Software Tools for High-Resolution Movement Tags
Practical 3

9 August 2017

Contents

1 Introduction **1**

2 Detecting and Summarising Dives **2**
2.1 Exploring a ready-made dataset 2
2.2 Finding dives and summarising them 3

3 A regression model **4**

4 Rotation test **7**

5 Mahalanobis distance **9**

6 HMM for inferring behavioural states from tag data **10**

References **11**

1 Introduction

The practical (like all the ones before it!) contains more exercises than you are likely to be able to complete in the time available, but each section is designed to be relatively stand-alone, so please feel free to pick and choose the topics that are most interesting to you.

Data are provided for each example, but please feel free to try to incorporate your own data as time and ambition allow.

Many of the statistical analyses are much easier to do in R. This is because packages to do much more sophisticated stats are available, and they are better documented (i.e. you can determine exactly what they are doing with

the data and what the output means!). The tasks that I don't know how to do in Matlab are clearly marked and no Matlab code is given for them (although you are welcome to try to figure out a way!).

2 Detecting and Summarising Dives

2.1 Exploring a ready-made dataset

OK in both R and Matlab/Octave.

Consider a dataset on 272 dives by 15 Cuvier's beaked whales. The data were collected using DTAGs, and published along with DeRuiter et al. [2013]. The data are available from `http://dx.doi.org/10.5061/dryad.n77k3`, but we will load a slightly cleaned-up version of the dataset with more manageable variable names. (If you are not sure what any of the variables are measuring or want to check their units, have a look at the file on Dryad.)

1. If you want practice tidying up the variable names yourself, fetch the original text file from the Dryad repository and get to work.

2. Read in the clean data from the file `zc_dives.csv` on animaltags.org, or from the url `http://www.calvin.edu/~sld33/data/zc_dives.csv`. The main dataset has one column of whale IDs which are strings rather than numeric values. If you would prefer not to deal with these in Matlab/Octave, there is a version of the file called zc_dives_numeric.csv that omits that column.

```
zc_dives <- read.csv(file='http://www.calvin.edu/~sld33/data/zc_dives.csv')
```

```
zc_dives = csvread('zc_dives_numeric.csv',1);
```

3. Create a simple box plot of the whole dataset (one boxplot per column, since each column of the dataset is one dive summary metric).

```
boxplot(zc_dives)
```

   (a) What do you notice about the data?

(b) How could the visualization be improved (so you can better see patterns in all the variables)? Think creatively and check out the help for the boxplot function for more ideas...What alternatives to box plots might be more informative here?

2.2 Finding dives and summarising them

OK in both R and Matlab/Octave.

Now consider a dataset from a DTAG attached to a Cuvier's beaked whale, *Ziphius cavirostris*. Load the data from file `zc11_267a.nc`.

1. Make a plot of the dive profile. What do you notice?
2. You probably want to crop the data before further analysis, because there is a period at the start of the recording when the tag was not yet deployed on the whale.
3. What minimum depth threshold do you think you would use to detect dives by this animal? Consider how you would justify your choice.
4. Use `find_dives` to detect all dives deeper than your chosen depth *mindepth* (to run in matlab, just omit the "mindepth="):

   In R:

   ```
   dt <- find_dives(zc$P, mindepth=mindepth)
   ```

   In Matlab/Octave:

   ```
   dt = find_dives(P, mindepth);
   ```

5. Now use `dive_stats` to compute summary statistics for all the dives you detected.

   In R:

   ```
   ds <- dive_stats(zc$P, dive_cues=dt[,c('start', 'end'),])
   ```

   In Matlab/Octave:

```
ds = dive_stats(P, [ds.start, ds.end]);
```

6. Have a look at the dive stats and perhaps make a plot of some or all of them. Do you notice anything interesting?

7. Choose an auxiliary variable (could be anything of interest - pitch, roll, heading, MSA, ODBA, njerk...). Compute the auxiliary variable, and then recompute the dive stats including the auxiliary variable.

   In R:

```
ds <- dive_stats(zc$P, dive_cues=dt[,c('start', 'end'),],
                 X=my_aux_var)
```

   In Matlab/Octave:

```
ds = dive_stats(P, [ds.start, ds.end], my_aux_var);
```

8. Examine and/or plot again.

3 A regression model

R only Let's consider again the beaked whale dive data. The dataset contains data on 272 dives by 15 Cuvier's beaked whales. The data were collected using DTAGs, and published along with DeRuiter et al. [2013]. The data are available from `http://dx.doi.org/10.5061/dryad.n77k3`, but we will load a slightly cleaned-up version of the dataset with more manageable variable names. If you haven't already loaded it, it is available from the file `zc_dives.csv` on animaltags.org, or from the url `http://www.calvin.edu/~sld33/data/zc_dives.csv`.

