

# 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,  $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$ ,  $i = 1, 2, \dots, N$ , where  $N$  is the size of the population. This is a model of “spatial-randomness”<sup>1</sup> 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  $\mathcal{S}$ , 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.

---

<sup>1</sup>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. First we will begin with a review of homogeneous point process models.

## 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 processes, 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,  $\mu$  which describes 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  $k = 1, \dots, K$  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 another 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<sup>2</sup>, under a homogeneous model, the intensity is  $\mu = 1/4$ .

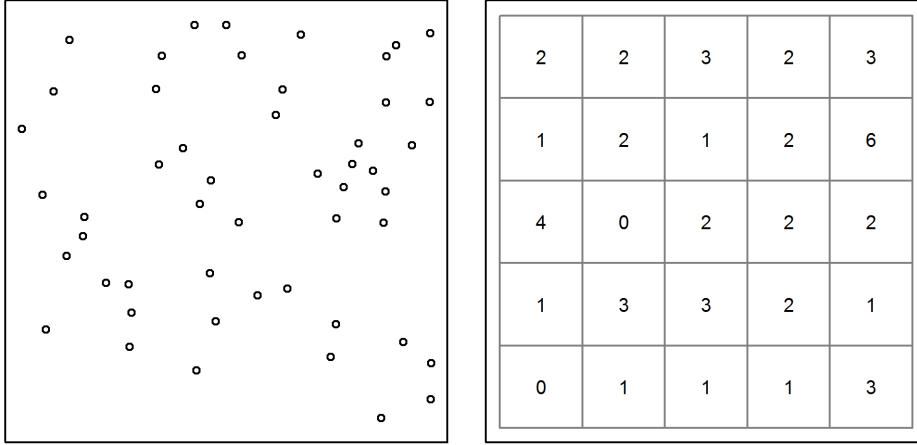


Figure 1.1: Homogeneous binomial point process with  $N=50$  points represented in continuous and discrete space.

Although the Poisson model is typically described in terms of  $\mu$ , the binomial model is not; rather, it is more common to consider a discrete state space, such as a grid with  $K$  pixels. Under the binomial model, the number of points in each region is  $n(B_k) \sim \text{Bin}(N, p_k)$  where  $p_k = A(B)/A(S)$ , 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 the 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 the empirical mean of the data in Fig. 1.1. 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  $\mathbf{n}(\mathbf{B}) \sim \text{Multinomial}(N, \pi = (p_1, p_2, \dots, p_K))$  (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)

```

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

## 105 1.2 Inhomogeneous binomial point process

106 As with the homogeneous model, the inhomogeneous binomial point process  
 107 model is developed conditional on  $N$ . The primary distinction is that the uni-  
 108 form distribution is replaced with another distribution allowing for the intensity  
 109 parameter to vary spatially. To arrive at this new distribution, define  $\mu(x, \beta)$  to  
 110 be a function of spatially-referenced covariates ( $\beta$ ) available at all points of the  
 111 state space. To be concise we will subsequently drop the vector of coefficients  
 112 from our notation, and simply use  $\mu(x)$ . Since an intensity must be strictly  
 113 positive, it is natural to model  $\mu(x)$  using the log-link.

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

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

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

$$f(x) = \frac{\mu(x)}{\int_{x \in \mathcal{S}} \mu(x) \, dx} \quad (1.1)$$

130 Substituting this distribution for the uniform prior allows us to fit inhomoge-  
 131 neous 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 proportion of  $N$  expected to occur in any region  $B$  when heterogeneity in density is present is  $p(B) = \int_B f(x) dx$ . 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 denominator of  $f(x)$  is evaluated over space, and since we almost always regard 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 (IPP) models. Before doing so, we note that the IPP for the activity centers results in another IPP 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 product of the point process intensity and the encounter rate function,  $\lambda(x) = \mu(x)g(x)$ .

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. Example three shows an analysis in discrete space using both **secr** (Efford, 2011) and **JAGS** (Plummer, 2003). In the last example, we model the intensity of activity centers for a real dataset collected on jaguars (*Panthera onca*) in Argentina.

## 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, *i.e* the observed number of activity centers.

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

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

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

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

