\documentclass{book}

\usepackage{amsmath}

\usepackage{amsfonts}

\usepackage{graphicx}

\usepackage{lineno}

\linenumbers

\begin{document}

\chapter{State-space Covariates}

Underlying all spatial capture recapture models is a point process

model describing the distribution of individual activity

centers (${\bf s}\_i$) within the state space ($\cal{S}$). So far we have focused our

discussion on the homogeneous binomial point process,

${\bf s}\_i \sim Uniform({\cal S}), i=1,2,\dots,N$, where $N$ is the

size of the population. This is often referred to as a model of

``complete spatial randomness'' (CSR) because the intensity of the

activity centers is constant across the study area and the activity

centers are distributed independently of each other.

The CSR assumption is often viewed as restrictive

because ecological processes such as

territoriality and habitat selection typically result in non-random

distributions of organisms. We have argued, however, that the CSR

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), using a

link function to connect the intensity parameter to the linear predictor.

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 using GLMs. The

covariates we consider differ

from those covered in previous chapters, which were typically

attributes of the animal ({\it 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 covariates, or abundance or density covariates. These may

include continuous covariates such as elevation, or discrete

covariates such as habitat type.

citet{borchersefford:2010} 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 {\it 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.

\section{Homogeneous point process revisited}

The homogeneous Poisson point process may be the most commonly-used model of

spatial randomness in ecology, thus it is helpful to compare it with

the binomial model that we will expand upon in this chapter. The

primary difference between the two models is that the binomial model

conditions on $N$, the number of points to be simulated; whereas under

the Poisson model $N$ is random. Here is some simple R code to

illustrate this difference.

\begin{verbatim}

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

\end{verbatim}

Note that in both models, the $N$ points are independent

of one another and distributed uniformly

throughout $\mathcal{S}$. Thus, the ``point process

intensity'' at any point $x \in \cal{S}$ is $\mu(x) = 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 $\mu(x) = 1/4$.

Although the Poisson model is typically described in terms of $\mu(x)$,

the binomial model is not; rather, it

is more common to consider a discrete state space, which we mention

here for clarity. Suppose that $\mathcal{S}$ is divided into $K$ non-overlapping

regions, the number of points in each region $B$ is $n(B) \sim Bin(N, p)$

where $\pi(B) = A(B)/A(\cal{S})$, ie $\pi(B)$ is simply the fraction of

the state-space area in $B$.

One additional property of the binomial model is that the $K$

realizations of $n$ are not independent since they must sum to

$N$. Instead, the model for the entire vector

is ${\bf n(B)} \sim Multinomial(N, {\mathbf{\pi}} = (p\_1, p\_2, \dots, p\_K))$.

[below is text from Chapter 4.3 – and it is great to have this material reiteratred here. I’m thinking you could copy the last 2 mini-paragraphs here to expand on the last point you made above – especially be sure to cite Illian et al. because that’s a good book. Also I ordered a copy for you]

The collection of individual activity centers s[1], …, s[N] represent a realization of a {\it binomial point process} (e.g., Illiana et al. 2008, p. XYZ). The binomial point process (BPP) is analogous to a Poisson point process in the sense that it represents a “random scatter” of points in space – except that the total number of points is {\it fixed}, whereas, in a Poisson point process it is random (having a Poisson distribution). It is natural to consider a binomial point process in the context of capture-recapture models because it preserves N in the model and thus preserves the linkage directly with closed population models. In fact, under the binomial point process model then Model M0 and other closed models are simple limiting cases of SCR models. In addition, use of the BPP model allows us to use data augmentation for Bayesian analysis of the models as in chapter 3, thus yielding a methodologically coherent approach to analyzing the different classes of models.

One consequence of having fixed N, in the BPP model, is that the BPP is not strictly a model of “complete spatial randomness”. This is because if you form counts in any set of regions say A1, …, Ak, even disjoint regions, let n(A1), .., n(Ak) be the counts then these counts are not independent. In fact, they have a multinomial distribution (see Illian et al. (2008. P XYZ )).

Thus, the BPP model introduces a slight bit of dependence among the point locations (and areal totals). However, in most situations this will have no practical effect on any inference or analysis and, as a practical matter, we will usually regard the BPP model as one of spatial independence among individual activity centers.

\section{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 $\mu(x,\mathbf{\alpha})$ to be a function of

spatially-referenced covariates ($\mathbf{\alpha}$) available at all regions 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 \cal{S}

\]

where $\alpha\_j$ 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 with the exception that we have no need

for an intercept because it would be confounded with

$N$. This is intuitive since an intercept would

represent the expected value of $N$ when $\alpha=0$, but we already

have a parameter in the model for $E[N]$, namely $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

\begin{equation}

f(x) = \frac{\mu(x)}{\int\_{\mathcal{S}} \mu(x)\, \mathrm{d}x}, \quad x \in

\mathcal{S}

\label{eq:pdf-ipp}

\end{equation}

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 $\pi(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 space as two-dimensional, this is a two-dimensional integral that can

be approximated using the methods discussed in Chapter \ref{chapter.xyz}. These methods include

Monte Carlo integration, Gaussian quadrature, etc... One

issue that often arises is that continuous spatial covariates are

\emph{not} represented as continuous, and instead are

defined on discrete grids, called ``rasters'' in GIS-speak. In such

cases, the integral in the denominator can be replaced with a sum over

all pixels citep(diggle:2003), which is much more efficient

computationally.

The inhomogeneous point process model for the activity centers results

in another point process model for the

observation process, which we have previously called $\lambda(x)$. 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 ($g(x,s)$),

$\lambda(x) = \mu(x) g(x,s)$.

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 tigers.

