Petrel example: smooth covariate effects and non-parametric movement kernel

Natasha Klappstein

2024-01-12

This document illustrates a simple example in which we implement a SSF with smooth effects for both movement and habitat covariates. We use a publicly available dataset of GPS locations from Antarctic petrels (described in Descamps et al. 2016a).

Contents

Set-up

First, load the packages we need for the analysis. `mgcv` is the main package for fitting the SSFs. `gratia` is full of useful tools for visualising and extracting `mgcv` outputs. `hmmSSF` is a package to fit state-switching SSFs, which has some data processing tools that we will be using. You can install the package with `devtools`:

```
devtools::install_github("NJKlappstein/hmmSSF")
```

```
library(mgcv)
library(ggplot2)
library(gratia)
library(hmmSSF)
library(moveHMM)
library(sp)
library(cowplot)
```

Data preparation

The data for this example were obtained from MoveBank (Descamps et al. 2016b). We subset the data to fewer individuals and conducted other pre-processing of the data (not shown here for simplicity). The data have columns for individual `ID`, `time`, longitude/latitude coordinates (`lon` and `lat`) and projected coordinates (`x` and `y`).

```
# load data
data <- readRDS("data/petrel.RData")
```

```r
head(data)
```

```
          ID                time      lon       lat        x         y
1231 4155777 2011-12-13 00:22:57 7.231573 -69.89723 184.7219 -954.3163
1232 4155777 2011-12-13 00:52:57 7.258316 -69.82245 188.7048 -946.8354
1233 4155777 2011-12-13 01:22:57 7.311297 -69.71320 195.0409 -936.1053
1234 4155777 2011-12-13 01:52:57 7.389327 -69.59586 202.6395 -924.8867
1235 4155777 2011-12-13 02:22:57 7.488740 -69.48424 210.8209 -914.5862
1236 4155777 2011-12-13 02:52:57 7.586307 -69.37217 218.9876 -904.2286
```

Next, we need to process the data to fit an SSF. Mainly, we need to: i) generate random points and ii) get covariates.

Generating random points

One goal of the analysis is to model a non-parametric movement kernel. This means that we need to sample spatially uniform points on a disc with radius R (rather than sampling from a parametric distribution of step lengths and/or turning angles). Unfortunately the package `amt` doesn't implement this form of sampling. We will use the `hmmSSF` package (see beginning of this document for download instructions), although you could also write code to do this yourself. The general steps are:

1. define R as the maximum observed step length (or something similar)
2. sample each turning angle from $\text{Unif}(-\pi, \pi)$
3. sample each step length as the square root of a random draw of $\text{Unif}(0, R^2)$

The `get_controls` function of `hmmSSF` can do this for us, and will also return step lengths and turning angles. Note that uniform sampling generally needs quite a few random points for model fitting, and so we will use 50 random points in this example.

```r
## get control/random locations ##
data <- hmmSSF::get_controls(obs = data,
                             n_controls = 50,
                             distr = "uniform")

# remove weights column returned by get_controls (unneeded here)
data$w <- NULL
```