```

186 # spatial covariate (with mean 0)
187 elev.fn <- function(x) x[1]+x[2]-1
188 # intensity function
189 mu <- function(x, beta) exp(beta*elev.fn(x=x))
190
191 # Simulate PP using rejection sampling
192 set.seed(300225)
193 N <- 100
194 count <- 1
195 s <- matrix(NA, N, 2)
196 beta <- 2 # parameter of interest
197 int.mu <- cuhre(2, 1, mu, beta=beta)$value
198 elev.min <- elev.fn(c(0,0)) #elev.fn(cbind(0,0))
199 elev.max <- elev.fn(c(1,1)) #elev.fn(cbind(1,1))
200 Q <- max(c(exp(beta*elev.min) / int.mu, #2d(beta),
201           exp(beta*elev.max) / int.mu)) #2d(beta))
202 while(count <= 100) {
203   x.c <- runif(1, 0, 1); y.c <- runif(1, 0, 1)
204   s.cand <- c(x.c,y.c)
205   pr <- exp(beta*elev.fn(s.cand)) / int.mu #2d(beta)
206   if(runif(1) < pr/Q) {
207     s[count,] <- s.cand
208     count <- count+1
209   }
210 }

```

211 The simulated data are shown in Fig 1.3.1. High elevations are represented  
 212 by light green and low elevations by dark green. The activity centers of one  
 213 hundred animals are shown as points, and it is clear that these simulated animals  
 214 prefer the high elevations. The underlying model describing this preference is

---

<sup>2</sup>Such functional forms of covariates are rarely available, which is why continuous spatial covariates are more often measured on a discrete grid.

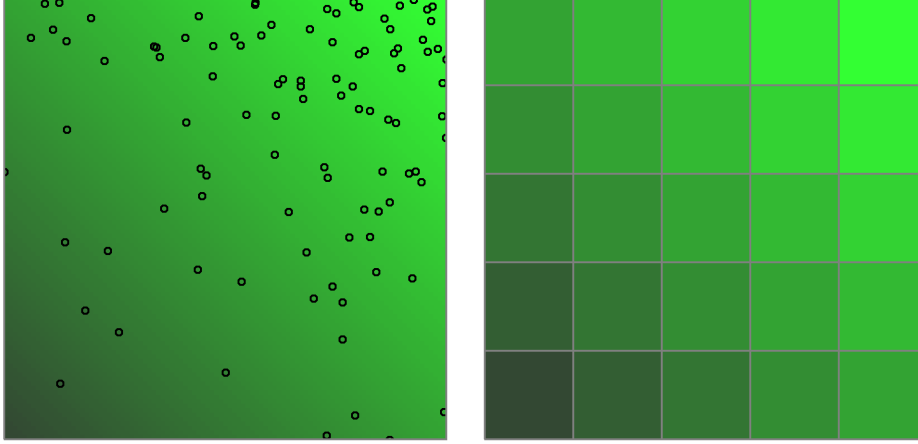


Figure 1.2: An example of a spatial covariate, say elevation, and a realization of a inhomogeneous binomial point process with  $N=100$  and  $\mu(x) = \exp(\beta \text{Elev})$  where  $\beta = 2$ .

215  $\log(\mu(x)) = \exp(\beta \times \text{Elevation}(x))$  where  $\beta = 2$  is the parameter to be estimated  
 216 and  $\text{Elevation}(x)$  is a function of the coordinates at  $x$ , as displayed on the map.  
 217 Given these points, we will now estimate  $\beta$  by minimizing the negative-log-  
 218 likelihood using R's `optim` function.

```
219 # Negative log-likelihood
220 nll <- function(beta) {
221   int.mu <- cuhre(2, 1, mu, beta=beta)$value
222   -sum(beta*elev.fn(s) - log(int.mu))
223 }
224 starting.value <- 0
225 fm <- optim(starting.value, nll, method="Brent",
226             lower=-5, upper=5, hessian=TRUE)
227 c(Est=fm$par, SE=sqrt(1/fm$hessian)) # estimates and SEs
```

228 Maximizing the likelihood took a small fraction of a second, and we obtained  
 229 an estimate of  $\hat{\beta} = 1.99$ . We could plug in this estimate to our linear model at  
 230 each point in the state-space to obtain the MLE for the intensity surface.

231 This example demonstrates that if we had the data we wish we had, *i.e.*  
 232 if we knew the coordinates of the activity centers, we could easily estimate  
 233 the parameters governing the underlying point process. Unfortunately, in SCR  
 234 models, the activity centers cannot be directly observed, but spatial re-captures,  
 235 that is captures of individuals at multiple locations in space, provide us with  
 236 the information needed to estimate these latent parameters.

### 237 1.3.2 Fitting inhomogeneous point process SCR models

#### 238 Continuous space

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

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

