Chapter 1

State-space Covariates

Underlying all spatial capture recapture models is a point process model describing the distribution of individual activity centers ($\mathbf{s}_i$) within the state space ($\mathcal{S}$). So far we have focused our discussion on the homogeneous binomial point process, $\mathbf{s}_i \sim Uniform(\mathcal{S}), i = 1, 2, \ldots, N$, where N is the size of the population. This is a model of "spatial-randomness"[1] because the intensity of the activity centers is constant across the study area and the activity centers are distributed independently of each other.

The spatial-randomness assumption is often viewed as restrictive because ecological processes such as territoriality and habitat selection can result in non-random distributions of organisms. We have argued, however, that this assumption is less restrictive than may be recognized because the homogeneous point process actually allows for infinite possible configurations of activity centers. Furthermore, given enough data, the uniform prior will have very little influence on the estimated locations of activity centers. Nonetheless, the homogeneous point process model does not allow one to model population density using covariates—a central objective of much ecological research. For example, a homogeneous point process model may result in a density surface map indicating that individuals were more abundant in one habitat than another, but it does not do so explicitly. A more direct approach would be to model density using covariates as is done in generalized linear models (GLMs).

In this chapter we will present a method for fitting inhomogeneous binomial point process models using covariates in much the same way as is done with GLMs. The covariates we consider differ from those covered in previous chapters, which were typically attributes of the animal (*e.g.* sex, age) and were used to model movement or encounter rate. In contrast, here we wish to model covariates that are defined for all points in the the state-space, which we will refer to as state-space, or density, covariates. These may include continuous covariates such as elevation, or discrete covariates such as habitat type.

[1] The phrase "complete spatial-randomness" is reserved for the homogeneous Poisson point process

Borchers and Efford (2008) were the first to propose an inhomogeneous point process model for SCR models, and our approach is similar to theirs with the exception that we will use a binomial rather than a Poisson model because the binomial model is easily integrated into our data augmentation scheme and is consistent with the objective of determining how a *fixed* number of activity centers are distributed with respect to covariates.

The method we use to accommodate inhomogeneous binomial point process models within our MCMC algorithm is simple—we replace the uniform prior with a prior describing the distribution of the N activity centers conditional on the covariates. Development of this prior, which does not have a standard form, is a central component of this chapter.

1.1 Homogeneous point process revisited

The homogeneous Poisson point process is *the* model of "complete spatial randomness" and it is often used in ecology as a null model to test for departures from randomness. Given its central role in the analysis of point procesess, it is helpful to compare it with the binomial model that we use in our SCR models. The primary descriptor of the homogeneous point process model is the "intensity" parameter, μ which describles the expected number of points in an infinitesimally small area. Thus the intensity parameter can also be used to determine the expected number of points in any region of the state-space $\mathcal{S}$. To denote this, we say that the expected number of points in region $B \in \mathcal{S}$ is $n(B) = A(B)\mu$ where $A(B)$ is the area of region B. One property of the Poisson model is that if we divide the entire state-space into $k = 1, \ldots, K$ disjunct regions, the counts $\{n(B_k)\}$ are independent and identically distributed, (*i.i.d.*). This is one of the distinctions between the Poisson model and the binomial model, for which the counts $n(B_k)$ are not *i.i.d.* as we will explain shortly. This difference is also related to more important distinction between the two models, namely that the binomial model conditions on the number of points to be simulated N; whereas under the Poisson model N is random. Here is some simple R code to illustrate this point.

```
mu <- 4                             # intensity
Np <- rpois(1, mu)                  # Np is random
PPP <- cbind(runif(Np), runif(Np)) # Poisson point process

Nb <- 4
BPP <- cbind(runif(Nb), runif(Nb)) # Binomial point process
```

Note that in both models, the N points are independent of one another and distributed uniformly throughout $\mathcal{S}$. Thus, the intensity at any point $x \in \mathcal{S}$ is $\mu = 1/A(\mathcal{S})$ where $A(\mathcal{S})$ denotes the area of the state-space. For example, if the area of our state-space is 4 km^2, under a homogeneous model, the intensity is $\mu = 1/4$.

