Quantitative Wildlife Ecology: Lab Exercise 9

Single Season Occupancy Modeling

Background

Occupancy modeling aims to estimate the proportion of sites occupied by a species of interest. A single season occupancy modeling approach requires multiple visits to sites during a season in which a species may be detectable. The model assumes that during these visits no individuals enter or leave the population (i.e., closed model). In each visit an observer employs some sort of sampling methodology to detect the presence ("`1`") or absence ("`0`") of the species of interest. Note that an absence may be a real absence or a failure to detect the species. In this exercise we are going to use detection histories to estimate the probability of species presence and detectability using a maximum-likelihood framework. Also, we are going to explore how covariates for sites or observations can improve model fit.

The input data for single season occupancy models are detection histories. Each spatial sampling unit (e.g., sites) will be characterized by one of these histories. For example, let's say that site 1 had a detection history `101`. This means that the individual was detected in the first census, was missed in the second census and was again detected in the third census. Given that we are assuming that the system is closed, then the "`0`" means that the individual was there but was not detected by the observer. Let's take another example: say site 2 had a detection history of `000`. Two things can explain this detection history: (1) either the species was present and was not detected by the observer, or (2) the species was absent.

Reminder on notation used in occupancy modeling:
Psi (ψ): probability of species presence
p: probability of detection

The data

The objective of this exercise is to use a simulated data set of detection histories and covariates to model detection probability and occupancy. We will evaluate different candidate models and interpret model outputs. We are going to use the library `unmarked` in R to estimate occupancy. Outside of R, other software such as Mark or Presence may also be used to estimate occupancy.

Let's pretend that you have just returned from a field visit to the Dudhwa National Park at the foothills of the Himalayas. You have been collecting data across the park on the presence of the fishing cat *Prionailurus viverrinus* (see http://en.wikipedia.org/wiki/Fishing_cat). Not much is known about this cat, and you want to:

> (a) obtain some basic information on the proportion of the study area occupied by the species, and
> (b) determine drivers of its spatial distribution within and around the park.

You used track-pads over 135 systematically placed sites and sampled them over six occasions (replicates). You used bait from the third replicate onward to increase your detection probability. You also collected information on certain covariates. You believe that distance to water bodies will influence the species' distribution as the fishing cats' diet is predominantly comprised of fish. You also think that shrub density (as an index of understory cover) may influence its whereabouts. Increased human density is often correlated with decreased fish density in water bodies, and there is some evidence of illegal poaching of the fishing cats, so you have also collected data on human presence, in the form of a 'disturbance category' (low disturbance, medium, high or very high). Notice you have two continuous covariates (distance to water and shrub) and one categorical variable (disturbance).

Remember to set your working directory. Now let's install the package 'unmarked'.

```
> install.packages("unmarked")
> library(unmarked)
```

Remember that by typing `?unmarked` you can see the documentation for the package.

We will now input the fishing cat data. It is in a file called 'fishcats.txt'.

```
> fishcats.data = read.table("fishcats.txt", header=T)
```

Now let's take a look at this input dataset.

```
> head(fishcats.data)
```

The first six columns (`Detection.1`, `Detection.2`, etc) correspond to the detection histories. The rest of the columns represent covariates detailed above. `Dist.water` is the distance to the nearest water body. `Disturbance` is the human disturbance category. `Shrub` is the localized shrub density. `Bait.1` through `Bait.6` indicates if

you have used bait for that replicate. The bait covariate is indexed from 1 to 6, corresponding to the particular replicate. For example, the column called `Bait.1` corresponds to whether bait was used for the first occasion and so on.

We first need to convert these data into a framework recognized within the unmarked package. We will use the function `unmarkedFrameOccu` for this purpose. Type `?unmarkedFrameOccu` to open the helpfile. You will notice that the function takes in the detection history (y), site-specific occupancy covariates (siteCovs), and survey(replicate)-specific detection covariates (obsCovs). Let's input these in:

```
> fishcats =unmarkedFrameOccu(y=fishcats.data[,1:6],
siteCovs=fishcats.data[,7:9],
obsCovs=list(Bait=fishcats.data[,10:15]))
```

You have just indicated that the first 6 columns of the inputted dataset are the detection history, columns 7 to 9 represent site covariates and the 10th to the 15th column represents a single time-covariate, which you have called 'Bait'.

Certain packages, such as `unmarked`, use a more rigorous object oriented system called "S4" (you can search the internet if you are interested in learning more about all of R's object oriented systems). Our function above coerced complex variables (y, siteCovs, obsCovs) into what is called 'member variables.' To access these variables, we need to use the '@' symbol instead of the '$' symbol when retrieving data within an object. We will see how this is used below.

First, use `str(fishcats)` to ensure that the data are inputted as required.