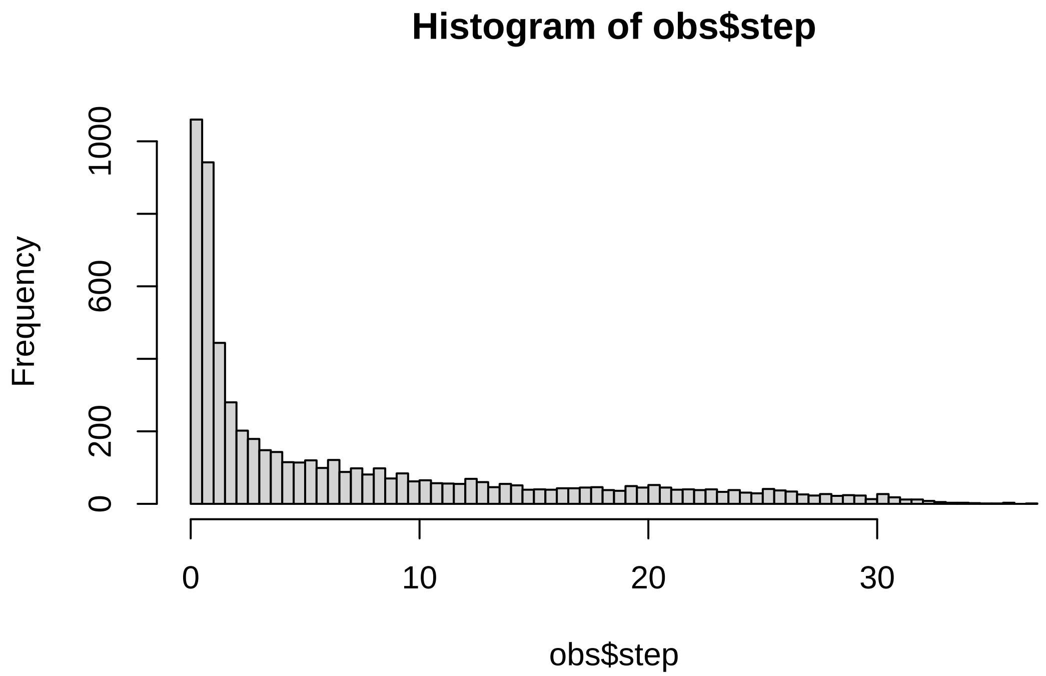
The data is returned as a dataframe with new columns `stratum` (identifier for each observed location and its associated random points), `obs` (binary variable to identify whether the location is observed or random; `obs = 1` if observed, and `obs = 0` if random), `step` is the step length, and `angle` is the turning angle. We removed the column `w` which are importance weights (used in the fitting functions of `hmmSSF` and not applicable here).

```r
head(data)
```

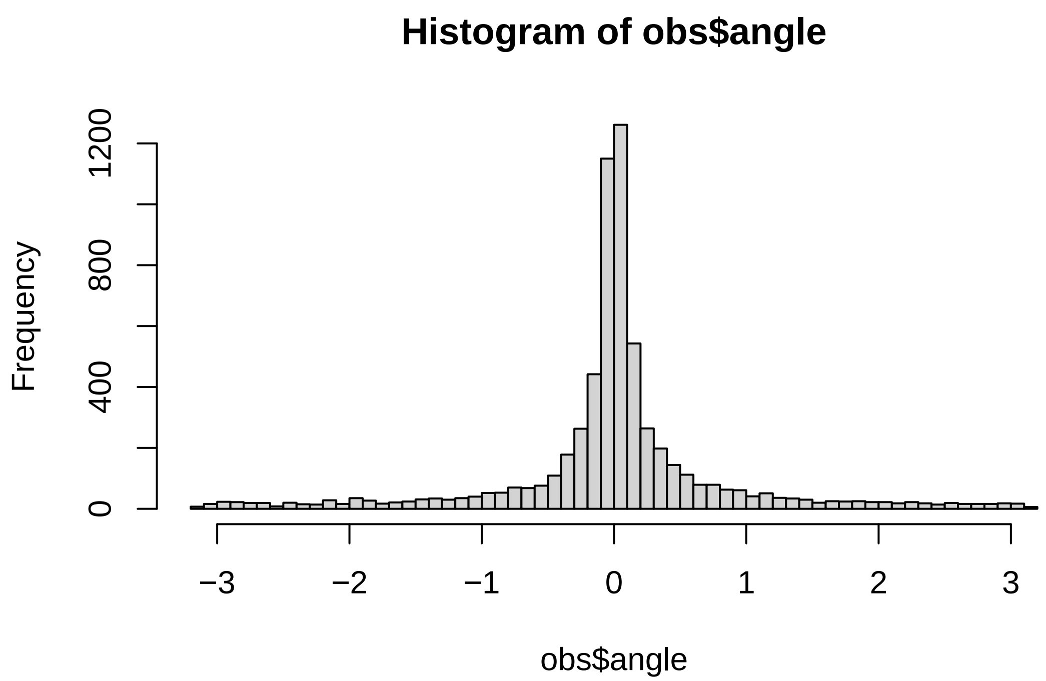
```
       ID   stratum obs        x         y                time     step
1 4155777 4155777-3   1 195.0409 -936.1053 2011-12-13 01:22:57 12.46126
2 4155777 4155777-3   0 157.8290 -933.8752 2011-12-13 01:22:57 33.48555
3 4155777 4155777-3   0 198.4964 -926.0304 2011-12-13 01:22:57 22.99395
4 4155777 4155777-3   0 200.2060 -976.7166 2011-12-13 01:22:57 32.01809
5 4155777 4155777-3   0 183.3082 -974.5418 2011-12-13 01:22:57 28.22708
6 4155777 4155777-3   0 161.5386 -923.4350 2011-12-13 01:22:57 35.85506
        angle
1 -0.04415178
2  1.66262572
3  0.04936751
4 -2.28492752
```

```
5 -2.84471575
6  1.34898090
```

```r
# plot histograms of observed step lengths and turning angles
obs <- subset(data, obs == 1)
hist(obs$step, 80)
```




```r
hist(obs$angle, 80)
```



