Zebra example: varying coefficients and spatial smoothing

Natasha Klappstein

2024-01-12

This document will cover the analysis of the zebra data in Klappstein et al. (2024).

Data and setup

```
# load packages and data

library(mgcv)
library(gratia)
library(ggplot2)
library(wesanderson)
library(dplyr)

data <- readRDS("data/zebra.RData")
```

The data are from Klappstein et al. (2023), in which random points were sampled from a gamma distribution with the mean and standard deviation from the observed data. So, we will get those parameters to use later to correct the estimated movement parameters.

```
# subset the data to the observed locations
obs <- subset(data, obs == 1)

# get sampling parameters
mu_samp <- mean(obs$step, na.rm = TRUE)
sd_samp <- sd(obs$step, na.rm = TRUE)
scale_samp <- sd_samp^2 / mu_samp
shape_samp <- mu_samp^2 / scale_samp^2
```

Model fitting

We want to fit a model with a time-varying gamma distribution of step lengths (where movement speed varies throughout the day), a von Mises distribution of turning angles, a linear effect of habitat type, and a spatial smooth. First, we need to create a dummy column for times (used in a Cox PH model, but not relevant to an SSF) that all contain the same value.

To model step lengths that follow a gamma distribution of step lengths, we include the step length and its log as covariates. We want to allow the movement speed to vary throughout the day, and so we will specify a time-varying step length. This translates to a time-varying scale parameter of the gamma distribution (a time-varying log step length would model a time-varying shape parameter). We can fit the varying-coefficient term using the `by` argument of the smooth; here, `s(tod, by = step)`. We use cyclic cubic regression spline to ensure a cyclic pattern across the day (i.e., the start and end of the day will match). We model the spatial smooth with two-dimensional thin-plate regression splines, with default settings (although there are many other options for spatial smoothing). It's a complex model so it takes several minutes to fit.

```r
data$times <- 1 # dummy time variable

# fit model with varying coefficient for step
fit_tod <- gam(cbind(times, stratum) ~
                 log(step) +
                 s(tod, by = step, bs = "cc", k=10) + # time varying step length
                 cos(angle) +
                 s(x, y) + # spatial smooth
                 veg,
               data = data,
               family = cox.ph,
               weights = obs)
```

Model interpretation

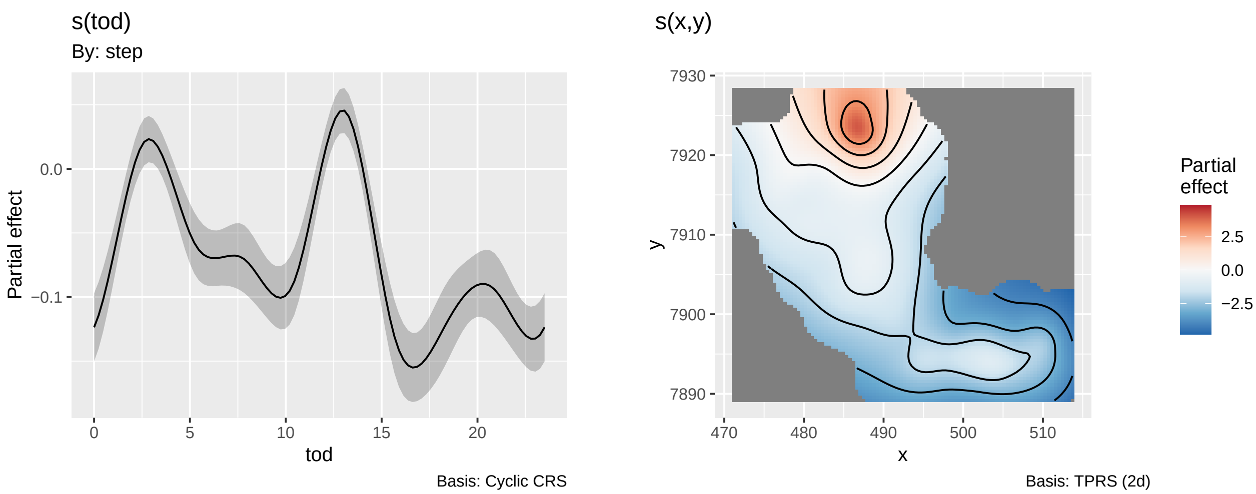
We can look at the model summary for the fixed effect coefficients, effective degrees of freedom, and other ouputs of interest. The coefficient log(step) relates to the shape parameter of the gamma distribution, and the coefficient for cos(angle) indicates that the turning angles follow a von Mises distribution with a mean of 0 and concentration fof 0.47. The coefficients for habitat type are relative to grassland (the reference), and indicate that the zebra avoids all other habitat types.

```r
summary(fit_tod)
```

```
Family: Cox PH
Link function: identity

Formula:
cbind(times, stratum) ~ log(step) + s(tod, by = step, bs = "cc",
    k = 10) + cos(angle) + s(x, y) + veg

Parametric coefficients:
                     Estimate Std. Error z value Pr(>|z|)
log(step)             0.216153   0.009255   23.36   <2e-16 ***
cos(angle)            0.473109   0.018086   26.16   <2e-16 ***
vegbushed grassland -0.781038   0.046552  -16.78   <2e-16 ***
vegbushland         -1.662605   0.068772  -24.18   <2e-16 ***
vegwoodland         -1.681015   0.104133  -16.14   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
               edf Ref.df Chi.sq p-value
s(tod):step  8.686  8.971 295.08 < 2e-16 ***
s(x,y)      19.932 25.102  50.22 0.00225 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Deviance explained = -86.4%
-REML =  21397  Scale est. = 1         n = 181953
```

For smooth effects, most of our interpretation happens when we visualise the smooth functions, and we can use gratia for plotting. For a varying-coefficient model, the plot for this interaction how the selection for step varies smoothly with time of day. The spatial smooth can be interpreted in terms of relative selection strength (see Klappstein et al. 2024).