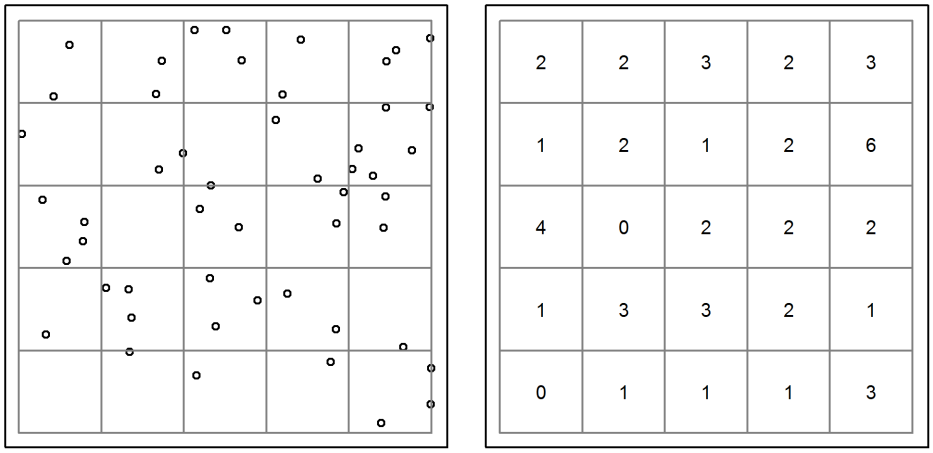

Figure 1.1: Homogeneous binomial point process with N=50 points represented in continuous and discrete space.

Although the Poisson model is typically described in terms of μ, the binomial model is not; rather, it is more common to consider a discrete state space, such as a grid with with K pixels. Under the binomial model, the number of points in each region is $n(B_k) \sim Bin(N, p_k)$ where $p_k = A(B)/A(\mathcal{S})$, ie p_k is simply the fraction of the state-space area in B_k. This discrete space representation of the binomial point process is shown in Fig. 1.1. The state-space in this case is the unit square, and thus probability of a point falling in each of the 25 disjunct regions is $p_k = 1/25$ and thus the expected counts are simply $\mathbb{E}(n(B_k)) = Np_k$. In the figure $N = 50$ and thus we would expect 2 points per pixel, which happens to be true in this case. Note also that these counts are not independent realizations from a binomial distribution since $\sum_k n(B_k) = N$. Instead, the model for the entire vector is $\mathbf{n}(\mathbf{B}) \sim Multinomial(N, \pi = (p_1, p_2, \ldots, p_K))$ (Illian, 2008). The dependence among counts has virtually no practical consequence when the number of pixels is large. For example, if we have 100 pixels, the number of counts in one pixels tells you very little about the expected count in another pixel. However, if there are only 2 pixels, then clearly the number of points in one pixel tells you exactly how many will occur in the remaining pixel. To gain familiarity with the multinomial distribution and the discrete representation of space, use the `rmultinom` function in R to simulate counts similar to those shown in Fig. 1.1, for example using a command such as:

```
n.B_k <- rmultinom(1, size=50, probs=rep(1/25, 25))
matrix(n.B_k, 5, 5)
```

The discrete space representation of the binomial point process is of practical importance when fitting SCR models because spatial covariates are almost always represented in a discrete format, often called "rasters" in GIS-speak. In such cases, we often need to change our definition of the prior for an activity center from $s_i \sim Uniform(\mathcal{S})$ to $s_i \sim Multinomial(1, \pi)$. In the latter case, the activity center is simply defined as an integer representing pixel "id". Note also that the multinomial distribution with an index of 1 (*i.e.* `size=1` in `rmultinom`) is referred to as the categorical distribution, which we will often make use of in the `BUGS` language.

