12

MODELING LANDSCAPE CONNECTIVITY

Every spatial capture-recapture model that we have considered so far has expressed encounter probability as function of the Euclidean distance between individual activity centers $\mathbf{s}$ and trap locations $\mathbf{x}$. As a practical matter, models based on Euclidean distance imply circular, symmetric, and stationary home ranges of individuals, and these are not often biologically realistic. While these simple encounter probability models will often be sufficient for practical purposes, especially in small data sets, sometimes developing more complex models of the detection process as it relates to space usage of individuals will be useful. Animals may not judge distance in terms of Euclidean distance but, rather, according to the configuration of habitat patches, quality of local habitat, perceived mortality risk, and other considerations. Together, the degree to which these factors facilitate or impede movement determines landscape connectivity (Tischendorf and Fahrig, 2000), which is widely recognized to be an important component of population viability (With and Crist, 1995; COMPTON et al., 2007). Moreover, because encounter probability and the distance metric upon which it is based represent outcomes of individual movements about their home range, ecologists might have explicit hypotheses about how environmental variables affect the distance metric, and it is therefore desirable to incorporate these hypotheses directly into SCR models so that they may be formally evaluated statistically.

Although much theory has been developed to predict the effects of decreasing connectivity, few empirical studies have been conducted to test these predictions due to the paucity of formal methods for estimating connectivity parameters (Cushman et al., 2010). Instead, ecologists often rely on expert opinion or *ad hoc* methods of specifying connectivity values, even in important applied settings (Adriaensen et al., 2003; Beier et al., 2008; Zeller et al., 2012). In addition, no methods are available for simultaneously estimating population density and connectivity parameters, in spite of theory predicting interacting effects of density and connectivity on population viability (Tischendorf et al., 2005; Cushman et al., 2010). In this chapter, following Royle et al. (2012a), we provide a framework for modeling landscape connectivity using SCR models, by parameterizing models for encounter probability based on “ecological distance”. A natural candidate framework for modeling ecological distance is the least-cost path which is used widely in landscape ecology for modeling connectivity, movement and gene flow (Adriaensen et al., 2003; Manel et al., 2003; McRae et al., 2008). In practical applications, variables that influence landscape connectivity, or the effective cost of moving across the landscape, include things like highways (e.g., Epps et al., 2005), elevation (Cushman et al., 2006), ruggedness (Epps et al., 2007), snow cover (Schwartz et al., 2009), distance to escape terrain (Shirk et al., 2010), range limitations (McRae and Beier, 2007), or distance from urban areas, highways, human disturbance or other factors that animals might avoid.

Royle et al. (2012a) provided an SCR framework based on least-cost path for modeling landscape connectivity. They parameterized encounter probability based not on Euclidean distance but, rather, on the least-cost path between an individual's activity center and a trap location. This is parameterized in terms of one or more parameters that relate the *resistance* of the landscape to explicit covariates. In this way, SCR models can explicitly accommodate landscape structure and account for connectivity of the landscape. Using this methodological extension of SCR models, it is possible to make formal statistical inferences about movement and connectivity from capture-recapture studies that generate sparse individual encounter history data without subjective prescription of resistance or cost surfaces. While we believe there should be much ecological interest in developing SCR models that account for landscape connectivity, it is also important for obtaining more accurate estimates of density; under simple models of landscape connectivity, incorrectly fitting the basic model SCR0 produces substantial bias in estimates of N and hence density (Royle et al., 2012a).

12.1 SHORTCOMINGS OF EUCLIDEAN DISTANCE MODELS

In the standard SCR models, encounter probability is modeled as a function of Euclidean distance. For example, using the binomial observation model (Chapt. 5), let y_{ij} be individual- and trap-specific binomial counts with sample size K and probabilities p_{ij}. The Gaussian model is

$$p_{ij} = p_0 \exp(-d_{ij}^2/(2\sigma^2))$$

where $d_{ij} = ||\mathbf{x}_j - \mathbf{s}_i||$ is Euclidean distance. As usual, we will sometimes adopt the log-scale parameterization based on $\log(p_{ij}) = \alpha_0 + \alpha_1 d_{ij}^2$ where where $\alpha_0 = \log(p_0)$ and $\alpha_1 = -1/(2\sigma^2)$.

The main problem with the Euclidean distance metric in this encounter probability model is that it is unaffected by habitat or landscape structure, and it implies that the space used by individuals is stationary and symmetric, which may be unreasonable assumptions for some species. By stationary we mean in the formal sense of invariance to translation. That is, the properties of an individual home range centered at some point $\mathbf{s}$ are exactly the same as any other point say $\mathbf{s}'$. As an example, if the common detection model based on a bivariate normal probability distribution function is used, then the implied space usage by *all* individuals, no matter their location in space or local habitat conditions, is symmetric with circular contours of usage intensity.

In the framework of Royle et al. (2012a), SCR models explicitly incorporate information about the landscape so that a unit of distance is variable depending on identified covariates, say $C(\mathbf{x})$. Thus, where an individual lives on the landscape, and the state of the surrounding landscape, will determine the character of its usage of space. In particular, they suggest distance metrics, based on least-cost path, that imply irregular, asymmetric and non-stationary home ranges of individuals. As an example, Fig. 12.1 shows a typical symmetric home range (left panel), and a compressed home range (right panel) resulting from the effect of an environmental variable (center panel) on an animal's movement behavior. We might think of the environmental variable as representing an elevation gradient of a valley and so, for a species that avoids high elevation, space usage will be concentrated in flatter terrain at lower elevations and therefore producing the elliptical home range shape.

Figure 12.1. A symmetric home range (left), a habitat variable (center) such as representing an elevation gradient, and a non-symmetric home range (right) resulting from the cost imposed on movement by the habitat variable.