Get covariates (and other data preparation)

We already have step lengths and turning angles, but we want to assess selection for "distance to colony" as well (as described in Michelot et al. 2023). We can derive this covariate using the code below. We also set a single observation with a step length of zero to a small number so we don't have to bother with zero inflation.

```
## Add distance to colony ##

# Define colony for each track as first observation
i0 <- c(1, which(data$ID[-1] != data$ID[-nrow(data)]) + 1)
centres <- data[i0, c("ID", "x", "y")]
centre_x <- rep(centres$x, rle(data$ID)$lengths)
centre_y <- rep(centres$y, rle(data$ID)$lengths)

# Add distance to colony as covariate (based on sp for great circle distance)
data$d2c <- sapply(1:nrow(data), function(i) {
  spDistsN1(pts = matrix(as.numeric(data[i, c("x", "y")]), ncol = 2),
            pt = c(centre_x[i], centre_y[i]),
            longlat = FALSE)
})

## Replace zero step length to very small number ##

# because it's overkill to use a zero-inflated distribution just for one zero
wh_zero <- which(data$step == 0)
data$step[wh_zero] <- runif(length(wh_zero),
                            min = 0,
                            max = min(data$step[-wh_zero], na.rm = TRUE))
```

Model fitting

The response needs to be a combination of a column of `times` (all same value) and stratum ID (as a factor). Since we sampled spatially uniform points, we can fit a non-parametric movement kernel by specifying smooth terms for step lengths and turning angles. For step lengths, we can use the default thin plate regression splines (TPRS). Turning angles are a circular distribution, and so we should use a cyclic spline. We also include a smooth effect of distance to colony, using TPRS. We specify `family = cox.ph` (identity link function) and use the `weights` argument to specify whether each row is an observed or random location (`obs`). We will use the default basis dimension for now.