1.2 Inhomogeneous binomial point process

As with the homogeneous model, the inhomogeneous binomial point process model is developed conditional on N. The primary distinction is that the uniform distribution is replaced with another distribution allowing for the intensity parameter to vary spatially. To arrive at this new distribution, define $\mu(x, \alpha)$ to be a function of spatially-referenced covariates (α) available at all points of the state space. Subsequently we will drop the vector of cofficients from our notation to be concise. Since an intensity must be strictly positive, it is natural to model $\mu(x)$ using the log-link.

$$\log(\mu(x)) = \sum_{j=1}^{J} \alpha_j v_j(x), \quad x \in \mathcal{S}$$

where α_j is the regression coefficient for covariate $v_j(x)$. To be clear, $v(x)$ is the value of any covariate, such as habitat type or elevation, at location x. This equation should look familiar because it is the standard linear model used in log-linear GLMs. Note, however, that we have no need for an intercept because it would be confounded with N. This should be intuitive since an intercept would represent the expected value of N when $\alpha = 0$, but we already have a parameter in the model for expected abundance, namely $\mathbb{E}[N] = \psi M$. Thus an intercept would be redundant, and without it we are still able to achieve our goal of describing the distribution of N activity centers as a function of spatial covariates.

Now that we have a model of the intensity parameter $\mu(x)$, we need to develop the associated probability density function to use in the place of the uniform prior used in the homogeneous model. Remembering that the integral of a pdf must be unity, we can create a pdf by dividing $\mu(x)$ by a normalizing constant, which in this case is the integral of $\mu(x)$ evlauated over the entire state-space. The probability density function is therefore

$$f(x) = \frac{\mu(x)}{\int_{x \in \mathcal{S}} \mu(x)\, \mathrm{d}x} \tag{1.1}$$

Substituting this distribution for the uniform prior allows us to fit inhomogeneous binomial point process models to spatial capture-recapture data. We can

also use this distribution to obtain the expected number of individuals in any given region. Specifically, the proprotion of N expected to occur in any region B when heterogeneity in density is present is $p(B) = \int_B f(x)\,\mathrm{d}x$. These are also the multinomial cell probabilities if the regions are disjoint and compose the entire state-space.

As a practical matter, note that the integral in the demoninator of $f(x)$ is evaluated over space, and since we almost always regard as space as two-dimensional, this is a two-dimensional integral that can be approximated using the methods discussed in refChXXX. These methods include Monte Carlo integration, Gaussian quadrature, etc... Alternatively, if our state-space covariates are in raster format, *i.e* they are in discrete space, the integral can be replaced with a sum over all pixels, which is much more efficient computationally.

We now have all the tools needed to fit inhomogeneous point process models. Before doing so, we note that this results in another point process model for the observation process, $\lambda(x)$. As a reminder, $\lambda(x)$ is the expected number of captures for a trap at point x. As was true for the homogeneous model, this intensity function is a convolution of the point process intensity ($\mu(x)$) and the encounter rate function, $\lambda(x) = \mu(x)g(x, s)$.

In the next section we walk through a few examples, building up from the simplest case where we actually observe the activity centers as though they were data. In the second example, we fit our new model to simulated data in which density is a function of a single continuous covariate. In the last example, we model the intensity of activity centers for a real dataset collected on jaguars (*Panthera onca*) in Argentina +citepsome paper by Augustin.

1.3 Examples

1.3.1 Simulation and analysis of inhomogeneous point processes

In SCR models, the point process is not directly observed, but in other contexts it is. Examples include the locations of disease outbreaks or the locations of trees in a forest. Fitting inhomogeneous point process models to such data is straight-forward and illustrates the fundamental process that we will later embed in our MCMC algorithm used to fit SCR models.

Suppose we knew the locations of 100 animals' activity centers. To estimate the intensity surface $\mu(x)$ underlying these points, we need to derive the likelihood for our data under this model. Given the pdf $f(x)$, and assuming that the points are mutually independent of one another, we may write the likelihood as the product of R such terms, where $R = 100$ is the sample size in this case, *ie* the observed number of activity centers.

$$\mathcal{L}(\alpha|\mathbf{x}_i) = \prod_{i=1}^{R} f(x_i)$$

Having defined the likelihood we could choose a prior and obtain the posterior for α using Bayesian methods, or we can find the maximum likelihood estimates (MLEs) using standard numerical methods as is demonstrated below.