```

251 # Create trap locations
252 xsp <- seq(-0.8, 0.8, by=0.2)
253 len <- length(xsp)
254 X <- cbind(rep(xsp, each=len), rep(xsp, times=len))
255
256 # Simulate capture histories, and augment the data
257 ntraps <- nrow(X)
258 T <- 5
259 y <- array(NA, c(N, ntraps, T))
260
261 nz <- 50 # augmentation
262 M <- nz+nrow(y)
263 yz <- array(0, c(M, ntraps, T))
264
265 sigma <- 0.1 # half-normal scale parameter
266 lam0 <- 0.5 # basal encounter rate
267 lam <- matrix(NA, N, ntraps)
268
269 set.seed(5588)
270 for(i in 1:N) {
271   for(j in 1:ntraps) {
272     distSq <- (s[i,1]-X[j,1])^2 + (s[i,2] - X[j,2])^2
273     lam[i,j] <- exp(-distSq/(2*sigma^2)) * lam0
274     y[i,j,] <- rpois(T, lam[i,j])
275   }
276 }
277 yz[1:nrow(y),,] <- y # Fill

```

278 Now that we have a simulated capture-recapture dataset  $y$ , and we have  
 279 augmented it to create the new data object  $yz$ , we are ready to begin sampling  
 280 from the posteriors. A commented Gibbs sampler written in **R** is available  
 281 in the accompanying **R** package **scrbook** (see ?scrIPP). 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  $\beta$  conditional on all other parameters in the model. The code to do so is:

```

285 D1 <- cuhre(2, 1, mu, lower=c(xlims[1], ylims[1]),
286           upper=c(xlims[2], ylims[2]), beta=beta1)$value
287 beta1.cand <- rnorm(1, beta1, tune[3])
288 D1.cand <- cuhre(2, 1, mu, lower=c(xlims[1], ylims[1]),
289           upper=c(xlims[2], ylims[2]), beta=beta1.cand)$value
290 ll.beta1 <- sum( beta1*elev.fn.v(S) - log(D1) )
291 ll.beta1.cand <- sum( beta1.cand*elev.fn.v(S) - log(D1.cand) )
292 if(runif(1) < exp(ll.beta1.cand - ll.beta1) ) {
293   beta1<-beta1.cand
294 }
```

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

```

296 #ln(prior), denominator is constant
297 prior.S <- beta1*cov(S[i,1], S[i,2]) # - log(D1)
298 prior.S.cand <- beta1*(Scand[1] + Scand[2]) # - log(D1)
299 if(runif(1)< exp((ll.S.cand+prior.S.cand) - (ll.S+prior.S))) {
300   S[i,] <- Scand
301   lam <- lam.cand
302   D[i,] <- dtmp
303 }
```

We can apply this modified sampler to our data using the code shown in the following panel. We obtain posterior distributions summarized in Table 1.2. Mixing is good, and as usual, life is very nice when we are working with simulated data.