1. Let's try to formulate a model for max_depth (the maximum depth attained during each dive). Here is a first attempt - feel free to follow along through this example as written, or try other predictor or response variables if you like.

```
lm1 <- lm(max_depth ~ descent_rate + fluke_rate +
            odba + dive_type,
          data=zc_dives)
summary(lm1)
```

2. It would have been a good idea to look at scatter plots of the data for each candidate predictor variable before fitting the model, in order to verify that there isn't a non-linear relationship between the predictor(s) and the response variable. Do that now - do you see any problems? If so you may want to consider fitting smooth terms instead for those predictors (function gam from package mgcv for example).

   For example:

```
plot(max_depth~descent_rate, data=zc_dives)
```

3. Considering the summary output, is anything suprprising? Interesting?

4. Before interpreting the results too intently, we should do some model assessment. Consider some plots of the data and residuals (feel free to add to the suggestions below if you have others you like to see). Do you see any surprises?

```
plot(resid(lm1)~fitted(lm1))
hist(resid(lm1))
acf(resid(lm1))
```

5. When we plotted the ACF above, it computed ACF values regardless of whale ID. The data includes observations of multiple animals, and while we might expect temporal autocorrelation within an animal, we really wouldn't expect much between animals tagged independently. It would be better to compute the ACF only *within* whales, respecting the whale IDs. The tag tool kit provides a function to do this:

```
block_acf(resid(lm1), blocks = zc_dives$whale_ID,
          max_lag=15)
```

6. The next step would be to adjust the model to try to account for any problems found during the model assessment. In the example above, there doesn't seem to be an issue with autocorrelation! Why do you think that is? However, there is a problem with non-constant error variance. We might be able to fix it by transforming the response variable:

```
lm2 <- lm(log(max_depth) ~ descent_rate + fluke_rate +
            odba + dive_type,
          data=zc_dives)
```

7. Examine the summary and diagnostic plots again. Would you trust the model results now, or make further adjustments? Which variables do you think should be retained in the best model for this response variable?

8. Now consider a second model for post_dive_surf (with predictor(s) of your choice). Fit the new model and do model assessment again. (Be sure to use block_acf.) In this case, you will see a small but potentially worrisome amount of temporal autocorrelation in the residuals.

9. Try fitting a GEE instead of a linear model to account for this correlation over time. (Adjust the model below to fit your predictor and response variables.) You will have to have the package geeglm installed.

```
gee1 <- geepack::geeglm(log(post_dive_surf) ~ max_depth +
                          descent_rate + dive_dur,
                        data=zc_dives,
                        id=zc_dives$whale_ID)
```

10. Examine the summary and diagnostic plots again. How have the p-values and standard errors changed, and why? How has the ACF plot changed, and why?

11. Consider whether, for these data, you would prefer to fit a GEE or a random effects model to account for the temporal correlation within individuals. How would you justify your choice?

4 Rotation test

OK in both R and Matlab/Octave We were very fortunate to obtain a number of test datasets from different sources that we have permission to make publicly available with the tag tool kit. One dataset (obtained from anonymous Scottish contacts) is particularly exciting and possibly unique in the world: a fragment of tag data obtained from a high-resolution movement tag deployed on Nessie, the Loch Ness Monster. Unfortunately several of the tag sensors malfunctioned, but we were able to salvage some dive depth data to be used in this example. The dataset is called nessie.nc.

1. Read in the data and make a plot of the dive profie (because of course you want to see it).

2. According to some Scottish lore, Nessie surfaces more often in the hour around noon than during the rest of the day (because the glare on the water, and the lure of lunch, make it more difficult for people to spot her then). But does she really? Use find_dives to find start times for all her submergences, which we will use as a proxy for breath times. In this case you will want to use a threshold that is as shallow as practicable.

3. Do you think you could just use a regression model for surfacing rate to answer this question? Why or why not?

4. Use a rotation test to test whether the number of surfacings between 11:30 and 12:30.

```
th = 0.6;
```

In R:

```
setwd('C:/Users/Stacy DeRuiter/Dropbox/TagTools/data')
nessie <- load_nc('nessie.nc')
#make time variables
t <- as.POSIXct(nessie$info$dephist_device_datetime_start,
                tz='GMT') +
  c(1:nrow(nessie$P$data))/nessie$P$sampling_rate
#find data times between 11:30 and 12:30
s <- as.POSIXct('2017-01-13 11:30:00', tz='GMT')
```

```
e <- as.POSIXct('2017-01-13 12:30:00', tz='GMT')
noon <- range(which(t < e & t > s))
#convert to seconds
noon <- noon/nessie$P$sampling_rate
#find dives
dt <- find_dives(nessie$P, mindepth=th)
#do test
RTR <- rotation_test(event_times = dt$start,
                     exp_period=noon,
                     full_period=c(0,
      length(nessie$P$data)/nessie$P$sampling_rate),
      n_rot=10000, ts_fun=length)
```