12.2 LEAST-COST PATH DISTANCE

We adopt a cost-weighted distance metric here which defines the effective distance between points by accumulating pixel-specific costs determined using a cost function defined by the user. The idea of cost-weighted distance to characterize animal use of landscapes is widely used in landscape ecology for modeling connectivity, movement and gene flow (Beier et al., 2008). For reasons of computational tractability we consider a discrete landscape defined by a raster of some prescribed resolution. The distance between any two points $\mathbf{x}$ and $\mathbf{x}'$ can be represented by a sequence of line segments connecting neighboring pixels, say $\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_m$. Then the cost-weighted distance between $\mathbf{x}$ and $\mathbf{x}'$ is

$$d(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{m-1} \text{cost}(\mathbf{l}_i, \mathbf{l}_{i+1}) ||\mathbf{l}_i - \mathbf{l}_{i+1}|| \tag{12.2.1}$$

where $\text{cost}(\mathbf{l}_i, \mathbf{l}_{i+1})$ is the user-defined cost to move from pixel $\mathbf{l}_i$ to neighboring pixel $\mathbf{l}_{i+1}$ in the sequence. Given the cost of each pixel, it is a simple matter to compute the cost-weighted distance between any two pixels, along *any* path, simply by accumulating the incremental costs weighted by distances. In the context of spatial capture-recapture models (and, more generally, landscape connectivity) we are concerned with the *minimum* cost-weighted distance, or the *least-cost path*, between any two points which we will denote by d_{lcp}, which is the sequence $\mathcal{P} = (\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_m)$ that minimizes the objective function defined by Eq. 12.2.1. That is,

$$d_{lcp}(\mathbf{x},\mathbf{x}') = \min_{\mathcal{P}} \sum_{i=1}^{m-1} \text{cost}(\mathbf{l}_i,\mathbf{l}_{i+1})||\mathbf{l}_i - \mathbf{l}_{i+1}|| \tag{12.2.2}$$

The least-cost path distance can be calculated in many geographic information systems and other software packages, including the **R** package `gdistance` (van Etten, 2011) which we use below.

The key ecological aspect of least-cost path modeling is the development of models for pixel-specific cost. A natural approach is to model cost as a function of one or more covariates defined on every pixel of the according raster. For example, using a single covariate $C(\mathbf{x})$ we define the cost of moving from some pixel $\mathbf{x}$ to neighboring pixel $\mathbf{x}'$ as

$$\log(cost(\mathbf{x},\mathbf{x}')) = \alpha_2 \left(\frac{C(\mathbf{x}) + C(\mathbf{x}')}{2} \right) \tag{12.2.3}$$

Thus, if $\alpha_2 = 0$ then substituting $\text{cost}(\mathbf{x},\mathbf{x}') = \exp(0) = 1$ into Eq. 12.2.2 will produce the ordinary Euclidean distance between points. Here we assume the covariate C is positive-valued, and we constrain $\alpha_2 \geq 0$ so as to avoid negative costs. While not necessarily problematic from a mathematical standpoint, negative costs are unrealistic biologically.

The use of least-cost path models to model landscape connectivity has been around for a long time. And, although α_2 is rarely known, conservation biologists design linkages that require this resistance value as input (see Beier et al., 2008, and articles cited therein). However, formal inference (e.g., estimation) of parameters is not often done. Instead, in many existing applications of least-cost path analysis, the parameter α_2 is fixed by the investigator, or based on expert opinion (Beier et al., 2008), although recently researchers have begun to define costs based on resource selection functions[1], animal movement (Tracy, 2006; Fortin et al., 2005), or genetic distance data (e.g., Gerlach and Musolf (2000); Epps et al. (2007); Schwartz et al. (2009).

To formalize the use of cost-weighted distance in SCR models, we substitute Eq. 12.2.2 in the expression for encounter probability (Eq. **??**) and maximize the resulting likelihood (see below). In doing so, we can directly estimate parameters of the least-cost path model, evaluate how landscape covariate influence connectivity, and test explicit hypotheses about these things using only individual level encounter history data from capture-recapture studies.

12.2.1 Example of Computing Cost-weighted distance

As an example of the cost-weighted distance calculation consider the following landscape comprised of 16 pixels with unit spacing identified as follows, along with the pixel-specific cost:

```
      pixel ID                Cost
     4 8 12 16            100   1    1  1
     3 7 11 15            100 100    1  1
```

[1] We address the integration of resource selection models based on telemetry data with SCR models in Chapt. 13.

```
   2 6 10 14           100 100 100   1
   1 5  9 13           100 100   1   1
```

We assume the scale is such that the distance between neighboring pixels in any cardinal direction is 1 unit, and the distance between neighbors on a diagonal is $\sqrt{2}$ units. We assigned low cost of 1 to "good habitat" pixels (or pixels we think of as "highly connected" by virtue of being in good habitat) and, conversely, we assign high cost (100) to "bad habitat". This simple cost raster is shown in Fig. 12.2. The **R** commands for creating this simple example are as follows (which can be run using the **R** script `SCRed` – see the help file for that):

```
> library(raster)
> library(gdistance)
> r<-raster(nrows=4,ncols=4)
> projection(r)<- "+proj=utm +zone=12 +datum=WGS84" # Sets the projection
> extent(r)<-c(.5,4.5,.5,4.5) #sets the extent of the raster
> costs1<- c(100,100,100,100,1,100,100,100,1,1,100,1,1,1,1,1)
> values(r)<-matrix(costs1,4,4,byrow=FALSE) #assign the costs to the raster
> par(mfrow=c(1,1))
> plot(r)
```

This produces Fig. 12.2.