```

308 library(scrbook)
309 fm1 <- scrIPP(yz, X, M, 3000, xlims=c(0,0), ylims=c(-1,1),
310           tune=c(0.002, 0.1, 0.25, 0.07) )
311 plot(mcmc(fm1$out))
```

Table 1.1: Posterior summaries from inhomogeneous point proces model

	Mean	SD	2.5%	50%	97.5%
$\sigma = 0.10$	0.1026	0.0048	0.0935	0.1025	0.1123
$\lambda_0 = 0.50$	0.4419	0.0493	0.3496	0.4400	0.5390
$\psi = 0.66$	0.6826	0.0554	0.5762	0.6820	0.7923
$\beta = 2.00$	2.1601	0.3390	1.5193	2.1583	2.8043
$N = 100$	102.7696	6.2689	92.0000	102.0000	117.0000

Fitting continuous space IPP models is somewhat difficult in **BUGS** because our prior  $f(x)$  is not one of the available distributions<sup>3</sup> **secr** allows users to fit

<sup>3</sup>It is possible, if somewhat cumbersome, to add new distributions in **BUGS**.

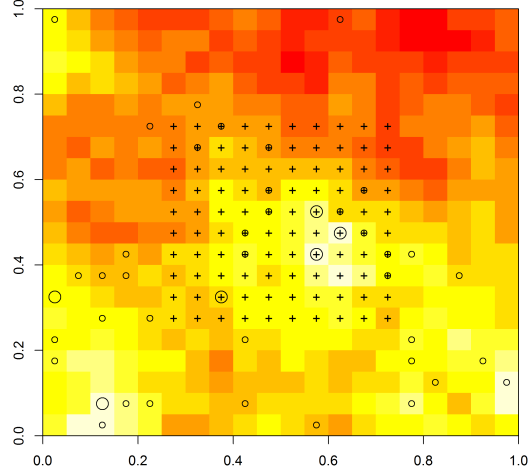


Figure 1.3: Simulated activity centers in discrete space. The spatial covariate, elevation, is highest in the higher areas. Density of activity centers (circles) increases with elevation. Trap locations are shown as crosses.

continuous space using polynomials of the x- and y- coordinates, but not for  
truly continuous covariates. However, these are not really important limitations  
because discrete space versions are straight-forward, and virtually all spatial  
covariates are defined as such.

### Discrete space

To fit the discrete space models, we follow the same steps as outlined in Chapter  
5—we define  $s_i$  as pixel ID, and we use the categorical distribution as a prior. A  
good example of this is in `+citeKery capricaille`. Here we present an analysis of  
the simulated data shown in the right panel of Fig. 1.3.1. The spatial covariate,  
let's call it elevation again, was simulated from a kriging type of model as  
shown on the help page `ch9simData` in `scrbook`. The points are the number  
of activity centers in each pixel, generated from a single realization of the IPP  
 $\mu(x) = 2elev$ .

The **BUGS** code to fit an IPP model to these data is shown in the following  
panel.

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

