Local minima in multistate capture-recapture models

## Model

We consider a state-space formulation of multistate capture-recapture model. We refer to [Gimenez et al. (2007). State-space modelling of data on marked individuals. Ecological Modelling 206: 431-438](https://dl.dropboxusercontent.com/u/23160641/my-pubs/Servantyetal2010Ecology.pdf) and [Servanty et al. (2010). Assessing whether mortality is additive using marked animals: a Bayesian state-space modeling approach. Ecology 91: 1916–1923](https://dl.dropboxusercontent.com/u/23160641/my-pubs/Gimenezetal2007EcologicalModelling.pdf) for the theory.

## Read in data

I use female roe deer data monitored over 20 years and provided by J.-M. Gaillard. Population in Trois Fontaine managed by ONCFS. Two states, with and without fawn.

his <- matrix(c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,1,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,2,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,2,2,0,1,1,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,2,1,2,0,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,1,1,1,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,0,2,2,0,2,1,2,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,1,0,0,2,0,1,2,2,0,1,  
0,0,0,0,0,0,0,0,0,0,0,1,2,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,1,0,1,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,2,2,1,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,2,1,2,0,1,1,1,0,0,1,  
0,0,0,0,0,0,0,0,0,0,0,2,2,0,2,2,2,2,1,0,1,  
0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,2,1,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,2,0,0,0,2,2,0,1,1,0,1,  
0,0,0,0,0,0,0,0,0,0,2,1,2,1,1,2,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,0,2,2,2,1,0,2,2,2,0,0,1,  
0,0,0,0,0,0,0,0,0,1,2,2,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,0,2,0,2,2,0,0,0,0,0,1,0,0,1,  
0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,1,0,0,2,2,0,2,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,2,0,0,2,1,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,0,2,1,1,0,2,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,1,0,0,0,0,0,2,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,2,2,0,2,2,0,0,2,0,2,0,0,0,0,1,  
0,0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,0,2,2,0,0,0,1,0,2,0,0,0,0,0,0,1,  
0,0,0,0,0,0,2,2,0,0,0,2,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,1,1,1,0,0,1,2,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,1,1,2,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,1,1,2,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,1,2,2,0,1,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,2,1,2,0,0,2,2,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,2,1,2,2,2,2,2,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,0,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,2,0,2,0,2,1,0,0,0,2,0,0,0,0,0,0,1,  
0,0,0,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,2,0,2,0,0,2,2,0,0,0,0,0,0,0,0,0,1,  
0,0,0,0,2,0,2,2,2,2,2,2,0,0,0,0,0,0,0,0,1,  
0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,2,2,0,2,2,0,2,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,2,2,2,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,0,2,2,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,2,2,2,1,2,0,0,0,2,2,0,1,0,0,0,0,0,0,1,  
0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,2,0,0,0,2,0,0,0,0,1,0,0,0,0,0,0,0,0,1,  
0,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,2,2,0,0,2,2,1,0,0,0,0,0,0,0,0,0,0,0,1,  
0,0,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,  
0,2,2,2,2,2,2,2,0,0,2,2,0,1,0,0,0,0,0,0,1,  
2,1,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1),byrow=T,ncol=21)

## Frequentist Analysis with Mark

Let's format the data to be analysed.

his = his[,-ncol(his)] # remove last column, these are the counts  
k = ncol(his) # nb of capture occasions   
n = nrow(his) # nb of individuals   
out = array(dim=n)   
for (i in 1:n){   
 y = (his[i,] > 0) \* his[i,]   
 out[i] = paste(y,collapse="")   
}   
capt.hist = data.frame(ch = out)

Load RMark package

library(RMark)

## This is RMark 2.2.0

Process data and create default design matrix

mstrata.processed=process.data(capt.hist,model="Multistrata")   
mstrata.ddl=make.design.data(mstrata.processed)

Define structure on parameters, all state dependent

# survival probability   
S.site = list(formula=~stratum)   
# detection probability   
p.site = list(formula=~stratum)   
# transition probs   
Psi.site = list(formula=~-1+stratum:tostratum)

Fit model

mstrata.mod = mark(mstrata.processed,mstrata.ddl,model.parameters=list(S=S.site,p= p.site,Psi=Psi.site),output=FALSE,silent=T,delete=T)

Let's have a look to the parameter estimates

mstrata.mod$results$real[c(1:5,195),]