For this simple case we can easily compute the shortest cost-weighted distance between any pixels "by eye". For example, the shortest cost-weighted distance between pixels 5 and 9 in this example is 50.5 units: $1 * (100 + 1)/2 = 50.5$, the shortest distance between pixels 4 and 8 is also 50.5, while the shortest cost-distance between 4 and 12 is 51.5. What is the shortest distance between 7 and 16? Suppose an individual at pixel 7 can move diagonal (which has distance $\sqrt{2}$) and pay $\sqrt{2}(100 + 1)/2$, and then move once to the right to pay 1 additional unit cost, for a total of 72.4. However, if the individual instead moved one unit to the right, to pixel 11, and then diagonally, the total cost is 51.914 which is the minimum cost-weighted distance in getting from pixel 7 to 16. These two ways of moving from 7 to 16 have the same Euclidean distance, but different cost-weighted distances according to our cost function.

The least-cost path distances can be computed with just a couple **R** commands, and these commands can be inserted directly into the likelihood construction for an ordinary spatial capture-recapture model. The **R** package `gdistance` calculates least-cost path using Dijkstra's algorithm (Dijkstra, 1959) (from the `igraph` package (Csardi and Nepusz, 2006)). To compute the least-cost path, or the minimum cost-weighted distances between every pixel and every other pixel, we make use of the helper function `transition`, which calculates the cost of moving between neighboring pixels. It operates on the inverse-scale ("conductance"), and so the `transitionFunction` argument is given as $1/mean(x)$. The function `geoCorrection` modifies this object depending on the projection of the coordinate system (e.g., it corrects for curvature of the earth's surface if longitude/latitude coordinates are used). The result is fed into the function `costDistance` to compute the pair-wise distance matrix. For that, we define the center points of each raster, here these are just integers on $[1, 4] \times [1, 4]$. The commands altogether are as follows:

```
> tr1<-transition(r,transitionFunction=function(x) 1/mean(x),directions=8)
```

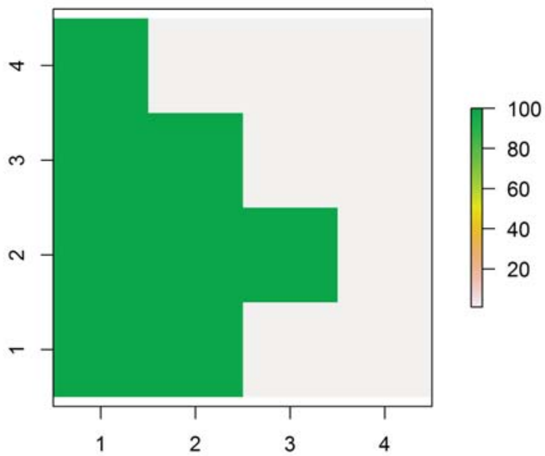

Figure 12.2. A 4×4 raster depicting a binary cost surface, with cost $= 1$ (white) or 100 (shaded) to represent ease of movement across a pixel.

```
> tr1CorrC<-geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
> pts<-cbind( sort(rep(1:4,4)),rep(4:1,4))
> costs1<-costDistance(tr1CorrC,pts)
> outD<-as.matrix(costs1)
```

Now we can look at the result and see if it makes sense to us. Here we produce the first 5 columns of this distance matrix to illustrate a couple of examples of calculating the minimum cost-weighted distance between points:

```
> outD[1:5,1:5]
         1        2        3        4        5
1   0.0000 100.0000 200.0000 205.2426 100.0000
2 100.0000   0.0000 100.0000 200.0000 141.4214
3 200.0000 100.0000   0.0000 100.0000 126.1604
4 205.2426 200.0000 100.0000   0.0000 105.2426
5 100.0000 141.4214 126.1604 105.2426   0.0000
```

An interesting case is that between point 1 and 4. Note that simply taking the shortest Euclidean distance, weighted by cost, produces a cost-weighted distance of 100×1 to move from pixel 1 to pixel 2, and similarly from 2 to 3 and 3 to 4, producing a total cost-weighted distance of 300. However, the actual *least-cost path* has cost-weighted distance 205.2426. See if you can figure out the shortest path by inspection.

The key point here is that, once we can compute this distance matrix, we can use it as the distance matrix in computing the encounter probability between acctivity centers and traps, and we can use our existing MLE technology (Chapt. 6) to fit models that are based on ecological distance.

12.3 SIMULATING SCR DATA USING ECOLOGICAL DISTANCE

Royle et al. (2012a) simulated capture-recapture data such that landscape connectivity was governed by a cost function having a single covariate, and they considered two hypothetical covariate landscapes (Fig. 12.3). The landscape here is a 20×20 pixel raster, with extent = $[0.5, 4.5] \times [0.5, 4.5]$. For example, think of each pixel as representing, say, a 1×1 km grid cell with something like "percent developed" or "trail/road density" representing the covariate. For sampling by capture-recapture, imagine that 16 camera traps are established at the integer coordinates $(1, 1), (1, 2), \ldots, (4, 4)$. The two covariates were constructed as follows (see `?make.EDcovariates` for the **R** commands): First is an increasing trend from the NW to the SE ("systematic covariate"), where $C(\mathbf{x})$ is defined as $C(\mathbf{x}) = row(\mathbf{x}) + col(\mathbf{x})$ and $row(\mathbf{x})$ and $col(\mathbf{x})$ are just the row and column, respectively, of the raster. This might mimic something related to distance from an urban area or a gradient in habitat quality due to land use, or environmental conditions such as temperature or precipitation gradients. In the second case we make up a covariate by generating a field of spatially correlated noise to emulate a typical patchy habitat covariate ("patchy covariate") such as tree or understory density.