In Matlab:

```
nessie = load_nc('nessie.nc');
%make time variables
t = datenum(info.dephist_device_datetime_start) + ...
  [1:length(P.data)]./P.sampling_rate/3600/24 ;
%find data times between 11:30 and 12:30
s = datenum('2017-01-13 11:30:00');
e = datenum('2017-01-13 12:30:00');
[st,et] = bounds(find(t < e & t > s));
%convert to seconds
st = st/P.sampling_rate;
et = et/P.sampling_rate;
%find dives
dt = find_dives(P, th);
%do test
RTR = rotation_test(dt.start,[st,et], ...
                     [0,length(P.data)/P.sampling_rate],...
                    100000, 'length', [],[],[]);
```

5. What can you conclude?

5 Mahalanobis distance

OK in both R and Matlab/Octave but Matlab code is not provided - use the example in the previous section to translate from R. (Also refer to help for the m_dist function.) Consider again the dataset zc11_267a.

1. How might you use Mahalanobis distance to summarise the multivariate tag data into a single data stream? What input variables might you choose to try to quantify how the whale might have changed its behavior in response to sonar sounds? (If you need more context, you can check out the paper at `http://rsbl.royalsocietypublishing.org/content/9/4/20130223`.)

2. Use the tag tool function m_dist to compute the Mahalanobis distance using your chosen inputs, using the model below as a guide (note that my choices of inputs are kind of ridiculous - do better!). You will have to choose an averaging window length and a between window overlap. You can use the experiment start and end times provided. How would you justify these choices?

```
MDdata <- data.frame(jerk=njerk(zc$A),
                     Mx = zc$P$data)
est <- as.numeric(as.POSIXct('2011-09-24 14:45:00') -
  as.POSIXct(zc$info$dephist_device_datetime_start))*3600

eet <- as.numeric(as.POSIXct('2011-09-24 15:15:00')-
  as.POSIXct(zc$info$dephist_device_datetime_start))*3600

MD <- m_dist(data=MDdata, sampling_rate=zc$P$sampling_rate,
             smoothDur = 10, overlap = 9.5,
             expStart = est, expEnd = eet)
plot(MD$t, MD$dist, type='l')
```

3. If you included dive depth as an input variable, how did it affect the resulting distance metric? Why do you think that is? Could there be another, better way to include information about dive profile in the Mahalanobis distance metric?

4. Do you think there was a "change" in behavior in response to the sonar exposure?

5. If you wanted to set a threshold for detecting a "change", how would you do it?

6 HMM for inferring behavioural states from tag data

R only Let's reconsider the sheep data that you used for yesterday's practical. Load in the file oa14_319a.nc, which contains the magnetometer corrections you worked on in Practical 2 already.

```
setwd('C:/Users/Stacy DeRuiter/Dropbox/TagTools/data')
sheep <- load('oa14_319a.nc')
behav <- read.csv('H2_split0_behaviors.csv')
```

1. We would like to fit an HMM to try to infer sheep behavior states. What are 1-3 variables that you think might be informative to help discriminate between activities like walking, running, grazing, etc.? Compute your variables of choice for the sheep data.

2. We will use the R package momentuHMM to fit a simple HMM to the data (the example shown here uses MSA, but please choose your own set of one or more input data streams). You should consider summarizing the data over longer time intervals than the sensor data sampling rate or cropping the data for model fitting.

   First some preparations:

   ```
   library(momentuHMM)
   library(MASS) #for fitdistr()
   Ac <- crop_to(sheep$A, tcues=c(10000,15000))$X
   data <- data.frame(MSA=msa(Ac))
   data <- prepData(data, coordNames=NULL)
   ```

   Now fit the model:

```
myHMM <- fitHMM(data, nbStates=2, dist=list(MSA='exp'),
                Par0 = list(MSA=c(rate1=1.6, rate2=1)))
myHMM
```

Now "decode" (identify the state that was most likely at each observed time point):

```
states <- viterbi(myHMM)
```

3. The model I showed above is hyper-simplistic - a good model for sheep behavior would probably use more than one input data stream (and maybe not MSA), would probably have more than one state, and might not use an exponential distribution for the state-dependent process. However, it is simple enough to be fitted easily and without any errors! See if you can progressively make the model more realistic and interesting.

4. If you think you may have done a decent job fitting a model to some of the sheep data, see how you did! The results of a behavior classification model fitted by the data owners (Juan Morales and coworkers) are available in the file sheep_behavior.csv. How do your results compare?

References

Stacy L DeRuiter, Brandon L Southall, John Calambokidis, Walter M X Zimmer, Dinara Sadykova, Erin A Falcone, Ari S Friedlaender, John E Joseph, David Moretti, Gregory S Schorr, Len Thomas, and Peter L Tyack. First direct measurements of behavioural responses by Cuvier's beaked whales to mid-frequency active sonar. *Biology letters*, 9(4):20130223, aug 2013. ISSN 1744-957X. doi: 10.1098/rsbl.2013.0223.