```

334
335 for(j in 1:nPix) {
336   theta[j] <- exp(beta*elevation[j])
337   probs[j] <- theta[j]/sum(theta[])
338 }
339
340 for(i in 1:M) {
341   w[i] ~ dbern(psi)
342   s[i] ~ dcat(probs[])
343   x0g[i] <- Sgrid[s[i],1]
344   y0g[i] <- Sgrid[s[i],2]
345   for(j in 1:ntraps) {
346     dist[i,j] <- sqrt(pow(x0g[i]-grid[j,1],2) + pow(y0g[i]-grid[j,2],2))
347     lambda[i,j] <- lam0*exp(-dist[i,j]*dist[i,j]/(2*sigma*sigma)) * w[i]
348     y[i,j] ~ dpois(lambda[i,j])
349   }
350 }
351
352 N <- sum(w[])
353 Density <- N/1 # unit square
354 }

```

355 This model can also be fit in **secr**, which refers to the pixel locations as  
356 a “mask”. **R** code to fit the models using **secr** and **JAGS** is available in  
357 **scrbook**, see `help(ch9secrYjags)`. Results of the comparison are shown in  
358 Table ?? and are very similar as expected.

359 Density surface maps can be created for fun, and of course to inform man-  
360 agement decisions. [describe how to do this]

### 361 1.3.3 The jaguar data

362 Estimating density of large felines has been a priority for many conservation  
363 organizations, but no robust methodologies existed before the advent of SCR.  
364 Distance sampling is not feasible for such rare and cryptic species, and tradi-  
365 tional capture-recapture methods yield estimates that are highly sensitive to the

Table 1.2: Comparison of **secr** and **JAGS** results

Software	Par	Est.	SD	lower	upper
secr	$N$	49.2803	5.7535	41.0087	64.3879
	$\beta$	2.1772	0.5628	1.0741	3.2804
	$\lambda_0$	0.9203	0.0764	0.7824	1.0825
	$\sigma$	0.0990	0.0038	0.0918	0.1068
JAGS	$N$	48.2072	5.4053	39.0000	60.0000
	$\beta$	2.1026	0.5323	1.0889	3.1506
	$\lambda_0$	0.9328	0.0766	0.7898	1.0921
	$\sigma$	0.1004	0.0041	0.0929	0.1089

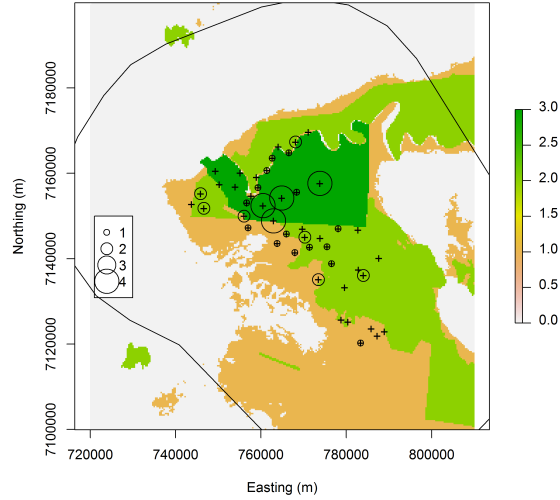


Figure 1.4: Jaguar detections

366 subjective choice of the effective survey area. In this example, we demonstrate  
 367 how readily density can be estimated for a globally imperilled species using  
 368 SCR. Furthermore, we show how inhomogeneous point process models can be  
 369 used to test important hypotheses regarding the factors affecting density.

370 [describe study]

371 A few aspects of this design are noteworthy. First, the dimensions and  
 372 configuration of the trap array differed among the regions of the trap array.  
 373 This fact alone could explain variation in the number of animals exposed to  
 374 sampling, which would have no biological meaning. Furthermore, the area of  
 375 inference is an irregular polygon that was not sampled uniformly. Only by  
 376 estimating density can we hope to extrapolate our estimates from the sampled  
 377 region to get what we are after. In this case, this is readily accomplished since  
 378 the entire state-space can be classified as one of the 3 levels of protection from  
 379 poaching. Of course, it general it is always preferable to sample more uniformly  
 380 throughout the area of interest in case some unmeasured covariate biases the  
 381 extrapolation.

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

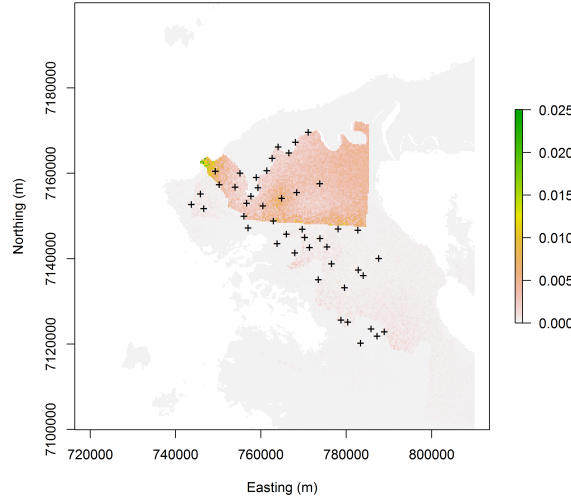


Figure 1.5: Estimated density surface for the jaguar dataset

## 1.4 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 normalized log-linear function of covariates. This yields a model of the inhomogeneous point process describing the location of activity centers, which can be used to test hypotheses about covariates affecting density. In rare cases, these covariates are truly continuous in the sense that they are defined as a function of space. More often, covariates are represented on rasters, which simplifies the analysis. Fitting these models can be accomplished using **BUGS**, **secr**, or the custom **R** code presented in this chapter and found in the package **scrbook**.

All the examples in this section included a single state-space covariate, but this was for simplicity only. Including multiple covariates poses no additional challenges. Likewise, additional model structure such sex-specific encounter rate parameters or behavioral responses can be accommodated.

## 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. Don't have to do it, just mention it. Also some kind of a spline model or similar.



# Bibliography

- Borchers, D. and Efford, M. (2008), “Spatially explicit maximum likelihood methods for capture–recapture studies,” *Biometrics*, 64, 377–385.
- Efford, M. (2011), “secr-spatially explicit capture–recapture in R,” .
- Illian, J. (2008), *Statistical analysis and modelling of spatial point patterns*, Wiley-Interscience.
- Plummer, M. (2003), “JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling,” in *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*. March, pp. 20–22.
- Robert, C. P. and Casella, G. (2004), *Monte Carlo Statistical Methods (second edition)*, New York: Springer-Verlag.