For both covariates we use a cost function in which transitions from pixel $\mathbf{x}$ to $\mathbf{x}'$ is given by:

$$\log(\text{cost}(\mathbf{x}, \mathbf{x}')) = \alpha_2 \left(\frac{C(\mathbf{x}) + C(\mathbf{x}')}{2} \right)$$

where $\alpha_2 = 1$ for simulating the observed data. Remember that with $\alpha_2 = 0$ the model reduces to one in which the cost of moving across each pixel is constant, and therefore Euclidean distance is operative. In the left panel of Fig. 12.3, a sample realization of $N = 100$ activity centers is shown. While encounter probability is assumed to be related to landscape connectivity according to the single-variable cost function, individual activity centers are assumed to be uniformly distributed, although we can modify this assumption (See Sec. 12.8 below).

When distance is defined by the cost-weighted distance metric given by Eq. 12.2.2 then individual space-usage varies spatially in response to the landscape covariate(s) used in the distance metric. As a consequence, home range contours are no longer circular, as in SCR models based on Euclidean distance. For example, using one of the covariates we use in our simulation study below (Fig. 12.3, right panel) with a Gaussian encounter model but having distance metric defined by Eq. 12.2.2, produces home ranges such as those shown in Fig. 12.4.

To simulate data, we have to load the `scrbook` package and call the function `make.EDcovariates` to generate our raster covariates (see the help file for how that is done). We process the covariate into a least-cost path distance matrix, and then simulate observed encounter

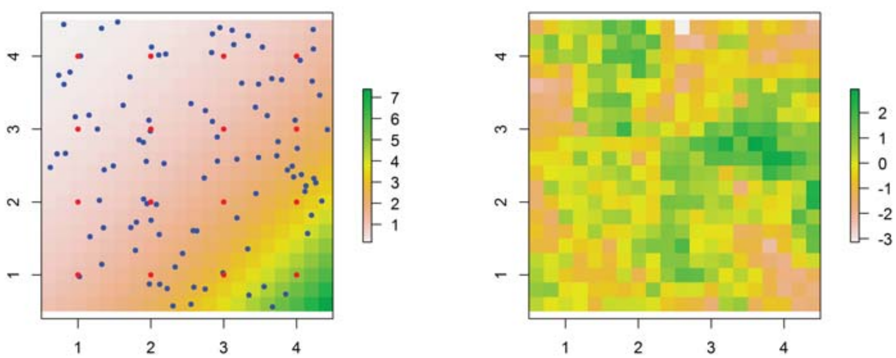

Figure 12.3. Two covariates (defined on a 20×20 grid) used in simulations. Left panel shows a covariate with systematic structure meant to mimic distance from some feature, and the right panel shows a "patchy" covariate. A hypothetical realization of $N = 100$ activity centers (blue dots) is superimposed on the left figure, along with 16 trap locations.

data using standard methods which we have used many times previously in this book. The complete set of **R** commands is:

```
### Grab a covariate
library(scrbook)
set.seed(2013)
out<-make.EDcovariates()
covariate<-out$covariate.patchy

### prescribe some settings
N<-200
alpha0<- -2
sigma<- .5
alpha1<- 1/(2*sigma*sigma)
alpha2<-1
K<- 5
S<-cbind(runif(N,.5,4.5),runif(N,.5,4.5))

# make up some trap locations
xg<-seq(1,4,1); yg<-4:1
traplocs<-cbind( sort(rep(xg,4)),rep(yg,4))
points(traplocs,pch=20,col="red")
ntraps<-nrow(traplocs)
```