First, let's simulate some data. Simulating data under an inhomogeneous point process model is often accomplished using indirect methods such as rejection sampling. Rejection sampling proceeds by simulating data from a standard distribution and then accepting or rejecting each sample using probabilities defined by the distribution of interest. For more information, readers should consult an accessible text like Robert and Casella (2004). In our example, we simulate from a uniform distribution and then accept or reject using the (scaled) probability density function $f(x)$. Note that we first define a spatial covariate (elevation) that is a simple function of the spatial coordinates increasing from the southwest to the northeast of our state-space.[2]

The following R commands demonstrate the use of rejection sampling to simulate an inhomogeneous point process for the covariate depicted in Fig. 1.3.1.

```
# spatial covariate (with mean 0)
elev.fn <- function(x) x[,1]+x[,2]-1

# 2-dimensional integration over unit square
int2d <- function(alpha, delta=0.02) {
  z <- seq(delta/2, 1-delta/2, delta)
  len <- length(z)
  cell.area <- delta*delta
  S <- cbind(rep(z, each=len), rep(z, times=len))
  sum(exp(alpha*elev.fn(S)) * cell.area)
  }

# Simulate PP using rejection sampling
set.seed(300225)
N <- 100
count <- 1
s <- matrix(NA, N, 2)
alpha <- 2 # parameter of interest
while(count <= 100) {
  x.c <- runif(1, 0, 1); y.c <- runif(1, 0, 1)
  s.cand <- cbind(x.c,y.c)
  elev.min <- elev.fn(cbind(0,0))
  elev.max <- elev.fn(cbind(1,1))
  pr <- exp(alpha*elev.fn(s.cand)) / int2d(alpha)
  Q <- max(c(exp(alpha*elev.min) / int2d(alpha),
             exp(alpha*elev.max) / int2d(alpha)))
  if(runif(1) < pr/Q) {
    s[count,] <- s.cand
    count <- count+1
    }
```

[2]Such functional forms of covariates are rarely available, which is why continuous spatial covariates are more often measured on a discrete grid.

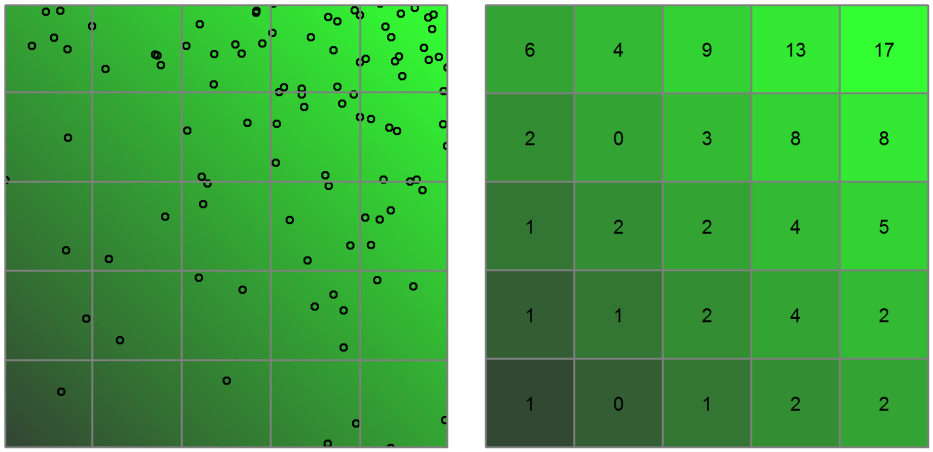

Figure 1.2: An example of a spatial covariate, say elevation, and a realization of a inhomogeneous binomial point process with N=100 and $\mu(x) = exp(\alpha Elev)$ where $\alpha = 2$.

```
  }
```

The simulated data are shown in Fig 1.3.1. High elevations are represented by light green and low elevations by dark green. The activity centers of one hundred animals are shown as points, and it is clear that these simulated animals prefer the high elevations. The underlying model describing this preference is $\log(\mu(x)) = exp(\alpha \times Elevation(x))$ where $\alpha = 2$ is the parameter to be estimated and $Elevation(x)$ is a function of the coordinates at x, as displayed on the map.

