Simulated occupancy image analysis for Murray et al.

Mason Fidino

6/26/2020

To keep things tractable we will simulate data for a single season at $n = 300$ sites. Finally, we assume that there a total of $J = 4$ repeat visits to each site within this single sampling season (e.g., four weeks of camera trapping).

```r
# Set seed for reproducibility
set.seed(-565)

# The number of sites for this example
nsite <- 300

# The number of visits to each site
j <- 4
```

The Occupancy model

This is for the first part of the model. In our own analysis, this part of the analysis estimates the proability of coyote occupancy, mangy or otherwise, as well as the probability of detecting coyote, mangy or otherwise, on one of the repeat visits assuming they are present. More generally, the log-odds a species is detected at s in $1, ..., S$ sites is $logit(\psi_s) = \boldsymbol{a}^T \boldsymbol{x_s}$, where $\boldsymbol{a}$ is a vector of parameters and $\boldsymbol{X}$ is row s of a conformable matrix of covariates where the first column is all $1's$ to account for the intercept. For this simulation we will use one covariate on site occupancy. Thus, we will need to specify an intercept (a_0) and slope term (a_1) to estimate from the simulated data.

$$logit(\psi_s) = \boldsymbol{a}^T \boldsymbol{x_s} = a_0 x_{s,1} + a_1 x_{s,2} = 0.5 - 1x_s$$

With our environmental covariate ($\boldsymbol{x}$) and these logit-scaled parameters we can simulate the occupancy status of this species at the S sites.

```r
# An environmental covariate for occupancy
x1 <- rnorm(nsite)

# Occupancy  logit-linear predictor
z_det <- 0.5 - 1 * x1

# Probability of occupancy
z_prob <- plogis(
  z_det
)

# True occupancy status of species
```

```r
z <- rbinom(
  nsite,
  1,
  z_prob
)
```

If you have been following along then this species should be at `sum(z) == 180` sites.
We do not perfectly observe the occupancy status of this species. We will let the log-odds we detect this species to vary along one covariate.

$$logit(\rho_s) = \boldsymbol{b^T} \boldsymbol{x_s} = b_0 x_{s,1} + b_1 x_{s,2} = -0.5 + 0.5 x_s$$

```r
# A different covariate for detection
x2 <- rnorm(
  nsite
)

# Logit-linear predictor
y_det <- -0.5 + 0.5 * x2

# The probability of detection given presence
y_prob <- plogis(
  y_det
)

# The observed data
#  Detection probability is 0 if species is not present
y <- rbinom(
  nsite,
  j,
  y_prob * z
)
```

Thus, in this simulation we failed to detect this species at `sum(z) - sum(y>0) == 29` sites.
That is all there is to the first part of the model. It's just a standard occupancy model and could be traded out for literally any other style of occupancy model you may be interested in using.

The conditional by-image model

In our analysis this is the conditional mange model, as that is the additional state we are interested in estimating. More generally, all that is needed is for there to be an additional state that can be observed on the images of a target species. The latent-state part of this model is very similar to the latent-state of the occupancy model except it can only happen if the species is present at the site. We cannot view a 'coyote with mange' if 'coyote, mangy or otherwise' are not present. The log-odds of this latent state is

$$logit(\omega_s) = \boldsymbol{c^T} \boldsymbol{x_s} = c_0 x_{s,1} + c_1 x_{s,2} = -0.5 + 0.5 x_s$$

```r
# Mange logit-linear predictor.
#  We will use the same covariate as we did with the latent
```

```
#  occupancy state.
w_det <- -0.5 + 0.5 * x1

# the probability a coyote is mangy at the site
w_prob <- plogis(
  w_det
)

# The mange status of coyote across the sites
w <- rbinom(
  nsite,
  1,
  w_prob * z
)
```

The observation model here substantially varies from a standard occupancy model because each site has a different number of images. We know that every site we detected the species will have at least one image. As we are going to iterate through all of the images we simply need a way to link an image to the appropriate site within the model. Likewise, for this simulation we will generate a different number of images per site.

```
# Step 1. figure out where we collected photos
sites_with_photos <- which(
  y>0
)

# Step 2. simulate number of photos per site
photos_per_site <- sample(
  1:30,
  length(sites_with_photos),
  replace = TRUE
)

# This is the total number of images
n_photos <- sum(
  photos_per_site
)
```

Now that we have the images, we are going to introduce uncertainty in our ability to detect a mangy coyote in an image given that mangy coyote are present at a site. Let the log-odds we detect a mangy coyote in image i be:

$$logit(\gamma_i) = \boldsymbol{d^T x_i} = d_0 x_{i,1} + d_1 x_{i,2} + d_2 x_{i,3} = -0.5 + 0.25 x_{i,2} + 0.7 x_{i,3}$$

Now, we need _to come up with a way to link images to sites. We can use a vector of length `n_photos` where each element is an integer that represents what site image i belongs to.

```
# This is a data.frame where each row has info on the sites
#  with images and the number of images at that site.
my_sites <- data.frame(
  sites = sites_with_photos,
  count = photos_per_site
)

# This is the site index. It is of length n_photos. Each
#  element represents the site an image belongs to.
site_idx <- rep(
```

```
  my_sites$sites,
  my_sites$count
)
```

