NWT CONI results

This doc describes methods to estimate density from recognizer based detections.

## Logic

#### I. Outcomes at the site level:

1a) a site is unsuitable: we observe 0 1b) a site is suitable: we can observe 0 or >0

2a) a site is unoccupied (0 home range overlaps with it): we observe a 0 2b) if the site is occupied (at least one home range overlaps with the location): we observe 0 or >0

Levels 1 and 2 is of interest, this describes the population and we have some ideas about the distribution of overlapping home ranges based on the 10% sample. Up to this point we are looking at a zero-inflated Poisson distribution where we have lambda (expected value of overlapping home ranges) and delta (proportion of suitable sites), and our population size estimate is delta\*lambda. But there is a bit more complexity. Although we have an idea of lambda (10% sample), delta is confounded with other sources of 0s.

#### II. Outcomes at the visit level:

3a) CONI is not active: we observe a 0

3b) CONI is active: we observe a 0 or >0 –> we estimated this conditional probability from survical models as a function of dat/time (can use it as offset)

4a) a site is not used (occupied but nobody home): we observe 0

4b) a site is used (it is occupied, but individuals are in and out): we observe 0 or >0

(The 5th layer is detectability but we use our EDR estimate to standardize for effective area and ignore this component in the modeling otherwise.)

The only thing that bothers me is depending on the spacing of the sites and home range sizes, we are estimating the population size or the superpopulation size (with double counting the same individuals if sites are too close). Estimates of home range sizes might help thin the data to avoid double counting or at least to conduct a sensitivity analysis.

## True (latent) occupancy and abundance

We use the 10% validated data from ECK.

Randomly selected 10% of the recordings scanned. Filtered out recordings that had booms detected in them. Reviewed all recognizer detections for those recordings. Assigned individual ID to each boom detection in those recordings. There’s definitely more than one individual in some of these recordings. Here’s the distribution of the results:

* 1 CONI: 29 sites
* 2 CONI: 11 sites
* 3 CONI: 2 sites
* 4 CONI: 1 site The one thing to keep in mind is that these are almost entirely from the Alberta fire site. I suspect the densities will be lower at the other province/treatment combos.

Fitting a conditional likelihood model to home range overlap data. This follows Solymos et al. 2012 (Environmetrics).

**Conditional maximum likelihood** — Let be a random variable, and are observations. A zero inflated (ZI) distribution with non-zero inflated density function can be written as ; ; , where is a vector of model parameters, not including the ZI parameter that is the probability of observing 0 as part of the ZI process.

Now the probability mass function for the counts can be written as . The denominator can also be written as Thus .

This conditional mass function can be used in estimating conditional maximum likelihood estimates of based on the non-zero part of the data ().

#### Poisson

#### Negative Binomial

is Gamma variance in the Poisson-Gamma mixture parametrization (; ; ):

This is all (and more) implemented in the zi.fit function. We don’t bother with offsets because this was based on looking at multiple visits and counting total number of inds that used that site over the entire time

## number of inds at each site based on 10% of all sites by ECK  
Y1 <- rep(1:4, c(29, 11, 2, 1)) # mean 1.418605  
X1 <- data.matrix(rep(1, length(Y1)))  
Z1 <- X1  
  
cl0p <- zi.fit(Y1, X1, Z1, distr="pois", type="CL", hessian=TRUE)$CL  
cl0nb <- zi.fit(Y1, X1, Z1, distr="negbin", type="CL", hessian=TRUE)$CL  
  
## calculate AIC  
logLik.default <- function(object, ...)   
 structure(object$loglik,  
 df = length(object$coef),  
 class = "logLik")  
ic <- AIC(cl0p, cl0nb)  
ic$BIC <- AIC(cl0p, cl0nb, k=log(length(Y1)))$AIC  
ic$AICc <- ic$AIC + (2\*ic$df^2+2\*ic$df) / (length(Y1)-ic$df-1)  
ic # Poisson is better supported

## df AIC BIC AICc  
## cl0p 1 75.38446 77.14566 75.48202  
## cl0nb 2 77.37153 80.89393 77.67153

## mean of the poisson (including 0 counts and offsets too)  
(lambda <- exp(cl0p$coef))

## [1] 0.7454628

## P(N=0) based on Poisson, which is the suitable but unoccupied probability  
exp(-lambda)

## [1] 0.4745146

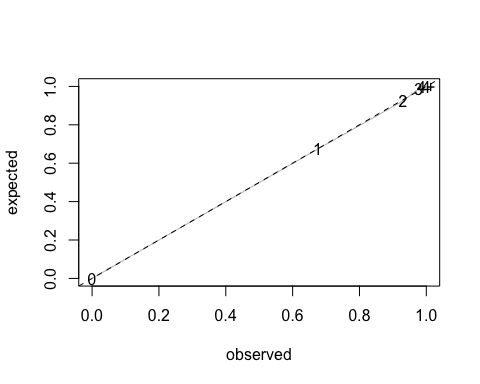
## suitable and unoccupied  
1-exp(-lambda)