## estimate se lcl ucl  
## S s1 g1 c1 a0 o1 t1 0.7894772 0.0407036000 6.988372e-01 0.8583663  
## S s2 g1 c1 a0 o1 t1 0.8624650 0.0484991000 7.377955e-01 0.9332233  
## p s1 g1 c1 a1 o1 t2 0.2503825 0.0381443000 1.831870e-01 0.3322017  
## p s2 g1 c1 a1 o1 t2 0.9999999 0.0001228101 1.731802e-297 1.0000000  
## Psi s1 to2 g1 c1 a0 o1 t1 0.3185984 0.0425212000 2.415652e-01 0.4070140  
## Psi s2 to1 g1 c1 a0 o1 t1 0.4814442 0.0457104000 3.933813e-01 0.5706752  
## fixed note  
## S s1 g1 c1 a0 o1 t1   
## S s2 g1 c1 a0 o1 t1   
## p s1 g1 c1 a1 o1 t2   
## p s2 g1 c1 a1 o1 t2   
## Psi s1 to2 g1 c1 a0 o1 t1   
## Psi s2 to1 g1 c1 a0 o1 t1

Now let's re-run the same model but changing the initial values

nb\_inits = 5 # five runs with diff inits  
set.seed(3) # to reproduce the results  
res\_dev = rep(NA,nb\_inits)  
res = list()  
for (i in 1:nb\_inits){  
 inits = rnorm(6,0,1)  
 mstrata.sim = mark(mstrata.processed,mstrata.ddl,model.parameters=  
 list(S=S.site,p= p.site,Psi=Psi.site),output=FALSE,silent=TRUE,delete=T,initial = inits)  
 res\_dev[i] = mstrata.sim$results$deviance  
 res[[i]] = mstrata.sim$results  
}

Display the deviance values

res\_dev

## [1] 674.9109 669.3424 669.3424 669.3424 674.9109

There seems to be a local minimum. Let's have a look to the estimates corresponding to the global minimum

res[[2]]$real[c(1:5,195),1:4]

## estimate se lcl ucl  
## S s1 g1 c1 a0 o1 t1 0.7894770 4.070360e-02 0.6988370 0.8583661  
## S s2 g1 c1 a0 o1 t1 0.8624653 4.849920e-02 0.7377955 0.9332236  
## p s1 g1 c1 a1 o1 t2 0.2503824 3.814430e-02 0.1831869 0.3322016  
## p s2 g1 c1 a1 o1 t2 1.0000000 6.492596e-05 0.9998727 1.0001272  
## Psi s1 to2 g1 c1 a0 o1 t1 0.3185984 4.252120e-02 0.2415651 0.4070139  
## Psi s2 to1 g1 c1 a0 o1 t1 0.4814445 4.571040e-02 0.3933816 0.5706754

What about the estimates with a local minimum?

res[[1]]$real[c(1:5,195),1:4]

## estimate se lcl ucl  
## S s1 g1 c1 a0 o1 t1 0.6927282 0.0688786000 5.445473e-01 0.8095590  
## S s2 g1 c1 a0 o1 t1 0.8360083 0.0221533000 7.878664e-01 0.8749588  
## p s1 g1 c1 a1 o1 t2 0.9999984 0.0003227151 3.387039e-162 1.0000000  
## p s2 g1 c1 a1 o1 t2 0.5358706 0.0358389000 4.653802e-01 0.6049563  
## Psi s1 to2 g1 c1 a0 o1 t1 0.7112871 0.0693975000 5.595391e-01 0.8269259  
## Psi s2 to1 g1 c1 a0 o1 t1 0.1324606 0.0212328000 9.608300e-02 0.1798703

## Graphical examination

We use a graph of profile deviance to examine local minima. Let's do it for survival first:

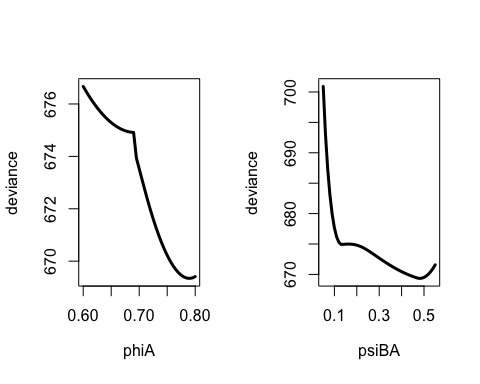
fr = mstrata.ddl$S$stratum  
ii = (fr==1)  
phi.fix = as.numeric(row.names(mstrata.ddl$S[ii,]))  
  
grid = seq(.6,0.8,0.01/2)  
#length(grid)  
dev = rep(NA,length(grid))  
ind = 1  
for (i in grid){  
 S.fix = list(formula=~stratum,fixed=list(index=phi.fix,value=i)) # survival depends on states   
 res.fix = mark(mstrata.processed,mstrata.ddl,model.parameters=list(S=S.fix,p=p.site,Psi=Psi.site),output = FALSE,delete=T,silent=T)  
 dev[ind] = res.fix$results$deviance  
 ind = ind + 1  
}  
dev\_phi = dev  
grid\_phi = grid