The vector `site_idx` can be used to link the images to sites.

```
# Covariates, one binary, one continuous
x3 <- rnorm(
  n_photos
)

x4 <- rbinom(
  n_photos,
  1,
  0.3
)

# Logit-linear predictor for detecting mange given presence
g_det <- -0.5 + 0.25 * x3 + 0.7 * x4

# The probability of detecting mange in an image given presence
g_prob <- plogis(
  g_det
)

# A binary vector that determines which sites we have observed
#  the species. We use this and the 'w' vector to simulate
#  our observed data. Basically we need a photo at the site
#  and mangy coyote must also be at the site.
y_observed <- as.numeric(
  y>0
)

# simulate the observed mange data
g <- rbinom(
  n_photos,
  1,
  g_prob * as.numeric(
    y_observed[site_idx] * w[site_idx]
  )
)
```

Fitting the model to simulated data The JAGS model we are using is `"./jags_script/conditional_model_single_seaso`
The code for this is

```
model{
  # priors
  for(pc in 1:ncov_psi){
    psi[pc] ~ dlogis(0, 1)
  }
  for(rc in 1:ncov_rho){
    rho[rc] ~ dlogis(0,1)
  }
  for(oc in 1:ncov_omega){
```

4

```
    omega[oc] ~ dlogis(0,1)
  }
  for(gc in 1:ncov_gamma){
    gamma[gc] ~ dlogis(0,1)
  }
  for(site in 1:nsite){
    logit(psi_mu[site]) <- inprod(psi, psi_cov[site,])
    z[site] ~ dbern(psi_mu[site])
  }
  for(site in 1:nsite){
    logit(rho_mu[site]) <- inprod(rho, rho_cov[site,])
    y[site] ~ dbin(rho_mu[site] * z[site], J[site])
  }
  for(site in 1:nsite){
    logit(ome_mu[site]) <- inprod(omega, omega_cov[site,])
    x[site] ~ dbern(ome_mu[site] * z[site])
  }
  for(photo in 1:nphoto){
    logit(gam_mu[photo]) <- inprod(gamma, gamma_cov[photo,])
    q[photo] ~ dbern(gam_mu[photo] * x[site_vec[photo]])
  }
  # derived quantities

    n_coyote <- sum(z)
    n_mange <-   sum(x)


}
```

And here is how we fit the simulated data to the `JAGS` model using the `run.jags` packages.

```
# This is the initial values for whether mange is at a site. It
#  equals 1 if we observed it and is NA otherwise.
x_guess <- rep(
  NA,
  nsite
)
x_guess[unique(site_idx[g == 1])] <- 1

# The number of parameters we are estimating
ncov_psi <- 2
ncov_rho <- 2
ncov_omega <- 2
ncov_gamma <- 3

# put together the data list that we need for this analysis
data_list <- list(
  y = y,
  q = g,
  psi_cov = cbind(1, x1),
  rho_cov = cbind(1, x2),
  omega_cov = cbind(1, x1),
  gamma_cov = cbind(1, x3, x4),
  nphoto = n_photos,
  nsite = nsite,
```

```r
    site_vec = site_idx,
    ncov_psi = ncov_psi,
    ncov_rho = ncov_rho,
    ncov_omega = ncov_omega,
    ncov_gamma = ncov_gamma,
    J = rep(j,nsite)
)


# generate initial values for the analysis
inits_simulated <- function(chain){
  gen_list <- function(chain = chain){
    list(
      z = as.numeric(y_observed),
      x = x_guess,
      psi = rnorm(ncov_psi),
      rho = rnorm(ncov_rho),
      omega = rnorm(ncov_omega),
      gamma = rnorm(ncov_gamma),
      .RNG.name = switch(chain,
                         "1" = "base::Wichmann-Hill",
                         "2" = "base::Marsaglia-Multicarry",
                         "3" = "base::Super-Duper",
                         "4" = "base::Mersenne-Twister",
                         "5" = "base::Wichmann-Hill",
                         "6" = "base::Marsaglia-Multicarry",
                         "7" = "base::Super-Duper",
                         "8" = "base::Mersenne-Twister"),
      .RNG.seed = sample(1:1e+06, 1)
    )
  }
  return(switch(chain,
                "1" = gen_list(chain),
                "2" = gen_list(chain),
                "3" = gen_list(chain),
                "4" = gen_list(chain),
                "5" = gen_list(chain),
                "6" = gen_list(chain),
                "7" = gen_list(chain),
                "8" = gen_list(chain)
  )
  )
}


# fit the conditional model
m1 <- run.jags(
  "./jags_script/conditional_model_single_season.R",
  monitor = c("psi", "omega","rho","gamma", "n_coyote", "n_mange"),
  data = data_list,
  n.chains = 4,
  inits = inits_simulated,
  burnin = 10000,
  adapt = 10000,
```

```
  sample = 20000,
  modules = 'glm',
  method = 'parallel'
)
```

And here are the posterior estimates compared to the simulated parameters. Blue dots are median estimates, blue horizontal lines are 95% CI, and black dots are the true simulated values.