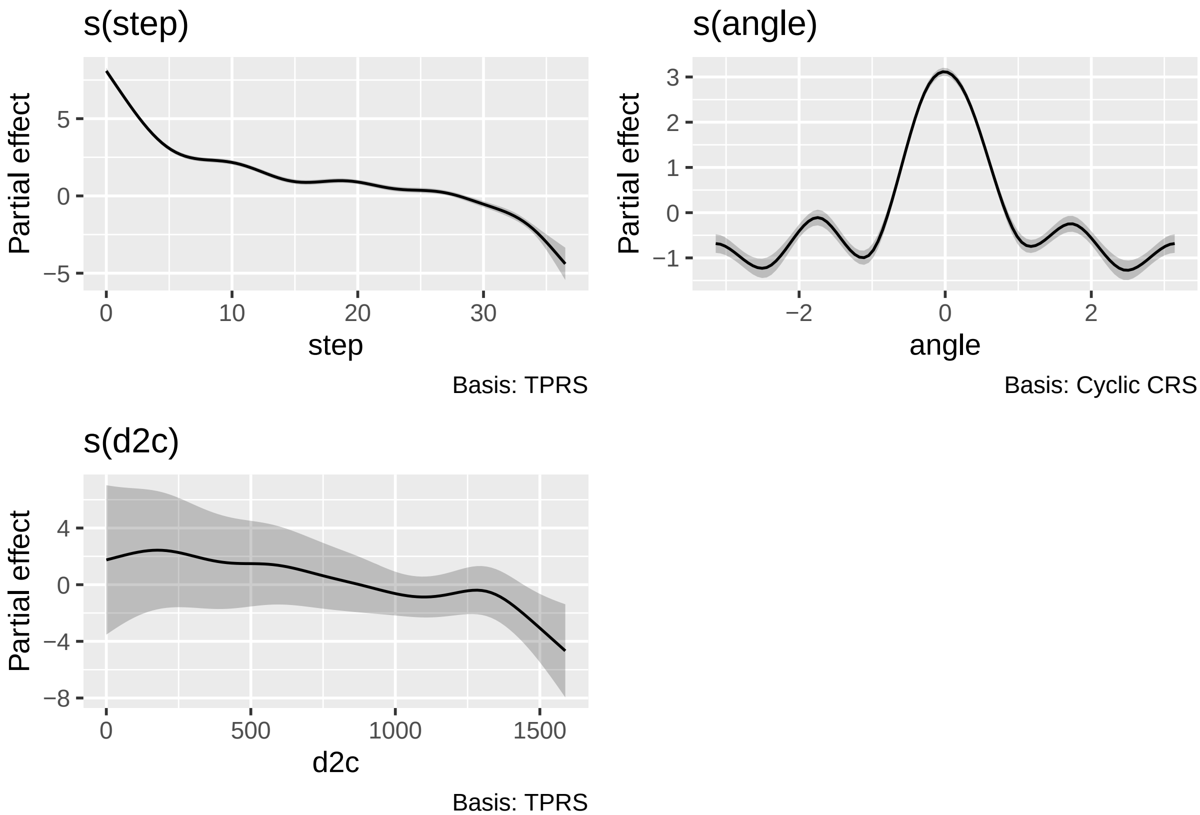
```
# add dummy time column
data$times <- 1

# fit in mgcv
fit <- gam(cbind(times, stratum) ~
             s(step) +
             s(angle, bs = "cc") +
             s(d2c),
           data = data,
           family = cox.ph,
           weights = obs)
```

Model interpretation

We can use `gratia` to visualise the smooth terms. I'm using `rug = FALSE` to omit plotting the observations on the x-axis. This shows us that there are non-linear patterns of movement and habitat selection.

```r
gratia::draw(fit, rug = FALSE)
```



Checking K

We can also check if the basis dimension K was set sufficiently high using the function `k.check()`. If the EDF are close to K, then this could indicate that K should be set higher.