Then for a transition

fr = mstrata.ddl$Psi$stratum  
to = mstrata.ddl$Psi$tostratum  
ii = (fr==2 & to==1)  
psi.fix = as.numeric(row.names(mstrata.ddl$Psi[ii,])) # careful not to define Psi.fix twice  
  
grid = seq(.05,0.55,0.01)  
#length(grid)  
dev = rep(NA,length(grid))  
ind = 1  
for (i in grid){  
 Psi.fix = list(formula=~-1+stratum:tostratum,fixed=list(index=psi.fix,value=i),link='logit') # needs logit here, mlogit is a mess  
 res.fix = mark(mstrata.processed,mstrata.ddl,model.parameters=list(S=S.site,p=p.site,Psi=Psi.fix),output = FALSE,delete=T,silent=T)  
 dev[ind] = res.fix$results$deviance  
 ind = ind + 1  
}  
dev\_psi = dev  
grid\_psi = grid

Let's visualize the results

par(mfrow=c(1,2))  
plot(grid\_phi,dev\_phi,type='l',xlab=expression(phiA),ylab='deviance',col='black',lwd=3)  
plot(grid\_psi,dev\_psi,type='l',xlab=expression(psiBA),ylab='deviance',col='black',lwd=3)



## Bayesian analysis with Jags

Let us get the occasion of first capture for each individual:

get.first <- function(x) min(which(x!=0))  
f <- apply(his, 1, get.first)

Recode the data such that 1 is seen alive in A (with fawn), 2 is seen alive in B (without fawn), 3 = not seen:

his\_recoded <- his  
his\_recoded[his\_recoded==0] <- 3

Then we fit a multistate model:

{  
sink("state\_on\_survival.jags")  
cat("  
model {  
  
# -------------------------------------------------  
# Parameters:  
# phiA: survival probability site A  
# phiB: survival probability site B  
# psiAB: movement probability from site A to site B  
# psiBA: movement probability from site B to site A  
# pA: recapture probability site A  
# pB: recapture probability site B  
# -------------------------------------------------  
# States (S):  
# 1 alive at A  
# 2 alive at B  
# 3 dead  
# Observations (O):   
# 1 seen at A   
# 2 seen at B  
# 3 not seen  
# -------------------------------------------------  
  
# Priors  
 phiA ~ dunif(0, 1)  
 phiB ~ dunif(0, 1)  
 psiAB ~ dunif(0, 1)  
 psiBA ~ dunif(0, 1)  
 pA ~ dunif(0, 1)  
 pB ~ dunif(0, 1)  
  
# Define state-transition and observation matrices  
for (i in 1:nind){   
 # Define probabilities of state S(t+1) given S(t)  
 for (t in f[i]:(n.occasions-1)){  
 ps[1,i,t,1] <- phiA \* (1-psiAB)  
 ps[1,i,t,2] <- phiA \* psiAB  
 ps[1,i,t,3] <- 1-phiA  
 ps[2,i,t,1] <- phiB \* psiBA  
 ps[2,i,t,2] <- phiB \* (1-psiBA)  
 ps[2,i,t,3] <- 1-phiB  
 ps[3,i,t,1] <- 0  
 ps[3,i,t,2] <- 0  
 ps[3,i,t,3] <- 1  
   
 # Define probabilities of O(t) given S(t)  
 po[1,i,t,1] <- pA  
 po[1,i,t,2] <- 0  
 po[1,i,t,3] <- 1-pA  
 po[2,i,t,1] <- 0  
 po[2,i,t,2] <- pB  
 po[2,i,t,3] <- 1-pB  
 po[3,i,t,1] <- 0  
 po[3,i,t,2] <- 0  
 po[3,i,t,3] <- 1  
 } #t  
 } #i  
  
# Likelihood   
for (i in 1:nind){  
 # Define latent state at first capture  
 z[i,f[i]] <- y[i,f[i]]  
 for (t in (f[i]+1):n.occasions){  
 # State process: draw S(t) given S(t-1)  
 z[i,t] ~ dcat(ps[z[i,t-1], i, t-1,])  
 # Observation process: draw O(t) given S(t)  
 y[i,t] ~ dcat(po[z[i,t], i, t-1,])  
 } #t  
 } #i  
}  
",fill = TRUE)  
sink()  
}