\section{Examples}

\subsection{Simulation and analysis of inhomogeneous point processes}

In SCR models, the point process is not directly observed, but in

other contexts the data in hand are the point locations

themselves. 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 that 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 $f(x)$, if we assume 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,

\emph{ie} the observed number of activity centers.

\[

\mathcal{L}({\bf \alpha} | {\bf x}\_i) = \prod\_{i=1}^R f(x\_i)

\]

Having defined the likelihood we may now obtain the posterior for

$\alpha$ using Bayesian methods, or we can find the maximum likelihood

estimates (MLEs) using standard numerical methods as demonstrated

below.

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 citet{robertcasella:2010}. 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. It should be obvious that such functional forms of

covariates are rarely available, which is why continuous spatial

covariates are more often measured on a discrete grid. Nonetheless, we

will proceed with our truly continuous covariate for illustrative

purposes. However, to evaluate the integral we end up discretizing the

state-space anyway.

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

%\newpage

\begin{small}

\begin{verbatim}

# spatial covariate

# Elevation as a function of the coordinates at point x

elev.fn <- function(x) x[,1]+x[,2]

# 2-dimensional integration over [-1, 1] square

int2d <- function(alpha, delta=0.02) {

z <- seq(-1+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(395)

N <- 100

count <- 1

s <- matrix(NA, N, 2) # matrix to hold simulated activity centers

alpha <- 2 # parameter of interest

Q <- max(c(exp(alpha\*elev.min) / int2d(alpha),

exp(alpha\*elev.max) / int2d(alpha))) # Rejection sampling bound

while(count <= 100) {

x.c <- runif(1, -1, 1); y.c <- runif(1, -1, 1) # proposed activity center

s.cand <- cbind(x.c,y.c)

elev.min <- elev.fn(cbind(-1,-1)); elev.max <- elev.fn(cbind(1,1))

pr <- exp(alpha\*elev.fn(s.cand)) / int2d(alpha)

if(runif(1) < pr/Q) {

s[count,] <- s.cand # accepted proposals

count <- count+1

}

}

\end{verbatim}

\end{small}

\begin{figure}

\centering

\includegraphics[width=7cm,height=7cm]{figs/elevMap}

\label{fig:elevMap}

\end{figure}

The simulated data are shown in Fig~\ref{fig:elevMap}. 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 $\alpha$ by minimizing the

negative-log-likelihood using \verb+R+'s \verb+optim+ function. Since,

we only have one parameter to estimate, we use method = ``Brent''.

\begin{small}

\begin{verbatim}

# Negative log-likelihood

nll <- function(beta) {

-sum(beta\*cov(S[,1], S[,2]) - 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

\end{verbatim}

\end{small}

Maximizing the likelihood took a small fraction of a second, and we

obtained an estimate of $\hat{\alpha}=2.01$. Not bad! 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, {\it 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. Rather they

are latent variables that we must either estimate or at least integrate out

of a likelihood. In SCR studies, spatial re-captures, that is captures of individuals at

multiple locations in space, provides us with the information needed to

estimate individual activity centers.

\subsection{Fitting the 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 $\alpha$ 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.~\ref{fig:elevMap}. 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.

\begin{small}

\begin{verbatim}

# 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])

}

}

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

\end{verbatim}

\end{small}

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. You will see that only two small parts of the R code

were changed. First, we need to update the parameter $\alpha$

conditional on all other parameters in the model. The code to do so is:

\begin{small}

\begin{verbatim}

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

}

\end{verbatim}

\end{small}

Next, we need to use $\alpha$ in the prior for the activity centers:

\begin{small}

\begin{verbatim}

#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

}

\end{verbatim}

\end{small}

Applying this modified sampler to our data we obtain posterior

distributions summarized in Table~\ref{tab:simIPP}. Mixing is good, and as usual,

life is very nice when we are working with simulated data.

\begin{table}

\centering

\begin{tabular}{lccccc}

Parameter & Mean & SD & q0.025 & q0.5 & q0.975 \\

\hline

$\alpha$ &&&&& \\

$\lambda\_0$ &&&&& \\

$\sigma$ &&&&& \\

$N$ &&&&& \\

Density &&&&& \\

\hline

\end{tabular}

\label{tab:simIPP}

\end{table}

It is worth noting that, although this method of fitting inhomogeneous

point process models does not require much modification of our custom MCMC

code, it is not so trivial to

implement these models in BUGS. The reason being

that the prior we use is not a standard distribution available by

default. It is, however, possible to use arbitrary distribution in

BUGS using the ??-trick.. Anyone remember how to do this? Here is an

example.

Actually what you do is this: Discretize the state-space (see Chapter 4.XYZ to be written by Andy) preferable at the same resolution as your raster and then define s[i] = integer from 1:Ngridpoints and

s[i] ~ dcat(probs[])

where

probs[k] = exp(alpha\*x)/[sum of all that stuff]

This works.

In fact Marc’s capricaillie paper has an example. In that case there were only 30 or so spatial units (forest tracts) and the covariate was “size of unit” so the model was putting activity centers in each pixel in proportion to area).

I think it is worth: (1) showing this example here possible or (2) give a sample WinBUGS analysis (simulated data) that shows this and discuss relevant issue: large rasters is really really expensive computationally. Might use JAGS/parallel to do big problems with thousands of raster points. I wonder if that’s possible?

In chapter 5 I’m adding information about the discrete state-space formulation of the model.

\subsection{The tiger data}

Hopefully Arjun can send me something. Something will turn up I’m sure. I also don’t object to using simulated data here and there if we have to. Maybe SECR has a canned example we could rip off?

\section{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.

\section{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.

\section{Summary}

When spatially-referenced 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.

\end{document}