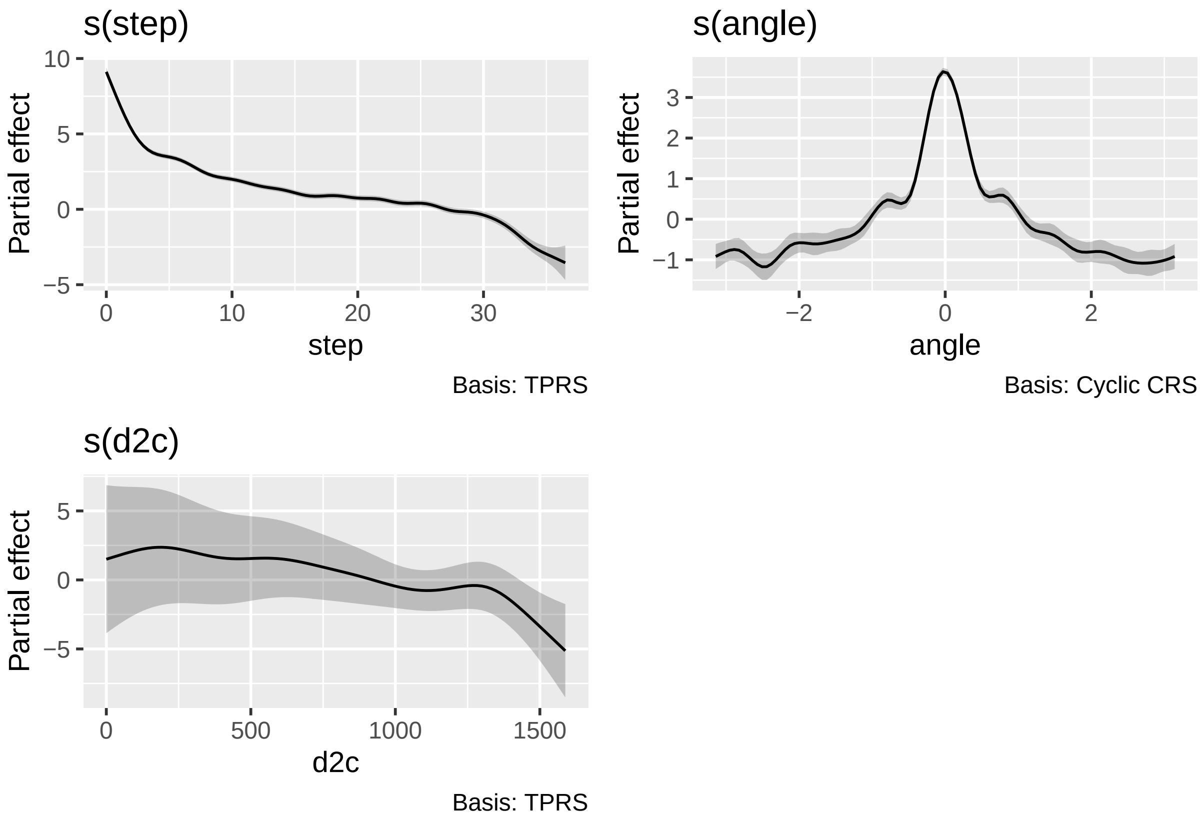
```r
k.check(fit)
```

```
         k'      edf   k-index p-value
s(step)   9 8.592056 0.9563010  0.0025
s(angle)  8 7.929789 0.9569600  0.0050
s(d2c)    9 5.576749 0.9751884  0.0375
```

This shows that the smooths for step length and turning angle may need a larger K. We will try 20 and see how that affects the model outputs.

```r
fit2 <- gam(cbind(times, stratum) ~
              s(step, k = 20) +
              s(angle, bs = "cc", k = 20) +
              s(d2c),
            data = data,
            family = cox.ph,
            weights = obs)
```

```r
gratia::draw(fit2, rug = FALSE)
```




```r
k.check(fit2)
```

```
          k'       edf   k-index p-value
s(step)   19 15.314278 0.7509220  0.0000
s(angle)  18 16.530564 1.0275359  0.9875
s(d2c)     9  5.651744 0.9922501  0.1250
```

The `k.check()` looks better, but it's maybe debatable whether this increase in EDF capturing biologically realistic complexity or not.

References

Descamps, S., Tarroux, A., Cherel, Y., Delord, K., Godø, O. R., Kato, A., Krafft, B. A., Lorentsen, S.-H., Ropert-Coudert, Y., Skaret, G., and Varpe, Ø. (2016a). At-sea distribution and prey selection of Antarctic petrels and commercial krill fisheries. PloS one, 11(8):e0156968.

Descamps, S., Tarroux, A., Cherel, Y., Delord, K., Godø, O. R., Kato, A., Krafft, B. A., Lorentsen, S.-H., Ropert-Coudert, Y., Skaret, G., and Varpe, Ø. (2016b). Data from: At-sea distribution and prey selection of Antarctic petrels and commercial krill fisheries.

Michelot T. (2023). hmmTMB: Hidden Markov models with flexible covariate effects in R. arXiv.