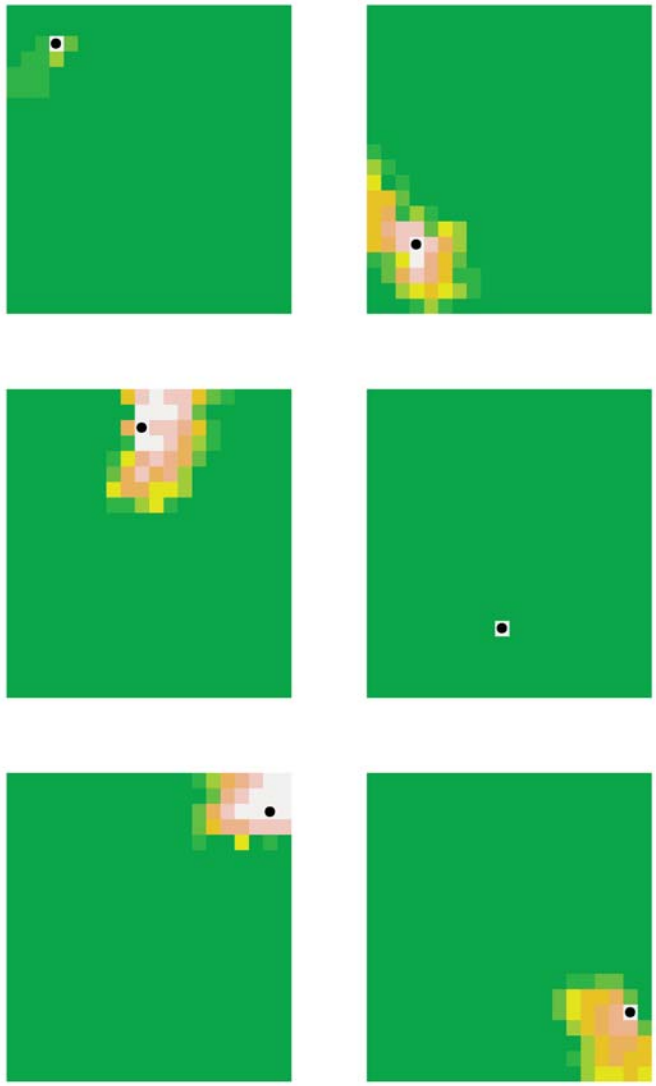

Figure 12.4. Typical home ranges for 6 individuals based on the cost surface shown in the right panel of Fig. 12.3 with $\alpha_2 = 1$. The black dot indicates the home range center and the pixels around each home range center are shaded according to the probability of encounter, if a trap were located in that pixel.

```
### make a raster and fill it up with the "cost"
r<-raster(nrows=20,ncols=20)
projection(r)<- "+proj=utm +zone=12 +datum=WGS84"
extent(r)<-c(.5,4.5,.5,4.5)
cost<- exp(alpha2*covariate)

### compute least-cost path distance
tr1<-transition(cost,transitionFunction=function(x) 1/mean(x),directions=8)
tr1CorrC<-geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
D<-costDistance(tr1CorrC,S,traplocs)
probcap<-plogis(alpha0)*exp(-alpha1*D*D)

# now generate the encounters of every individual in every trap
# discard uncaptured individuals
Y<-matrix(NA,nrow=N,ncol=ntraps)
for(i in 1:nrow(Y)){
 Y[i,]<-rbinom(ntraps,K,probcap[i,])
}
Y<-Y[apply(Y,1,sum)>0,]
```

12.4 LIKELIHOOD ANALYSIS OF ECOLOGICAL DISTANCE MODELS

Throughout much of this book we rely on Bayesian analysis by MCMC mostly using **BUGS**, but sometimes (as in Chapt. 17) developing our own implementations. However, occasionally we prefer to use likelihood estimation, such as when we can compare a set of models directly by likelihood either to do a direct hypothesis test of a parameter, or to tabulate a bunch of AIC values. For the class of models that use least-cost path, we also prefer likelihood methods not because they have any conceptual or methodological benefit, but simply because they are more computationally efficient to implement (Royle et al., 2012a).

There are no technical considerations in adapting our formulation of maximum likelihood estimation (Borchers and Efford, 2008) from Chapt. 6 for the class of models based on least-cost path (see the appendix in Royle et al. (2012a) for complete details). Likelihood analysis is really just a straightforward adaptation in which we replace the Euclidean distance with least-cost path. Consider the Bernoulli model in which the individual- and trap-specific observations have a binomial distribution conditional on the latent variable $\mathbf{s}_i$:

$$y_{ij}|\mathbf{s}_i \sim \text{Binomial}(K, p_{\boldsymbol{\alpha}}(d_{lcp}(\mathbf{x}_j, \mathbf{s}_i; \alpha_2); \alpha_0, \alpha_1) \quad (12.4.1)$$

where we have indicated the dependence of p on the parameters $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, \alpha_2)$, and also d_{lcp} which itself depends on α_2, and the latent variable $\mathbf{s}_i$. We note that the only difference between likelihood analysis of this model and the standard Bernoulli model, is the use of d_{lcp} here. For the random effect we have $\mathbf{s}_i \sim \text{Uniform}(\mathcal{S})$, we can easily compute the integrated (marginal) likelihood of an encounter history. The likelihood is given in the `scrbook` package as the function `intlik3ed`. The help file provides an example of its usage and for simulating data. To use this function the cost covariate $C(\mathbf{x})$ has to be of class

Table 12.1. Summary output of fitting models based on Euclidean and least-cost path distance to simulated data using the `intlik3ed` function (see `?intlik3ed`). Data were simulated based on the least-cost path model using the "patchy" covariate shown in Fig. 12.3.

Distance metric	-loglik	α_0	α_1	$\log(n_0)$	α_2
True value		-2	2	4.644	1
Euclidean	133.495	-1.885	1.247	3.549	–
Least-cost path (truth)	70.119	-1.780	2.471	4.459	0.046

`RasterLayer` which requires packages `sp` and `raster` to manipulate.

12.4.1 Example of SCR with least-cost path

Now we use the **R** function `nlm` along with our `intlik3ed` function to obtain the MLEs of the model parameters for the data simulated in Sec. 12.3. We'll do that for both the standard Euclidean distance and then for the ecological distance based on the "patchy" covariate using the following commands:

```
> frog1<-nlm(intlik3ed,c(alpha0,alpha1,3)),hessian=TRUE,y=Y,K=K,X=traplocs,
               distmet="euclid",covariate=covariate,alpha2=1)

> frog2<-nlm(intlik3ed,c(alpha0,alpha1,3,-.3),hessian=TRUE,y=Y,K=K,X=traplocs,
               distmet="ecol",covariate=covariate,alpha2=NA)
```

The summary output for the two model fits is shown in Table 12.1. The model based on least-cost path (the data generating model) appears to be much preferred in terms of negative log-likelihood. The output parameter order is $(\alpha_0, \alpha_1, \log(n_0), \text{and} \log(\alpha_2))$ (remember, we want to keep α_2 positive, so it's logarithm is estimated). The data generating parameter values were $\alpha_0 = -2$, $\alpha_1 = 2$ and $\log(\alpha_2) = 0$. The simulated sampling produced a sample of 96 individuals and so the number of individuals not captured is $n_0 = 104$, and $\log(n_0) = 4.64$. We see that the MLEs of the least-cost path model are pretty close whereas they are not so close under the misspecified model based on Euclidean distance.

12.5 BAYESIAN ANALYSIS

While implementation of these ecological distance SCR models is reasonably straightforward, the model cannot be fitted in the **BUGS** engines because least-cost path distance cannot be computed. It would be possible to fit the models in **BUGS** if the parameter α_2 was fixed. In that case, one could compute the distance matrix ahead of time and reference the required elements for a given $\mathbf{s}$. Alternatively, it would be possible to write a custom MCMC routine using the methods we present in Chapt. 17, although we have not yet developed our own MCMC implementation of SCR models with ecological distance metrics.