Now, let's plot the data to see if there are any preliminary patterns we can detect.

```
> plot(fishcats)
```

On the x-axis of the figure you have the different replicates, and on the y-axis are the 130 sites. The two colors represent detection and non-detection during a particular replicate and site.

What do you see? Was detection similar among censuses? Can you develop some hypotheses that might explain this pattern?

What if we had only gone in the field one time? What would one trip in the field suggest about occupancy? What is the value of multiple trips to each site?

Let's look at the proportion of sites occupied without accounting for non-detection. This is called the naïve occupancy estimate.

```
# fishcats@y is the detection history. Remember we used the
#apply function before (use ?apply to find out what it does
#if you have forgotten).

> occupied.sites=apply(fishcats@y, 2, sum)
```

Now we will calculate the proportion of occupied sites by simply dividing the number of sites where we detected cats at least once, by the total number of sites.

```
> (no.of.sites=nrow(fishcats@y))
> naive.occ1=occupied.sites/no.of.sites
> naive.occ1 # This is the naïve occupancy estimate for
each survey
> mean(naive.occ1)
```

Modeling occupancy: Dot Model

Now we are going to use these occupancy data to estimate two parameters: occupancy (ψ) and detection (p).

We are going to use the function `occu()`, which is used to estimate occupancy and detection in a single season occupancy model in the library `unmarked`.
Take a look at the help file by typing

```
> ?occu
```

The first model that we are going to run does not use covariates; that is, the estimation is solely based on the occupancy history for each site (the ones you just plotted).

The `occu` function requires the covariates for each parameter (detection and occupancy). For now, as we are not going to use covariates, we will just use `~1` as input. Later, we will denote the covariates for detection and then the covariates for occupancy. We will use the argument `data=` to specify where the data are stored, as in previous labs.

Note: It is always difficult to remember which model represents what. So usually we follow a convention**. *p*** represents **detection** and **psi** represents **occupancy**. A dot model

is a constant model. So we will call our first model pdot.psidot. This means we did not include any covariates in the model.

```
> pdot.psidot=occu(~1 ~1,data=fishcats)
> pdot.psidot
```

We have two estimates (and associated standard errors): one for occupancy and another for detection. Note that these do not represent probabilities because they are logit transformed (see the text box at the end of this document on the logit transformation). We discussed this in lecture, including why we use a logit transformation. We need to back-transform these logit-transformed values to get the actual estimates of the two parameters. To back-transform the estimates we are going to use the function `backTransform()`. This function needs the following arguments: (a) the model (`pdot.psidot` in this case) and (b) which variable we want to transform (occupancy probability-`"state"` or detection probability-`"det"`).

```
> backTransform(pdot.psidot,type="det") #estimate of "p"
> backTransform(pdot.psidot,type="state") #estimate of "psi
```

Now we can say that the detectability in this system is 0.359 and the proportion of sites occupied is 0.279.

What did you find? Did incorporating the detection probability change your inference of the proportion of sites occupied? Usually this becomes a big issue if we have low detection probabilities, or few replicates.

Modeling occupancy: Including covariates for detection probability

Alright, so we just fitted a very simple occupancy model. We found out that if we incorporate detection probability, our inference of the proportion of sites occupied changes. Now let us try improving our model by using covariates. Covariates help to increase model fit by explaining more of the variance. For example, we expected *a priori* that baiting would increase our detection probability. Let us incorporate this covariate to estimate detection probability better.

```
#this model assumes that detection probability (p) is
effected by baiting and occupancy (psi) is constant

> pbait.psidot = occu(~Bait ~1,data=fishcats)
> pbait.psidot
```

Now take a look at the occupancy value using the function backtransform, as you did before. Has the value changed?

Because we are using covariates for detection probability in this model, we cannot directly backtransform – we need to specify a value for the covariate. We can do this in one of two ways:

1st way: Using backtransform.

```
# First we will see what the detection probability is for
no-baits (intercept=1, bait=0).

> backTransform(linearComb(pbait.psidot, coefficients =
c(1,0), type = 'det'))

# Now we will see what the detection probability is for
baited replicates (intercept=1, bait=1).

> backTransform(linearComb(pbait.psidot, coefficients =
c(1,1), type = 'det'))

> backTransform(pbait.psidot, type = 'state')
```

2nd method: using the function predict (I prefer this way).

To use the `predict` function, we will first create a new dataset, with two values of bait, 0 and 1.

```
# Note: what we are doing below is creating a dataframe
with one column. The name of the column is Bait, and it
contains two values, 0 and 1.
> newbait=data.frame(Bait=c(0,1))

# Now use the predict function, with the arguments:
(a) model, (b) new data, (c) the parameter (state or det)
and (d) an indication that you want the data (newbait) to
be appended to the predicted values.
> predictedbait=predict(pbait.psidot, newdata=newbait,
type='det', appendData=T)
```

```
> predictedbait
```

Notice that you have the predicted values and the upper and lower confidence intervals.

