CSP4141\_Tuesday\_SS\_Model

Tuesday, May 22nd, 2018 Introduction to Single Season- Single Species Occupancy Model

First load the required packages. I suggest loading the ‘Environmeterics’ Task view (“CRAN Task View: Analysis of Ecological and Environmental Data”). It loads a bunch of ecology/biology relevant packages all at once. It does take quite a while to download all of the packages. So I won’t do it now. Code below.

install.packages(“ctv”)

library(“ctv”)

install.views(“Environmetrics”)

We are going to simulate data according to the MacKenzie model. One of the covariates that we want to consider is vegetation height. So we need to generate some data for that. We’ll constrain it to a uniform distribution bounded by 1 and 3. We will use ‘set.seed’ so that we all get the same values generated.

Suppose that occupancy probability actually increases with vegetation height in this example. The relationship is described by an intercept of -3 and a slope parameter of 2 on the logit scale. (plogis is the inverse-logit (constrains us back to the [0-1] scale))

**library**(unmarked) *#load unmarked*

## Warning: package 'unmarked' was built under R version 3.3.3

## Loading required package: reshape

## Loading required package: lattice

## Loading required package: parallel

## Loading required package: Rcpp

nSites <- 100 *#number of sites in this simulated example*  
**set.seed**(443)  
vegHt <- **runif**(nSites, 1, 3) *#create the vegetation height variable*  
psi <- **plogis**(-3 + 2\*vegHt) *#In this example occupancy is related to veg height*

We will set the true occupancy state of these 100 sites…

z <-**rbinom**(nSites, 1, psi) *#define latent Occupancy for our sites*

We are also interested in a covariate, wind. So we need to simulate some data for that.

nVisits <-3  
wind<-**array**(**rnorm**(nSites\*nVisits),dim = **c**(nSites, nVisits))  
p<-**plogis**(1-2\*wind)  
y<-**matrix**(NA, nSites, nVisits)  
for(i in 1:nSites) {  
 y[i, ]<-**rbinom**(nVisits, z[i], p[i, ])  
}  
  
**head**(**cbind**(z=z, y1=y[,1], y2=y[,2], y3=y[,3])) *#take a peek*

## z y1 y2 y3  
## [1,] 0 0 0 0  
## [2,] 1 0 1 1  
## [3,] 1 1 1 0  
## [4,] 1 1 1 0  
## [5,] 0 0 0 0  
## [6,] 1 1 0 0

Now that we have a dataset to work with, we can format those data for unmarked and summarize and inspect structure using the ‘str’ call. Recall that for a Single season occupancy model in unmakred, the data has to be in an ‘unmarkedFrameOccu’

umf <- **unmarkedFrameOccu**(y=y, siteCovs=**data.frame**(vegHt=vegHt), obsCovs=**list**(wind=wind))  
**summary**(umf)

## unmarkedFrame Object  
##   
## 100 sites  
## Maximum number of observations per site: 3   
## Mean number of observations per site: 3   
## Sites with at least one detection: 66   
##   
## Tabulation of y observations:  
## 0 1 <NA>   
## 159 141 0   
##   
## Site-level covariates:  
## vegHt   
## Min. :1.010   
## 1st Qu.:1.562   
## Median :1.947   
## Mean :1.976   
## 3rd Qu.:2.386   
## Max. :2.987   
##   
## Observation-level covariates:  
## wind   
## Min. :-2.730969   
## 1st Qu.:-0.718048   
## Median :-0.000248   
## Mean :-0.044961   
## 3rd Qu.: 0.600647   
## Max. : 3.179049

**str**(umf)

## Formal class 'unmarkedFrameOccu' [package "unmarked"] with 5 slots  
## ..@ y : int [1:100, 1:3] 0 0 1 1 0 1 0 0 1 1 ...  
## ..@ obsCovs :'data.frame': 300 obs. of 1 variable:  
## .. ..$ wind: num [1:300] -0.136 0.761 0.921 1.183 -0.605 ...  
## ..@ siteCovs:'data.frame': 100 obs. of 1 variable:  
## .. ..$ vegHt: num [1:100] 2.27 2.54 2.44 2.51 1.2 ...  
## ..@ mapInfo : NULL  
## ..@ obsToY : num [1:3, 1:3] 1 0 0 0 1 0 0 0 1 #obsToY is an optional matrix

# specifying relationship between observation-level covariates and response matrix

The data are now in the correct format for unmarked, so we can fit models. Recall that the fitting function for a single season occupancy model in unmarked is ‘occu’ Detection covariates follow first tilde, then come occupancy covariates. Note: Estimates are on logit scale

fm.occu0 <- **occu**(~1~1, data=umf) *#naive model with no predictors for detection or occupancy probabilities*  
fm.occu1 <- **occu**(~1 ~vegHt, data=umf) *#no covariate for det prob but using vegetation height for occ prob*  
fm.occu2<-**occu**(~wind~1, data=umf) *#wind covariate for det prob but no covariate for occu prob*  
fm.occu3<-**occu**(~wind~vegHt, data=umf) *#wind cov for det prob and veg height for occu prob*

We’ve fit the models above. The first model has no covariates for either occupancy or detection. The second has a covariate that may influence occupancy probability, vegetation height.The third model include a wind covariate that may be influencing detection probability but nothing for occupancy probability. The fourth model includes a covariate (wind) that we think may be influencing detection probability and a covariate (veg height) that we think may be influencing occupancy probability.

