HMM-SSF example (part 1: data processing)

Natasha Klappstein

2023-01-25

This document describes the basic steps of generating control locations and obtaining covariates, implemented in Klappstein et al. (https://doi.org/10.1101/2022.11.30.518554). Note, this is a working document and will be be updated.

```
# load all necessary functions
library(raster)
source("code/functions/sim_controls.R") # to simulate control locations
source("code/functions/get_step_and_angle.R") # calculate steps/angles in a case-control df
```

The data

We analyse a dataset of plains zebra. The data consist of 30-minute locations, and are accompanied by a habitat type raster with four categories (grassland, bushed grassland, bushland, and woodland).

```
track <- read.csv("data/zebra.csv")
head(track)
```

```
##   ID        x        y                time
## 1  1 491.8303 7897.887 2013-12-31 19:00:00
## 2  1 491.7455 7897.867 2013-12-31 19:30:00
## 3  1 491.8045 7897.680 2013-12-31 20:00:00
## 4  1 491.8040 7897.676 2013-12-31 20:30:00
## 5  1 491.8022 7897.682 2013-12-31 21:00:00
## 6  1 491.8044 7897.696 2013-12-31 21:30:00
```

Control locations

First, we need to generate control locations. We use the function `sim_controls` to do so. This functions takes a dataframe (with columns `ID`, `x`, `y`, `time`; and assumes that the coordinates are projected), and requires the user to specify the number of controls to generate as well as the step length distribution (only options are `"gamma"` or `"uniform"`).

```
# use sim_controls function loaded above
set.seed(250)
data <- sim_controls(obs = track,
                     n_controls = 25,
                     step_dist = "gamma")
```

This returns a dataframe with the original, plus two additional columns: i) an `obs` column which indicates whether the location is an observation (i.e., a case, set to 1) or a random location for Monte Carlo integration (i.e., a control, set to 0), and ii) a `stratum` column which indexes the observed location (i.e., the case and all associated controls are in a single stratum).

```
head(data)
```

```
##   obs        x        y                time stratum ID
## 1   1 491.8303 7897.887 2013-12-31 19:00:00       1  1
## 2   0       NA       NA 2013-12-31 19:00:00       1  1
## 3   0       NA       NA 2013-12-31 19:00:00       1  1
## 4   0       NA       NA 2013-12-31 19:00:00       1  1
## 5   0       NA       NA 2013-12-31 19:00:00       1  1
## 6   0       NA       NA 2013-12-31 19:00:00       1  1
```

Covariates

We need covariates for both the case and control locations. We will be modelling habitat selection and movement based on the following covariates: i) step length, ii) turning angle, and iii) habitat type. Further, we will be modelling the transition probabilities as a function of time of day, so well obtain all these covariates now.

Movement covariates

First, we need to calculate the step lengths and turning angles. This is more complicated than just taking the distance between each row, as each case must be matched to the next case (for the observed step length) and all its associated controls (for the control step lengths). The same is true for turning angle, but using the current location and the previous and next locations. We call the custom function `get_step_and_angle` to compute these:

```
data <- get_step_and_angle(data)
zero_steps <- which(data$step == 0)
zero_steps
```

```
## [1]   7697 125607
```

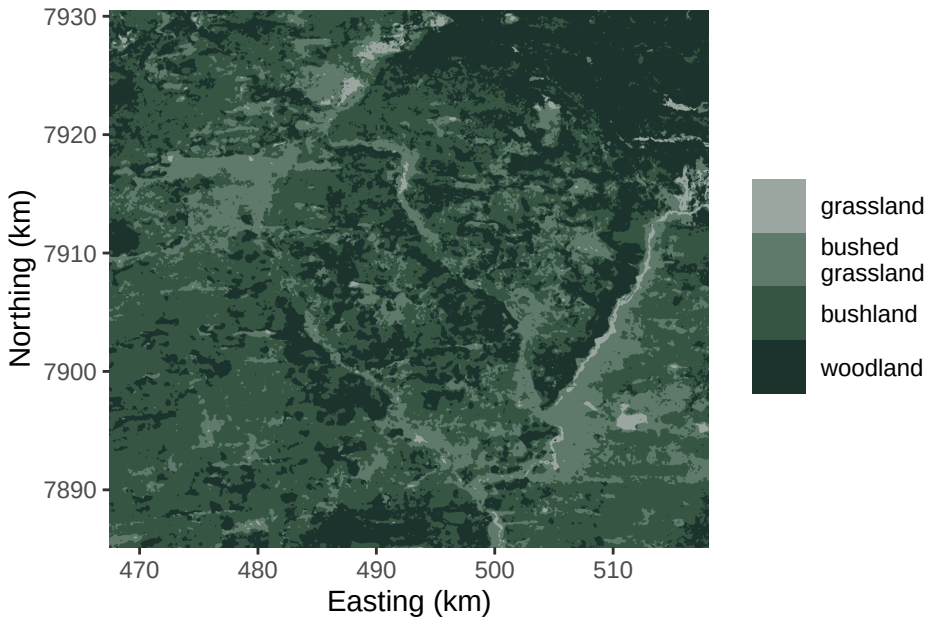
There are two observations with step lengths of 0km. Since we want to model step lengths with a gamma distribution, we need to either use a zero-inflated gamma, or do something to the data. Since we just have two observations, we simply add a very small number ($<$ the minimum non-zero step length). If there are alot of zeros in your data, this might not be a suitable approach, but currently the HMM-SSF can't easily accomodate zeros.

```
data$step[zero_steps] <- data$step[zero_steps] +
                         runif(length(zero_steps),
                               0,
                               min(data$step[-zero_steps], na.rm = T))
```

Environmental covariates

Now to obtain the environmental covariates, we can use the `raster` package, first ommitting the NAs in the location data and using a simple interpolation method (because we are dealing with a categorical variable). We also give explicit names to the habitat types, which are stored as integers 1 through 4 in the raster file.

```
hb <- raster("data/vegetation2.grd")
```




```
notNA <-  which(!is.na(data$x))
data$veg[notNA] <- extract(hb, data[notNA, c("x", "y")], method = "simple")
data$veg <- factor(data$veg)
levels(data$veg) <- c("grassland", "bushed grassland", "bushland", "woodland")
```

Temporal covariates

Lastly, we need a time of day covariate, and this is simply taken from the time column.

```
data$tod <- as.numeric(format(data$time, "%H")) +
  as.numeric(format(data$time, "%M")) / 60
```

```
# save processed data
saveRDS(data, file = "data/zebra_processed.RData")
```