Given these points, we will now estimate α by minimizing the negative-log-likelihood using R's `optim` function.

```
# Negative log-likelihood
nll <- function(beta) {
  -sum(beta*elev.fn(s) - log(int2d(beta)))
  }
starting.value <- 0
fm <- optim(starting.value, nll, method="Brent",
            lower=-5, upper=5, hessian=TRUE)
c(Est=fm$par, SE=sqrt(1/fm$hessian)) # estimates and SEs
```

Maximizing the likelihood took a small fraction of a second, and we obtained an estimate of $\hat{\alpha} = 2.47$. We could plug in this estimate to our linear model at each point in the state-space to obtain the MLE for the intensity surface.

This example demonstrates that if we had the data we wish we had, *i.e.* if we knew the coordinates of the activity centers, we could easily estimate the parameters governing the underlying point process. Unfortunately, in SCR

models, the activity centers cannot be directly observed, but spatial re-captures, that is captures of individuals at multiple locations in space, provide us with the information needed to estimate these latent parameters. This example should also have illustrated the point that spatial covariates are rarely defined as continuous functions as we did here. Furthermore, even when they are, many multi-dimensional integration functions approximate the integral by pretending as though the covariate was defined on a discrete grid of small pixels.

1.3.2 Fitting inhomogeneous point process SCR model

One of the nice things about hierarchical models is that they allow us to break a problem up into a series of simple conditional relationships. Thus, we can simply add the methods described above into our existing MCMC algorithm to simulate the posteriors of α conditional on the simulated values of $\mathbf{s}_i$. To demonstrate, we will continue with the previous example. Specifically, we will overlay a grid of traps upon the map shown in Fig. 1.3.1. We will then simulate capture histories conditional upon the activity centers shown on the map. Then, we will attempt to estimate the activity center locations as though we did not know where they were.

Here is some `R` code to simulate the encounter histories under a Poisson observation model, which would be appropriate if animals could be detected multiple times at a trap during a single occassion.

```
# Create trap locations
xsp <- seq(-0.8, 0.8, by=0.2)
len <- length(xsp)
X <- cbind(rep(xsp, each=len), rep(xsp, times=len))

# Simulate capture histories, and augment the data
ntraps <- nrow(X)
T <- 5
y <- array(NA, c(N, ntraps, T))

nz <- 50 # augmentation
M <- nz+nrow(y)
yz <- array(0, c(M, ntraps, T))

sigma <- 0.1  # half-normal scale parameter
lam0 <- 0.5   # basal encounter rate
lam <- matrix(NA, N, ntraps)

set.seed(5588)
for(i in 1:N) {
    for(j in 1:ntraps) {
        distSq <- (s[i,1]-X[j,1])^2 + (s[i,2] - X[j,2])^2
        lam[i,j] <- exp(-distSq/(2*sigma^2)) * lam0
        y[i,j,] <- rpois(T, lam[i,j])
    }
```

Parameter	Mean	SD	q0.025	q0.5	q0.975
α					
λ_0					
σ					
N					
Density					

```
}
yz[1:nrow(y),,] <- y # Fill
```

Now that we have a simulated capture-recapture dataset y, and we have augmented it to create the new data object yz, we are ready to begin sampling from the posteriors. A commented Gibbs sampler written in `R` is available online. There are two small parts of the `R` code that distinguish it from previous code we have shown to fit homogeneous point processes. First, we need to update the parameter α conditional on all other parameters in the model. The code to do so is:

```
D1 <- int2d(beta1, delta=.05)
beta1.cand <- rnorm(1, beta1, tune[3])
D1.cand <- int2d(beta1.cand, delta=0.05)
ll.beta1 <- sum(  beta1*cov(S[,1],S[,2]) - log(D1) )
ll.beta1.cand <- sum( beta1.cand*(S[,1]+S[,2]) - log(D1.cand) )
if(runif(1) < exp(ll.beta1.cand - ll.beta1) )  {
    beta1<-beta1.cand
}
```

Next, we need to put the new prior on the activity centers:

```
#ln(prior), denominator is constant
prior.S <- beta1*cov(S[i,1], S[i,2]) # - log(D1)
prior.S.cand <- beta1*(Scand[1] + Scand[2]) # - log(D1)
if(runif(1)< exp((ll.S.cand+prior.S.cand) - (ll.S+prior.S))) {
    S[i,] <- Scand
    lam <- lam.cand
    D[i,] <- dtmp
    }
```

Applying this modified sampler to our data we obtain posterior distributions summarized in Table 1.3.2. Mixing is good, and as usual, life is very nice when we are working with simulated data.