Now we can analyze the results. First, let’s look at the output.

**summary**(fm.occu0)

##   
## Call:  
## occu(formula = ~1 ~ 1, data = umf)  
##   
## Occupancy (logit-scale):  
## Estimate SE z P(>|z|)  
## 0.754 0.227 3.32 0.000886  
##   
## Detection (logit-scale):  
## Estimate SE z P(>|z|)  
## 0.805 0.17 4.74 2.11e-06  
##   
## AIC: 366.3688   
## Number of sites: 100  
## optim convergence code: 0  
## optim iterations: 13   
## Bootstrap iterations: 0

**summary**(fm.occu1)

##   
## Call:  
## occu(formula = ~1 ~ vegHt, data = umf)  
##   
## Occupancy (logit-scale):  
## Estimate SE z P(>|z|)  
## (Intercept) -2.67 1.02 -2.61 0.00911  
## vegHt 1.81 0.56 3.24 0.00118  
##   
## Detection (logit-scale):  
## Estimate SE z P(>|z|)  
## 0.8 0.17 4.7 2.66e-06  
##   
## AIC: 354.216   
## Number of sites: 100  
## optim convergence code: 0  
## optim iterations: 17   
## Bootstrap iterations: 0

**summary**(fm.occu2)

##   
## Call:  
## occu(formula = ~wind ~ 1, data = umf)  
##   
## Occupancy (logit-scale):  
## Estimate SE z P(>|z|)  
## 0.736 0.221 3.34 0.000842  
##   
## Detection (logit-scale):  
## Estimate SE z P(>|z|)  
## (Intercept) 1.51 0.285 5.31 1.11e-07  
## wind -2.83 0.432 -6.55 5.57e-11  
##   
## AIC: 258.0051   
## Number of sites: 100  
## optim convergence code: 0  
## optim iterations: 17   
## Bootstrap iterations: 0

**summary**(fm.occu3)

##   
## Call:  
## occu(formula = ~wind ~ vegHt, data = umf)  
##   
## Occupancy (logit-scale):  
## Estimate SE z P(>|z|)  
## (Intercept) -2.89 1.005 -2.87 0.004088  
## vegHt 1.93 0.544 3.55 0.000392  
##   
## Detection (logit-scale):  
## Estimate SE z P(>|z|)  
## (Intercept) 1.50 0.284 5.27 1.38e-07  
## wind -2.83 0.428 -6.63 3.44e-11  
##   
## AIC: 243.8544   
## Number of sites: 100  
## optim convergence code: 0  
## optim iterations: 37   
## Bootstrap iterations: 0

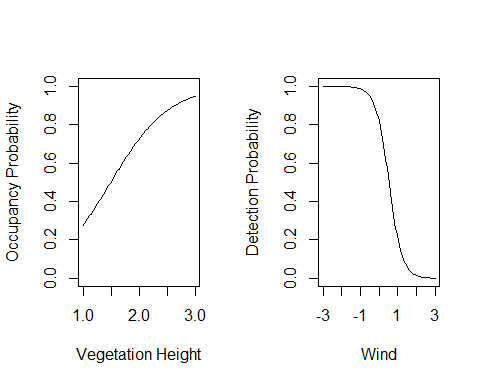
Now let’s do some model selection. We’ll look at the AIC values and ranking of the four models that we ran.

fms <- **fitList**(m1=fm.occu0, m2=fm.occu1, m3=fm.occu2, m4=fm.occu3)  
**modSel**(fms)

## nPars AIC delta AICwt cumltvWt  
## m4 4 243.85 0.00 1.0e+00 1.00  
## m3 3 258.01 14.15 8.4e-04 1.00  
## m2 3 354.22 110.36 1.1e-24 1.00  
## m1 2 366.37 122.51 2.5e-27 1.00

When there are no covariates, we can backtransform using the ‘backTransform’ call. When covariates are present we can do something like:

**par**(mfrow=**c**(1,2))  
beta1<-**coef**(fm.occu3)  
**plot**(function(x) **plogis**(beta1[1] + beta1[2]\*x), 1, 3,  
 xlab="Vegetation Height", ylab="Occupancy Probability", ylim=**c**(0,1))  
**plot**(function(x) **plogis**(beta1[3] + beta1[4]\*x), -3, 3,  
 xlab="Wind", ylab="Detection Probability", ylim=**c**(0,1))



You can use the ‘ranef’ call to look at the random effects of the best model.

**ranef**(fm.occu3)

## Mean Mode 2.5% 97.5%  
## [1,] 2.267932e-01 0 0 1  
## [2,] 1.000000e+00 1 1 1  
## [3,] 1.000000e+00 1 1 1  
## [4,] 1.000000e+00 1 1 1  
## [5,] 3.344834e-04 0 0 0  
## [6,] 1.000000e+00 1 1 1  
## [7,] 1.030669e-04 0 0 0  
## [8,] 6.877760e-02 0 0 1  
## [9,] 1.000000e+00 1 1 1  
## [10,] 1.000000e+00 1 1 1  
## [11,] 1.000000e+00 1 1 1  
## [12,] 1.000000e+00 1 1 1  
## [13,] 1.000000e+00 1 1 1  
## [14,] 1.000000e+00 1 1 1  
## [15,] 5.270097e-05 0 0 0  
## [16,] 8.024283e-05 0 0 0  
.

.

.