12.6 SIMULATION EVALUATION OF THE MLE

Royle et al. (2012a) carried-out a limited simulation study to evaluate the general statistical performance of the density estimator under this new model, the effect of mis-specifying the model with a normal Euclidean distance metric, and evaluate the general bias and precision properties of the MLE using the systematic and patchy landscapes shown in Fig. 12.3.

Their results showed extreme bias in estimates of N when the misspecified Euclidean distance is used, and only negligible small-sample bias of 3-5% in the MLE of N using the least-cost distance which becomes negligible as the expected seample size increases (either due to increasing K, or larger population sizes). The performance of estimating the other parameters, including the cost parameter α_2 mirrors the results for estimating N. We reproduce a subset of the results from Royle et al. (2012a) in Table 12.6.

Table 12.2. Simulation results for estimating population size N for a prescribed state-space with $N = 100$ or $N = 200$ and various levels of replication (K) using the "patchy" landscape shown in Fig. 12.3. For each simulated data set, the SCR model was fitted by maximum likelihood with standard Euclidean distance ("euclid"), or least-cost path ("lcp"), which was the true data-generating model. The summary statistics of the sampling distribution reported are the mean, standard deviation ("SD") and quantiles (0.025, 0.50, 0.975).

	N=100				
	mean	SD	0.025	0.50	0.975
$K = 3$					
euclid	78.68	18.12	49.40	76.34	125.47
lcp	110.96	28.65	69.55	106.98	181.84
$K = 5$					
euclid	77.85	11.55	59.17	77.44	101.14
lcp	104.44	15.79	78.38	101.47	139.55
$K = 10$					
euclid	78.01	5.26	68.00	77.96	87.81
lcp	100.42	7.56	86.72	100.34	115.47
	N=200				
$K = 3$					
euclid 154.34	33.74	107.00	146.34	221.43	
lcp 208.77	49.29	141.68	197.89	325.77	
$K = 5$					
euclid 153.39	15.57	129.31	149.54	185.38	
lcp 200.91	20.78	164.42	200.47	246.46	
$K = 10$					
euclid 156.27	8.51	142.17	156.05	174.55	
lcp 198.45	11.44	180.06	198.04	219.52	

12.7 DISTANCE IN AN IRREGULAR PATCH

We provide another illustration of how to employ ecological distance calculations in SCR models. This example is meant to mimic a situation where we have something like a hard habitat boundary such as a habitat corridor or park unit or some other block of relatively homogeneous good-quality habitat for some species. This particular system (shown in Fig. 12.5) could be habitat surrounded by a suburban wasteland of McDonuts and Beer-Marts, much less hospitable habitat for most species. For our purposes, we suppose that individuals live within the buffered "f-shaped" region, although we could also imagine the negative of the situation in which individuals live outside of the region, so that the polygon represents a barrier (a lake) or bad habitat (an urban area) or similar. We describe the steps for creating this landscape shortly, so that you can use a similar process to generate more relevant landscapes for your own problems.

In this case we're not going to estimate any parameters of the cost function (though you could adapt the analyses of the previous sections to do that) but instead we're going to use ecological distance ideas only to constrain movement within (or to avoid) landscape features. Note that, normally, distance "as the crow flies" would not be suitable for irregular habitat patches such as that shown in Fig. 12.5.

12.7.1 Basic Geographic Analysis in R

In practical applications our landscape will contain polygons which delineate good or bad habitat or other important characterisetics of the landscape. These might exist as GIS shapefiles or merely as a text file with coordinates defining polygon boundaries. To work with polygons in the context of SCR models we need to create a raster, overlay the polygon and assign values to each pixel depending on whether pixels are in the polygon or not, or how far they are from polygon boundaries. These operations are relatively easy to do within a GIS system but we need to be able to do them in **R** in order to compute the least-cost paths needed in the likelihood evaluation. Some additional geographic analyses have been discussed in Secs. **??** and 17.5 where we talked about reading in the shapefile and doing SCR calculations on that.

Often we will have GIS shapefiles that define polygons but, here, we create a set of polygons by buffering and joining some line segments. In the **R** package `scrbook`, we provide a function `make.seg` which allows you to make such lines segments given a specific trap region. To use `make.seg` we first create a plot region and then call `make.seg` which has a single argument being the number of points used to define the line segment. The user will click on the visual display until the required number of points has been obtained by `make.seg`. In the following set of commands we generate two line segments, `l1` consisting of 9 points and `l2` consisting of 5 points, and these reside in a geographic region enclosedd by $[0, 10] \times [0, 10]$:

```
library(scrbook)
library(sp)
plot(NULL,xlim=c(0,10),ylim=c(0,10))
l1<-make.seg(9)
plot(l1)
l2<-make.seg(5)
```

```
plot(l1)
lines(l2)
```

We used this function as above to create a habitat corridor compose of line segments of class `SpatialLines` from the **R** package `sp`. The corridor can be loaded from `scrbook` by typing the command `data(fakecorridor)`. This data list has 2 line files in it (`l1` and `l2`) and a trap locations file (`traps`). We use some functions from the **R** packages `sp` and `rgeos` to join and buffer (by 0.5 units) the two segments. The commands are as follows and the result is shown in Fig. 12.5.

```
data(fakecorridor)
library(sp)
library(rgeos)

buffer<- 0.5
par(mfrow=c(1,1))
aa<-gUnion(l1,l2)
plot(gBuffer(aa,width=buffer),xlim=c(0,10),ylim=c(0,10))
pg<-gBuffer(aa,width=buffer)
pg.coords<- pg@polygons[[1]]@Polygons[[1]]@coords

xg<-seq(0,10,,40)
yg<-seq(10,0,,40)

delta<-mean(diff(xg))
pts<- cbind(sort(rep(xg,40)),rep(yg,40))
points(pts,pch=20,cex=.5)

in.pts<-point.in.polygon(pts[,1],pts[,2],pg.coords[,1],pg.coords[,2])
points(pts[in.pts==1,],pch=20,col="red")
```