## [1] 0.5254854

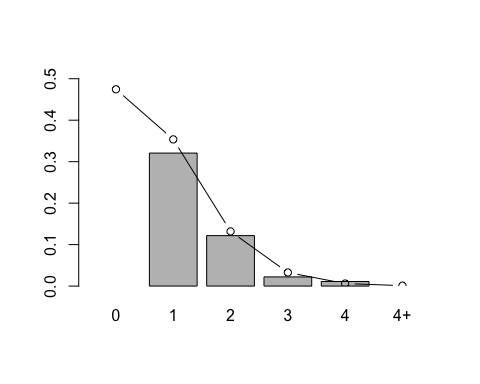
cnt <- 0:(max(Y1)+1)  
dp <- dpois(cnt, exp(cl0p$coef))  
names(dp) <- cnt  
names(dp)[length(dp)] <- paste0(cnt[length(dp)]-1, "+")  
dp[length(dp)] <- 1-sum(dp[-length(dp)])  
  
tab <- cbind(  
 observed=c(NA, (1-sum(dp[-c(1, length(dp))]))\*table(Y1)/length(Y1), NA),  
 expected=dp)  
rownames(tab) <- names(dp)  
round(tab, 3)

## observed expected  
## 0 NA 0.475  
## 1 0.321 0.354  
## 2 0.122 0.132  
## 3 0.022 0.033  
## 4 0.011 0.006  
## 4+ NA 0.001

tmp <- tab[-1,]  
tmp[is.na(tmp)] <- 0  
cs <- rbind("0"=c(0,0), apply(tmp, 2, cumsum))  
cs[,1] <- cs[,1]/max(cs[,1])  
cs[,2] <- cs[,2]/max(cs[,2])  
plot(cs,type="l", col="grey", xlim=c(0,1), ylim=c(0,1))  
text(cs, labels = rownames(tab))  
abline(0,1,lty=2)



tmp <- barplot(tab[,1], ylim=c(0, 0.5))  
lines(tmp, tab[,2], type="b")



y <- matrix(NA, nlevels(dat$station), 30)  
rownames(y) <- levels(dat$station)  
p <- y  
for (i in rownames(y)) {  
 tmp <- dat[dat$station == i,]  
 tmp <- tmp[order(as.character(tmp$datetime)),]  
 y[i, seq\_len(nrow(tmp))] <- tmp$detection  
 p[i, seq\_len(nrow(tmp))] <- tmp$p  
}  
dat2 <- nonDuplicated(dat, station, TRUE)  
dat2 <- dat2[rownames(y),]  
dat2$lc2 <- dat2$lc  
levels(dat2$lc2) <- c("Con", "Con", "DMO", "DMO")  
table(dat2$lc, dat2$lc2)

##   
## Con DMO  
## ConTmp 108 0  
## ConTai 155 0  
## DM 0 105  
## OP 0 109

M <- nrow(y)  
J <- ncol(y)  
X <- matrix(1, M, 1)  
Z <- matrix(1, M\*J, 1)  
X2 <- model.matrix(~lc2, dat2)  
Z2 <- NULL  
for (i in seq\_len(ncol(y))) {  
 Z2 <- rbind(Z2, X2)  
}  
  
# constant  
method <- "Nelder-Mead"  
o0 <- mvocc(y, X, Z, p, lambda, method=method)  
o1 <- mvocc(y, X2, Z, p, lambda, method=method)  
o2 <- mvocc(y, X, Z2, p, lambda, method=method)  
o3 <- mvocc(y, X2, Z2, p, lambda, method=method)  
  
aic <- AIC(o0,o1,o2,o3)  
aic$BIC <- BIC(o0,o1,o2,o3)$BIC  
aic$AICc <- aic$AIC + (2\*aic$df^2+2\*aic$df) / (nobs(o0)-aic$df-1)  
aic$dAICc <- aic$AICc - min(aic$AICc)  
aic

## df AIC BIC AICc dAICc  
## o0 2 4242.521 4250.856 4242.546 186.28163  
## o1 3 4146.892 4159.394 4146.943 90.67802  
## o2 3 4124.953 4137.455 4125.003 68.73881  
## o3 4 4056.180 4072.850 4056.265 0.00000

summary(o3)

## Estimate Std. Error z value Pr(>|z|)  
## Intercept 45.259 1060.563 0.0427 0.9660  
## lc2DMO -46.508 1060.563 -0.0439 0.9650  
## Intercept 16.191 1060.562 0.0153 0.9878  
## lc2DMO -16.716 1060.563 -0.0158 0.9874

plogis(o0$coef) # delta (prob of nonzero) and q (prob of use)

## X0 Z0   
## 0.6360655 0.9710858

delta <- plogis(o0$coef[1])  
q <- plogis(o0$coef[2])  
  
# cond probs  
delta # suitable

## X0   
## 0.6360655

1-exp(-lambda) # occupied

## [1] 0.5254854

mean(p, na.rm=TRUE) # active

## [1] 0.1961374

q # used

## Z0   
## 0.9710858

# pop size at the sites  
delta\*lambda \* nlevels(dat$station)

## X0   
## 226.1758

# density  
delta\*lambda/dat$A[1] # inds/ha

## X0   
## 0.03571869