Function to create known latent states z

known.state.ms <- function(ms, notseen){  
 # notseen: label for ënot seení  
 state <- ms  
 state[state==notseen] <- NA  
 for (i in 1:dim(ms)[1]){  
 m <- min(which(!is.na(state[i,])))  
 state[i,m] <- NA  
 }  
 return(state)  
 }

Function to create initial values for unknown z

ms.init.z <- function(ch, f){  
 for (i in 1:dim(ch)[1]){ch[i,1:f[i]] <- NA}  
 states <- max(ch, na.rm = TRUE)  
 known.states <- 1:(states-1)  
 v <- which(ch==states)  
 ch[-v] <- NA  
 ch[v] <- sample(known.states, length(v), replace = TRUE)  
 return(ch)  
 }

Create a list of data

jags.data <- list(y = his\_recoded, f = f, n.occasions = dim(his\_recoded)[2], nind = dim(his\_recoded)[1], z = known.state.ms(his\_recoded, 3))

Initial values

inits <- function(){list(phiA = runif(1, 0, 1), psiAB = runif(1, 0, 1), pA = runif(1, 0, 1), z = ms.init.z(his\_recoded, f))}

Parameters to be monitored

parameters <- c("phiA","phiB", "psiAB", "psiBA", "pA", "pB")

MCMC settings

ni <- 5000 # nb iter  
nt <- 6 # thinning  
nb <- 1000 # burn-in  
nc <- 2 # nb chains

Call JAGS from R

library(R2jags)

## Loading required package: rjags

## Loading required package: coda

## Linked to JAGS 4.2.0

## Loaded modules: basemod,bugs

##   
## Attaching package: 'R2jags'

## The following object is masked from 'package:coda':  
##   
## traceplot

ms <- jags(jags.data, inits, parameters, "state\_on\_survival.jags", n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb)

## module glm loaded

## Compiling model graph  
## Resolving undeclared variables  
## Allocating nodes  
## Graph information:  
## Observed stochastic nodes: 1139  
## Unobserved stochastic nodes: 769  
## Total graph size: 27933  
##   
## Initializing model

print(ms, digits = 3)

## Inference for Bugs model at "state\_on\_survival.jags", fit using jags,  
## 2 chains, each with 5000 iterations (first 1000 discarded), n.thin = 6  
## n.sims = 1334 iterations saved  
## mu.vect sd.vect 2.5% 25% 50% 75% 97.5% Rhat  
## pA 0.335 0.139 0.202 0.256 0.293 0.353 0.779 1.019  
## pB 0.842 0.126 0.545 0.765 0.872 0.944 0.995 1.026  
## phiA 0.769 0.051 0.657 0.741 0.773 0.805 0.853 1.002  
## phiB 0.856 0.040 0.783 0.829 0.854 0.882 0.939 1.005  
## psiAB 0.389 0.099 0.260 0.327 0.364 0.425 0.671 1.017  
## psiBA 0.391 0.096 0.161 0.340 0.412 0.462 0.529 1.025  
## deviance 649.793 43.561 567.810 616.549 650.741 684.022 724.738 1.028  
## n.eff  
## pA 82  
## pB 62  
## phiA 1300  
## phiB 390  
## psiAB 96  
## psiBA 64  
## deviance 60  
##   
## For each parameter, n.eff is a crude measure of effective sample size,  
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).  
##   
## DIC info (using the rule, pD = var(deviance)/2)  
## pD = 933.6 and DIC = 1583.4  
## DIC is an estimate of expected predictive error (lower deviance is better).

Plot of the transition to illustrate local minimum

par(cex.main = 1.5, mar = c(5, 6, 4, 5) + 0.1, mgp = c(3.5, 1, 0), cex.lab = 1.5 , font.lab = 2, cex.axis = 1.3, bty = "n", las = 1)  
h <- hist(ms$BUGSoutput$sims.matrix[,7],breaks=20,freq=F,main="",xlab="",ylab="",col="gray")  
axis(1, seq(0, .60, by = .1))  
axis(2, labels = FALSE, lwd.ticks = 0)  
rug(jitter(ms$BUGSoutput$sims.matrix[,7]))  
mtext("transition B->A", side = 1, line = 2.5, cex = 1.5, font = 2)  
mtext("Density", side = 2, line = 2.5, cex = 1.5, font = 2, las = 0)