In this example, we're not going to estimate parameters of the cost function. Instead, the point is to compute ordinary Euclidean distance but restricted by the boundaries of the corridor (or patch geometry in general) and thus not distance "as the crow flies." To do this, we imagine that animals will tend to severely avoid leaving the buffered habitat zone. Therefore, we assign `cost` = 1 if a pixel is within the buffer, and `cost` = 10000 if a pixel is outside of a buffer. Therefore the cost to move to a neighboring pixel outside of the buffered area is 5000.5 compared to the cost of 1 to move to a neighboring pixel inside the buffer. With this cost specification, we can compute the least-cost path distance matrix one time and modify our likelihood code to accept the distance matrix as input. We give that likelihood in the package `scrbook` as the function `intlik3edv2`. We note also that this function accepts a habitat mask in the form of a vector of 0's and 1's that define any potential state-space restrictions. i.e., 1 if the pixel is an element of the state-space and 0 if it is not, and so additional modifications to the geometry of the region could be made. However, in the analysis of this simulated data set, we define the state-space to be the buffered corridor system. Here we simulate a population of $N = 200$ individuals in the corridor system and so we restrict our state-space accordingly for purposes of fitting the

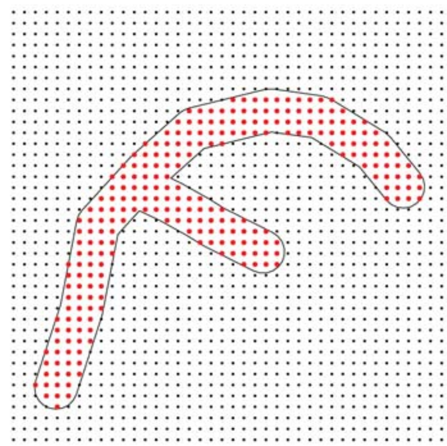

Figure 12.5. A fake wildlife corridor or reserve. The boundary outlines a polygon of suitable habitat surrounded by suburban development.

model. However we encourage you to refit the model without the state-space restriction (for fitting the model only) and then compare the results. The code for doing all of this is in the help file for `intlik3edv2`, which contains the likelihood function and sample **R** script (`?intlik3edv2`).

```
### Define the cost structure
cost<-rep(NA,nrow(pts))
cost[in.pts==1]<-1      # low cost to move among pixels but not 0
cost[in.pts!=1]<-10000  # high cost

### Stuff costs into a raster
library("raster")
r<-raster(nrows=40,ncols=40)
projection(r)<- "+proj=utm +zone=12 +datum=WGS84"
extent(r)<-c(0-delta/2,10+delta/2,0-delta/2,10+delta/2)
values(r)<-matrix(cost,40,40,byrow=FALSE)

# check what it looks like
plot(r)
```

```
points(pts,pch=20,cex=.4)

# compute ecological distances:
library("gdistance")
tr1<-transition(r,transitionFunction=function(x) 1/mean(x),directions=8)
tr1CorrC<-geoCorrection(tr1,type="c",multpl=FALSE,scl=FALSE)
costs1<-costDistance(tr1CorrC,pts)
outD<-as.matrix(costs1)
```

In the next block of code we simulate some data and then fit a model to the simulated data. Note that the object `traps` is loaded with `data(fakecorridor)` along with the data which define the f-shaped patch in Fig. 12.5:

```
library(scrbook)
traplocs<-traps$loc
trap.id<-traps$locid
ntraps<-nrow(traplocs)

set.seed(2013)
N<-200
S.possible<- (1:nrow(pts))[in.pts==1]
S.id<-sample(S.possible,N,replace=TRUE)
S<- pts[S.id,]

D<- outD[S.id,trap.id]
eD<- e2dist(S,traplocs)
Dtraps<-outD[trap.id,]

alpha0<- -1.5
sigma<- 1.5
alpha1<- 1/(2*sigma*sigma)
K<-10

probcap<-plogis(alpha0)*exp(-alpha1*D*D)
Y<-matrix(NA,nrow=N,ncol=ntraps)
for(i in 1:nrow(Y)){
 Y[i,]<-rbinom(ntraps,K,probcap[i,])
}
Y<-Y[apply(Y,1,sum)>0,]

frog1<-nlm(intlik3edv2,c(-2.5,2,log(4)),hessian=TRUE,y=Y,K=K,X=traplocs,
            S=pts,D=Dtraps,inpoly=in.pts)
frog2<-nlm(intlik3edv2,c(-2.5,2,log(4)),hessian=TRUE,y=Y,K=K,X=traplocs,
            S=pts,D=Deuclid,inpoly=in.pts)
```

These two models fit, with the correctly specified ecological distance, constrained by the patch boundaries, and that with the ordinary (misspecified) Euclidean distance are

Table 12.3. Summary output of fitting models to simulated data in which movement is restricted by the habitat corridor shown in Fig. 12.5. The two models fitted were those based on distance constrained by the corridor boundary ("constrained") and a misspecified model based on ordinary Euclidean distance which is "as the crow flies", and cuts through some boundaries. See `?fakecorridor` for the **R** commands to fit these models.

