Component Weights

J.P. Meagher

9 October 2017

Having performed a principal components analysis on the spectrograms of bat echolocation calls, approximations for evolutionary components have been identified.

The next step in the analysis is to calculate the weight of each component present in the spectrograms.

Packages

```
library(batwork)
library(tidyverse)
library(magrittr)
library(RColorBrewer)
```

Data

```
observed_frequencies <- get_spectrogram_details(
  mexican_bat_calls %>% use_series('call') %>% extract2(1),
  detail = 'freq')

band <- observed_frequencies > 9000 & observed_frequencies < 212000

df <- mexican_bat_calls %>%
  mutate(time = map(calls, get_spectrogram_details, detail = 'time'),
    psd = map(calls, get_psd)) %>%
  mutate(restricted = map(psd, extract, , band)) %>%
  mutate(restricted = map(restricted, log10)) %>%
  mutate(restricted = map(restricted, multiply_by, 10)) %>%
  mutate(smoothed = map(restricted, waveslim::denoise.dwt.2d, J = 2)) %>%
  mutate(regularised = map(smoothed, grid_interpolation)) %>%
  select(bat, species, sex, regularised)

f <- seq(observed_frequencies %>% extract(band) %>% min,
  observed_frequencies %>% extract(band) %>% max,
  length.out = df %>% use_series('regularised') %>% extract2(1) %>% ncol)

t <- seq(0, 1,
  length.out = df %>% use_series('regularised') %>% extract2(1) %>% nrow)
```

Vectorise Spectrograms

In order to perform the required calculations, the spectrograms need to be vectorised. The vectors can be returned to matrix form easily if necessary to aid interpretation.

```
df <- df %>%
  mutate(vectorised = map(regularised, c))
```

Centering

When estimating the weight of each component in a spectrogram it is the weight of the component in the variation of the spectrogram from the mean that is of interest. Thus spectrograms must be centred before calculating weights.

```
mean_spectrogram <- df %>%
  select(vectorised) %>%
  unlist %>%
  matrix(nrow = 50*100, ncol = df %>% nrow) %>%
  rowMeans

df <- df %>%
  mutate(centred = map(vectorised, subtract, mean_spectrogram))
```

Echolocation Call Components

A principal components analysis will yeild call components that can be used to approximate echolocation call components.

```
components <- df %>%
  select(regularised) %>%
  unnest %>%
  use_series('regularised') %>%
  array(dim = c(50*100, df %>% nrow)) %>%
  t %>%
  prcomp %>%
  use_series(rotation)
```

Component Weights

A linear combination of the components in each call can now be calculated. The full model is too large to store in its entirety, so only the coefficients associated with the model will be stored.

```
component_weights <- function(centred_spectrogram, evolutionary_components){
  weights <- lm(centred_spectrogram ~ evolutionary_components) %>% use_series(coefficients)
  return(weights)
}

df <- df %>%
  mutate(weights = map(centred, component_weights, evolutionary_components = components[, 1:6]))

wt <- df %>%
  select(bat, sex, species, weights) %>%
  mutate(weights = map(weights, matrix, nrow = 1)) %>%
  mutate(weights = map(weights, as.tibble)) %>%
  unnest
```

There is a lot of tuning that can be performed on this linear model, for example, weighting the regression to ensure it captures peaks in the spectrogram well. The residuals of these linear models should also be closely examined in order to better understand the nature of any variation not captured by the components.