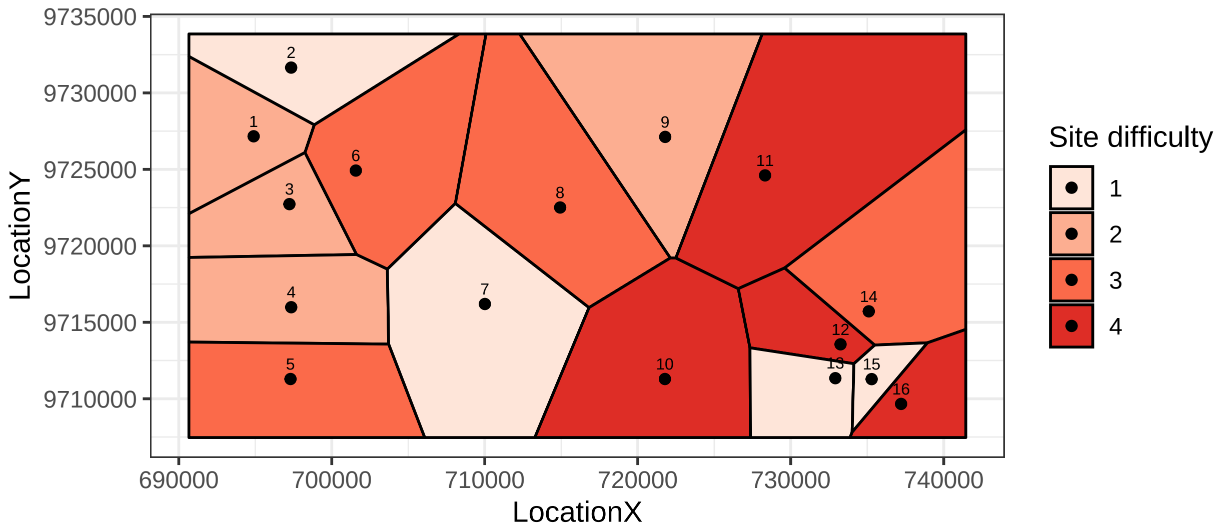
Now we can compare the participants' abilities:

```
abils <- stats[grep('abil\\[', rownames(stats)),]
abils <- cbind(abils, data %>% group_by(user) %>% summarize(ns = n(), prop = mean(correct)) %>% data.fra

names(abils)[grep('%', names(abils))] <- c('q2.5','q25', 'q50', 'q75', 'q97.5')

 ggplot(abils, aes(x= prop, y = mean)) + geom_point() +
  geom_text_repel(aes(x= prop, y = mean, label = user), size = 2.5) +
  xlab('Proportion of correct classification')+
  ylab('Abilities')
```