Distance	neg. LL	α_0	α_1	$\log(n_0)$
constrained	-21.892	-1.338	0.332	4.353
Euclidean	-21.128	-1.307	0.382	4.212

summarized in Table 12.3. We find little difference between the two models. In particular, 150 individuals were captured and so truth (the number of uncaptured individuals) is $\log(n_0) = 3.9$. The correct model produces only a slightly more accurate estimate, and it is favored by only 0.7 negative log-likelihood units. Therefore, for this single instance, the results are not too different. This is primarily because the distance between individuals, and traps that they are likely to be captured in, is well-approximated by Euclidean distance.

12.8 ECOLOGICAL DISTANCE AND DENSITY COVARIATES

Habitat characteristics that affect spatial variation in density can also affect home range size and movement behavior. For example, a species that occurs at high density in a forest may be reluctant to venture from a forest patch into an adjacent field. Thus, even if a trap placed in a field is located very close to an animal's activity center, the probability of capture may be very low. In this case, forest cover is a covariate of both density and encounter probability, and we could model it as such by combining the methods described in this chapter with those described in Chapter 11.

To demonstrate, we continue with our analysis of the data shown in Fig 11.4.2. Once again, we suppose that density increases with canopy height, but this time, we also allow home range size to decrease as density increases. This commonly-observed phenomenon can be explained by numerous factors such as intra-specific competition (Sillett et al., 2004) or optimal foraging behavior (Tufto et al., 1996; Saïd and Servanty, 2005).

A question that arises is: Is it possible to estimate the effect of the covariate on density ($\boldsymbol{\beta}_1$) and α_2 using standard SCR data? In other words, can we model spatial variation in density and connectivity at the same time, using standard SCR data? Currently, it is not possible to model least-cost distance using **JAGS** or `secr`, so we wrote our own function, `scrDED`, to fit the model using maximum likelihood. An example analysis is provided on the help page for the function in our **R** package `scrbook`. We briefly note here that the function requires the capture history data, the trap locations, and the raster data formatted using the `raster` package (van Etten, 2012). The linear model for the intensity parameter $\mu(\mathbf{s}, \beta)$ and the least-cost distance function $\text{lcd}(\theta)$ are specified using **R**'s formula interface. A simple function call is

```
fm <- scrDED(y, traplocs=X, den.formula=~elev, dist.formula=~elev,
             rasters=elev.raster)
```

To assess the possibility of estimating both $\boldsymbol{\beta}$ and α_2, we conducted a small simulation study, generating 500 datasets from the model with both parameters set to 1, which corresponds to the conditions described above. The results indicate that it is possible to estimate both parameters (Fig 12.6).

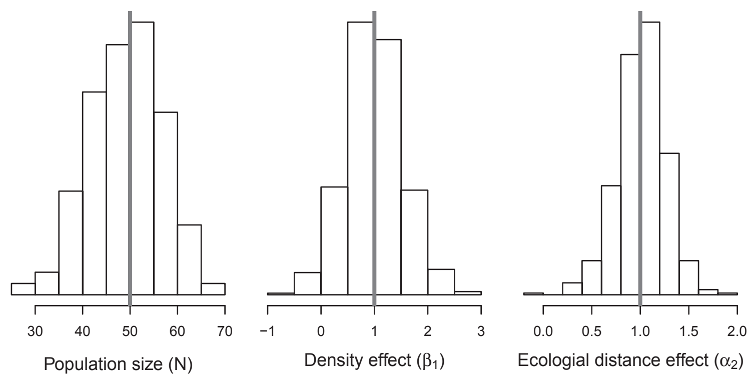

Figure 12.6. Histograms of parameter estimates from 500 simulations under the model in which both density and ecological distance are affected by the same covariate, canopy height. The vertical lines indicate the data-generating value.

12.9 SUMMARY AND OUTLOOK

Almost all published applications of SCR models to date have been based on models for the encounter probability that are functions of the Euclidean distance between individual activity centers and traps. The obvious limitations of such models are that Euclidean distance is unaffected by landscape or habitat structure and implies stationary, isotropic and symmetrical home ranges. These are standard criticisms of the basic SCR model which we have seen many times in referee reports, or heard in discussions with colleagues. However, this should not be seen as criticism that is inherent to the basic conceptual formulation of SCR models because, as we have shown here, one can modify the Euclidean distance metric to accommodate more realistic formulations of distance that allow for inference to be made about landscape connectivity, and model "distance" as a function of local habitat characterists. As such, effective distance between individual home range centers and traps varies depending on the local landscape.

How animals use space and therefore how distance to a trap is perceived by individuals is not something that can ever be known. We can only ever conjure up models to describe this phenomenon and fit those models to limited data on a sample of individuals during a limited amount of time. Here we have shown that there is hope to estimate connectivity parameters that describe how animals use space, from capture-recapture data alone, thereby allowing for irregular home range geometry that is influenced by landscape structure.

In the presence of unctional landscape connectivity, misspecification of the model by an ordinary SCR model based on Euclidean distance produces biased estimates of model parameters (Royle et al., 2012a). This is expected because the effect is similar to failing to model heterogeneity, i.e., if we mis-specify "model M_h" (Otis et al., 1978) with "model M_0" (Otis et al., 1978) then we will expect to under-estimate N. So the effect of mis-specifying the ecological distance metric with a standard homogeneous Euclidean distance has the same effect. In our view, this bias is not really the most important reason to consider models of ecological distance. Rather, inference about the structure of ecological distance is fundamental to many problems in applied and theoretical ecology related to modeling landscape connectivity, corridor and reserve design, population viability analysis, gene flow, and other phenomena. Models based on least-cost path distance allow investigators to evaluate landscape factors that influence movement of individuals over the landscape from non-invasively collected capture-recapture data. Therefore SCR models based on ecological distance metrics might aid in understanding aspects of space usage and movement in animal populations and, ultimately, in addressing conservation-related problems such as corridor design.