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



Updating the step length coefficient

Importantly, the plot for a time-varying step length coefficient is not on the scale of interest for two reasons: i) the partial effect is measuring how the selection for step deviates from the sampling distribution (i.e., we have not "corrected" or "updated" the parameter estimates), and ii) the partial effect on this particular plot (from gratia) shows the effect for the mean step (i.e., the axis is actually mean step × s(tod)).

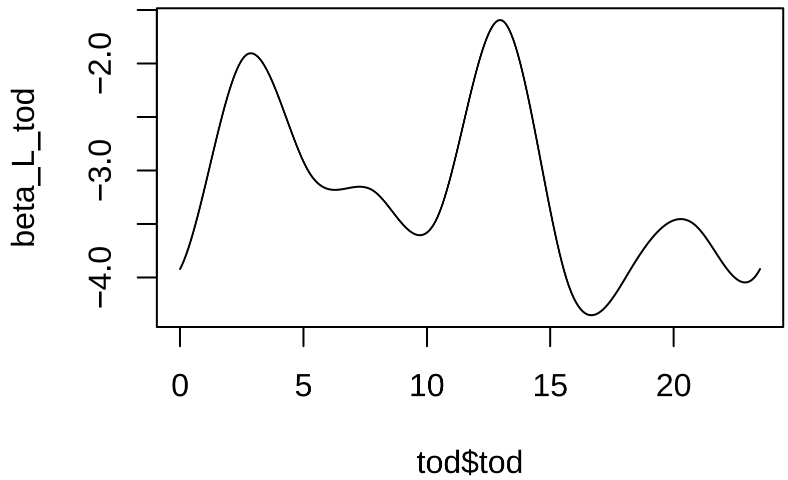
We need to "correct" the step length coefficient estimates to get the true β estimates (i.e., we need to account for our sampling). To do so, we first need to extract the smooth estimates from the model output. Note that the smooth has been multiplied by the mean step length.

```r
# get smooth terms
tod <- smooth_estimates(fit_tod, smooth = "s(tod):step", n = 1000)
head(tod)
```

```
# A tibble: 6 x 7
  smooth      type       by       est     se    tod   step
  <chr>       <chr>      <chr>  <dbl>  <dbl>  <dbl>  <dbl>
1 s(tod):step Cyclic CRS step  -0.124 0.0136 0      0.0729
2 s(tod):step Cyclic CRS step  -0.123 0.0136 0.0235 0.0729
3 s(tod):step Cyclic CRS step  -0.122 0.0135 0.0470 0.0729
4 s(tod):step Cyclic CRS step  -0.121 0.0135 0.0706 0.0729
5 s(tod):step Cyclic CRS step  -0.120 0.0135 0.0941 0.0729
6 s(tod):step Cyclic CRS step  -0.119 0.0135 0.118  0.0729
```

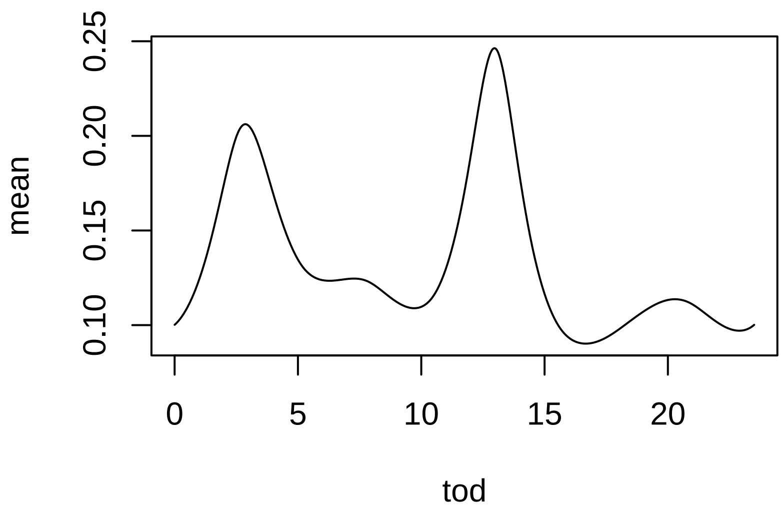
Then, we can correct the estimates. There are essentially two things happening in the code below: i) scale the estimates by the mean step (i.e., divide the estimate by the mean step because the smooth was multiplied by the mean step to produce the estimates within the `smooth_estimates()` output), and ii) perform the parameter corrections (Table 1 of Klappstein et al. 2024).

```r
# translate to mean/sd
beta_L_tod <- tod$est / tod$step[1] - (1/scale_samp)
beta_logL_tod <- fit_tod$coefficients[1] + shape_samp - 2
plot(beta_L_tod ~ tod$tod, type = "l")
```



Now that we have the correct β estimates for step length, we can convert this to the mean step length and plot this as a function of time of day.

```
mean_tod <- -(beta_logL_tod + 2) / (beta_L_tod)
sd_tod <- -sqrt(beta_logL_tod + 2) / beta_L_tod

#plot
tod_df <- data.frame(tod = tod$tod,
                     mean = mean_tod,
                     sd = sd_tod)

plot(mean ~ tod, tod_df, type = "l")
```



References

Klappstein NJ, L Thomas, and T Michelot. 2023. Flexible hidden Markov models for behaviour-dependent habitat selecton. Movement Ecology 11:30.

Klappstein NJ, T Michelot, J Fieberg, EJ Pedersen, C Field, and J Mills Flemming. 2024. Step selection analysis with non-linear and random effects in mgcv. bioRxiv.