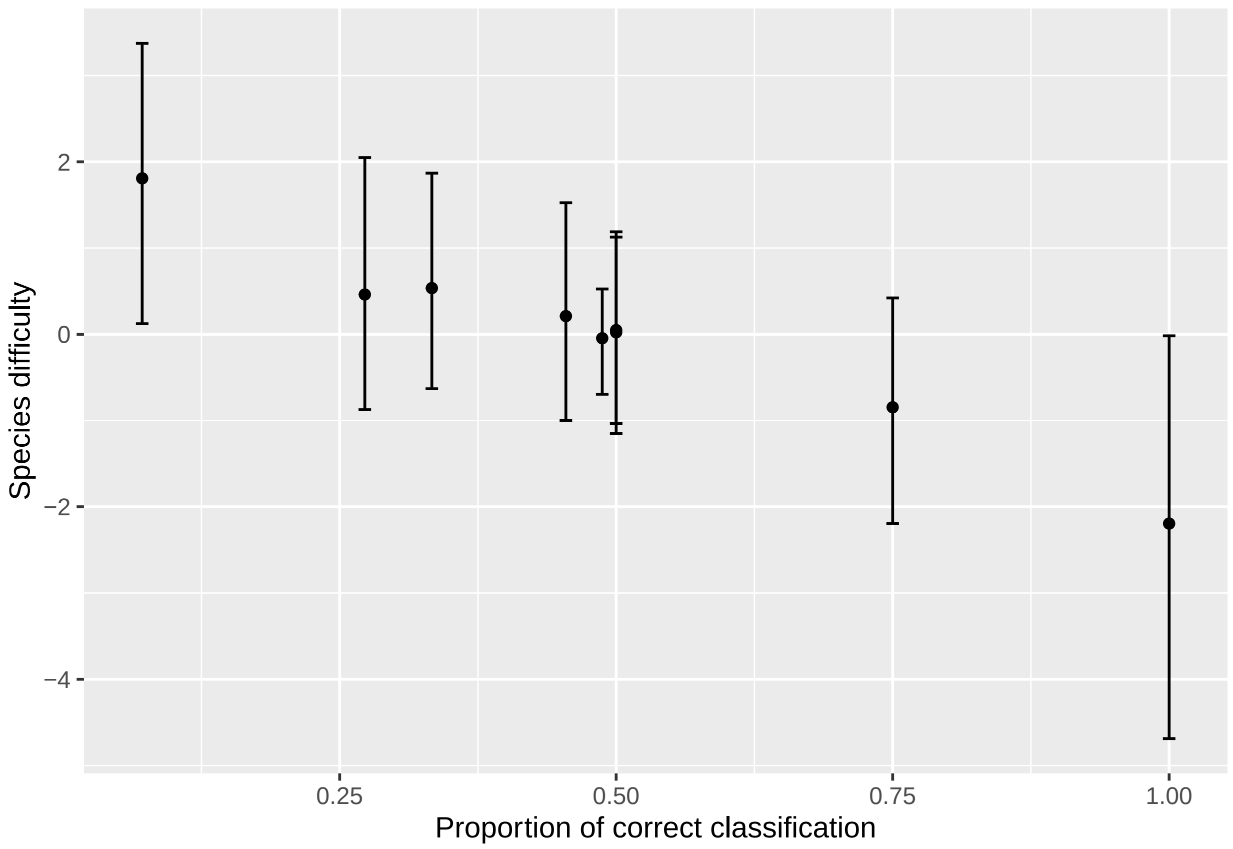
Plotting the difficulties:

```
diff <- stats[grep('difficulty\\[', rownames(stats)),]
diff <- cbind(diff, data %>% group_by(id, LocationX, LocationY) %>% summarize(ns = n(), prop = mean(cor

diff$diff_cat <- cut(diff$mean, breaks = c(-10,quantile(diff$mean)[2:4],10), 1:4, include.lowest = T)

cols = brewer.pal(5,'Reds')
ggplot(diff , aes(LocationX, LocationY, fill = diff_cat)) +
  stat_voronoi(color="black") + scale_fill_manual(values=cols) +
  geom_point() +
  geom_text(aes(LocationX, LocationY+1000, label = id), size = 2)+
  coord_fixed(ratio=1)+
  labs(fill = "Site difficulty")+
  theme_bw()
```



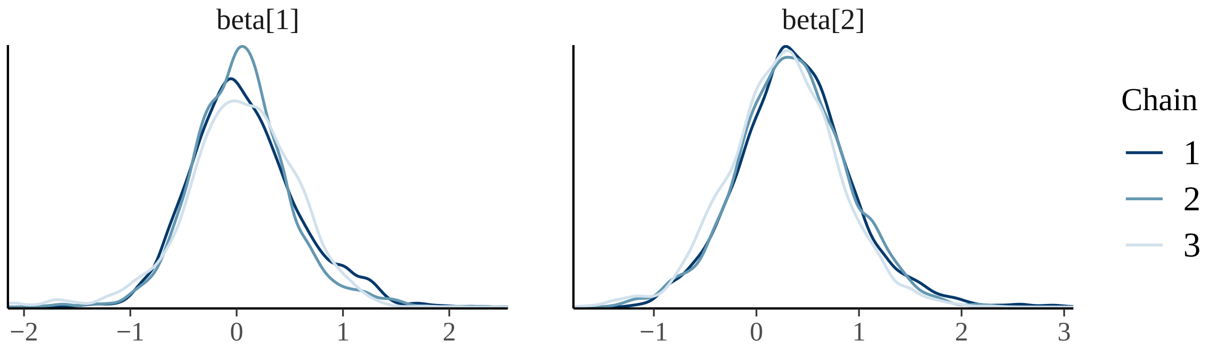
Plotting the estimated species difficulties:

```
species <- stats[grep('species\\[', rownames(stats)),]
species <- cbind(species, data %>% group_by(True_Species) %>% summarize(ns = n(), prop = mean(correct))


names(species)[grep('%', names(species))] <- c('q2.5','q25', 'q50', 'q75', 'q97.5')

ggplot(species, aes(x= prop, y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin=q2.5, ymax=q97.5))+
  xlab('Proportion of correct classification')+
  ylab('Species difficulty')
```



Posterior distributions of the regression coefficients:

```
mcmc_dens_overlay(
  array,
  pars = c(
    "beta[1]",
    "beta[2]"),
  facet_args = list(nrow = 1)
)
```


beta[1]
beta[2]
Chain