Discrete space versions of these models can be fitted using `BUGS`. As discussed in Chapter 5, we can define s_i as the pixel id, and use the categorical distribution as a prior.

A good example of this is in +citeKery capricaillie. Here we present an analysis of the simulated data shown in the right panel of Fig. 1.3.1. The **BUGS** code to fit the model is shown in the following panel.

```
model{
sigma ~ dunif(0, 1)
lam0 ~ dunif(0, 5)
beta ~ dnorm(0,0.1)
psi ~ dbeta(1,1)

for(j in 1:nPix) {
  theta[j] <- exp(beta*elevation[j])
}

for(j in 1:nPix) {
  probs[j] <- theta[j]/sum(theta[])
}

for(i in 1:M) {
  w[i] ~ dbern(psi)
  s[i] ~ dcat(probs[])
  x0g[i] <- Sgrid[s[i],1]
  y0g[i] <- Sgrid[s[i],2]
  for(j in 1:ntraps) {
    dist[i,j] <- sqrt(pow(x0g[i]-grid[j,1],2) + pow(y0g[i]-grid[j,2],2))
    lambda[i,j] <- lam0*exp(-dist[i,j]*dist[i,j]/(2*sigma*sigma)) * w[i]
    y[i,j] ~ dpois(lambda[i,j])
    }
  }

N <- sum(w[])
D <- N/1 # unit square
}
```

A complete **R** script to conduct the analysis is presented in the online supplement.

1.3.3 The jaguar data

Estimating density of large felines was difficult before the advent of SCR. This is because you would never be able to conduct a distance sampling analysis for such rare and cryptic species, and because traditional capture-recapture methods don't yield estimates of density, only population size within some unknown region. This example not only demonstrates how readily density can be estimated for a globally imperilled species, but it also shows the importance of estimating density rather than just population size.

[describe study]

A few aspects of this design are noteworthy. First, the dimensions and

configuration of the trap array differed among the regions of the trap array. This fact alone could explain variation in the number of animals exposed to sampling, which would have no biological meaning. Furthermore, the area of inference is an irregular polygon that was not sampled uniformly. Only by estimating density can we hope to extrapolate our estiamates from the sampled region to get what we are after. In this case, this is readily accomplished since the entire state-space can be classified as one of the 3 levels of protection from poaching. Of course, it general it is always preferable to sample more uniformly throughout the area of interset in case some unmeasured covariate biases the extrapolation.

To assess the influence of poaching on jaguar density, we considered 2 metrics of poaching pressure, one political and one continuous measure of accessibility (Fig xxx).

1.4 MLE

Maybe its easy to adapt the MLE code from chapter 5 for doing a spatial covariate? For completeness it might be worth having that.

1.5 Other ideas

Should have some discussion on some ideas for building flexible models. Might be cool to use the Ickstadt/Wolpert as a model for the inhomogeneous point process. Dont have to do it, just mention it. Also some kind of a spline model or similar.

1.6 Summary

When state-space covariates are available, we can model density by replacing the uniform prior on the activity centers with a prior based on a log-linear function of covariates.

Bibliography

D.L. Borchers and MG Efford. Spatially explicit maximum likelihood methods for capture–recapture studies. *Biometrics*, 64(2):377–385, 2008.

J. Illian. *Statistical analysis and modelling of spatial point patterns.* Wiley-Interscience, 2008.

Christian P Robert and George Casella. *Monte Carlo Statistical Methods (second edition).* Springer-Verlag, New York, 2004.