What are the detection and occupancy probabilities now? Have the estimates changed from the previous model?

We may be interested in plotting detection probability during baited and non-baited replicates. You can now use the predicted values to plot a bar plot, with confidence intervals (recall you did these in previous labs).

What can you say from this plot? How did baiting change the detection probability?

Model selection

We now have two models, one that assumes a constant detection probability, and one that models detection probability as a function of if the track-pads were baited. The two models result in a different set of estimates for both parameters of interest. Which model is correct?

To answer this question we are going to use model selection procedures. Usually adding covariates to a statistical model increases model fit because each covariate will explain a little bit more of the variance. Nevertheless, if you have a model with too many covariates the model may over-fit the data and lose generality. How do we balance between model generality and fit?

The information theoretical approach is commonly used to do model selection via the minimization of AIC values. AIC (Aikake information criterion) is defined as:

$$\text{AIC} = 2k - 2\ln(L)$$

where k is the number of parameters in the model and L is the likelihood. Under this paradigm, a model with the smallest AIC is the most parsimonious (preferred) model. Remember that your model needs to be biologically reasonable. If, for instance, we had elevation as a covariate and the model stated that fishing cats are only found above 3000 m, we know that there is something wrong with the model – we should check our data. Another example is the inclusion of covariates without a solid ecological foundation – for example, would you think that the density of monkeys in the region might influence the distribution of fishing cats?

The library unmarked has the function `modSel()` that can be used to do model selection. This function takes a list of models to compare. This list will have two components for each model: the name of the model and the object that we used to store the model. We are going to store the list of models in `fms`, and then use this object to do model selection.

```
> fms=fitList('p(.)psi(.)' = pdot.psidot,
                'p(Bait)psi(.)' = pbait.psidot)
> modSel(fms)
```

The result table will give you various outputs. `nPars` lists the number of parameters in each model. `delta` is the AIC value of each model minus the smallest AIC value. Therefore, the most parsimonious model will have a delta of 0. Note that the best model is the one with the bait covariate because it has the smaller AIC value.

ASSIGNMENT

For your assignment, use the same data to model occupancy based on covariates. We have three covariates for occupancy, distance to water, disturbance, and shrub density. Fit the following models. We will use bait as a covariate for detection probability for all models, as we found support for it above. For instance, the first model specifies that you need to include bait as a covariate for detection probability and distance to water as a covariate for psi. Notice that the last model includes two covariates for psi – distance to water and disturbance.

```
#here is some pseudo code for you to use
> pbait.psiwater = occu(~Bait ~Dist.water , data=fishcats)
> pbait.psidist = occu(~Bait ~Disturbance , data=fishcats)
> pbait.psishrub = occu(~Bait ~Shrub , data=fishcats)
> pbait.psiwaterdist = occu(~Bait ~Dist.water + Disturbance
, data=fishcats)
```

You can then name all the models and include them in one list to calculate the AIC values like this

```
> fms=fitList('p(.)psi(.)' = pdot.psidot,
>             'p(Bait)psi(.)' = pbait.psidot,
>             'p(Bait)psi(Water)' = pbait.psiwater,
>             'p(Bait)psi(Disturbance)' = #insert name here
```

```
>                'p(Bait)psi(Shrub)' = #insert name here
>                'p(Bait)psi(Water+Disturbance)' = #insert
> #name

> modSel(fms)
```

Sort the models (include all 6 models in the model list) by delta AIC and report the values in the following table. Include results from the two models you have already fitted.

Model	AIC	Delta AIC
p(Bait)psi(Dist.water+Disturbance)	407.56	0.00
p(Bait)psi(Disturbance)	411.42	3.86
p(Bait)psi(Dist.water)	414.22	6.66
p(Bait)psi(.)	420.57	13.00
p(Bait)psi(Shrub)	422.56	15.00
p(.)psi(.)	434.18	26.62

Q1. Which model would you select (based on biology and table you complete above)? How did you arrive at this conclusion? Use two criteria in your answer (AIC and biological reasonability).

Q2. Following the model with the lowest AIC, what is the detection probability in this system for baited and non-baited occasions? Report both the value for *p* and its corresponding standard error (fill in the shaded boxes). Compare the bait, no bait, and null models. Does bait improve the detection probability? I have given you pseudo code below for the "no covariates" model.

```
> nocovariates<-backTransform(pdot.psidot,type="det")
```

Bait	Parameter	SE	Model
Null	p= 0.359	0.0369	Null model
No Bait	p= 0.1880473	0.04615116	Best model
Bait	p= 0.4499513	0.04599445	Best model

Q3. Recall you started with two objectives: (a) to determine the proportion of area occupied by fishing cats and (b) what determines its spatial distribution. According to the best model, provide your conclusions, by answering the two sub-questions provided below (keep reading, there are hints at the end).

a. Report psi using an average value for distance to water, and each value of categorical covariates (disturbance category). Complete the table below to determine how disturbance influences fishing cat occupancy.

Disturbance	Psi
Low	
Med	
High	
VHigh	

b. How does distance to water influence occupancy probability, according to your best model? Include a plot of the relationship, with confidence intervals. Assume regions of medium disturbance.

Hint: The code below will help for part b. Do this part first, then follow the same logic to obtain the answer for part a.

```
# First generate potential values for distance to water.
From the summary information of the covariates, you can get
the range of this variable. You see that the distance to
water ranges from values of 0 to 110. Generate values
within this range using seq.
> water.values=seq(0,110,by=10)

#Now we will create a 'dist.values' object and set this
object to contain the same number of values as the length
of water.values and set all of the values equal to "Med"
(because we are assuming regions of medium disturbance)
> dist.values=rep("Med", length(water.values))

# Now we will specify that dist.values is a factor variable
with 4 possible levels.

> dist.values=factor(dist.values, levels=c("High", "Low",
"Med", "VHigh"))
> dist.values #check to make sure you only have "Med"
entered in the data frame, but that there are 4 possible
values that 'dist.values' could take
```

```
# Form your data frame of new values. You need to include
one column with Distance to water values, and a column for
Disturbance values. Notice the names of the column
specified below are the same as those in the data.

> newdata1=data.frame(Dist.water=water.values,
                      Disturbance=dist.values)

# Predict values from the new data
> predictedoccu=predict(pbait.psiwaterdist,
newdata=newdata1,type="state", appendData=T)
> predictedoccu

# Plot
> plot(predictedoccu$Predicted~predictedoccu$Dist.water,
type="l", ylim=c(0,1))
#(NOTE: type=l is a "L" not the number one)

# Include CI lines in the plot.
> lines(predictedoccu$lower~predictedoccu$Dist.water,
lty=2)
> lines(predictedoccu$upper~predictedoccu$Dist.water,
lty=2)
```

Hint for part a: Use the same general structure of the code above for part a. Remember that for this part, you need *one* value for water.values, and *four* values of dist.values. Use mean(fishcats$Dist.water) as your water.value.

Logit Transformation

In a traditional regression you have

$$y = \propto + \beta x$$

where *x* and *y* are your predictor and response variables respectively, β represents the slope and $\propto$ the intercept. This will only work if *y* is continuous. What happens when our predictors are binary (i.e., 1 or 0)? In this case the response variance is usually logit transformed when fitting a logistic regression.

$$\text{logit}(y) = \propto + \beta x$$

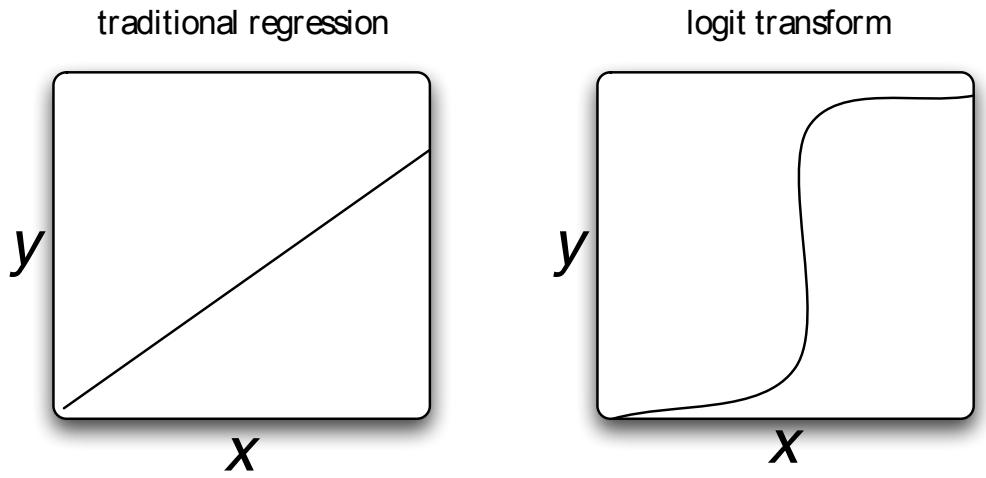
where y now is a binary variable. Recall that in the context of occupancy models *x* represents a covariate for occupancy or detection.

The logit transformation is very convenient because it allows us to use the same right hand side of the regression equation.

The logit transformation is defined as

$$\text{logit}(y) = \log\left(\frac{y}{1-y}\right)$$

where again *y* is a binary number.



What is important to remember is that when you run a logistic regression, you will get your predictions logit transformed. In order for them to become probabilities, you need to back-transform them using

$$\text{logit}^{-1}(u) = \frac{e^u}{e^u + 1}$$