Flexible density surface estimation for spatially explicit capture-recapture surveys

D.L. Borchers[1*]   D. Kidney[1]

*1. Centre for Research into Ecological and Environmental Modelling
University of St Andrews, The Observatory, Buchanan Gardens
Fife, KY16 9LZ, ScotlandClaremont 7735, Cape Town, South Africa*

*Corresponding author. dlb@st-andrews.ac.uk

Summary

1. Existing spatially explicit capture-recapture (SECR) software does not have the ability to fit flexible nonparametric models of animal density.

2. We describe and implement in the `R` package `secrgam`, a flexible method for estimating density surfaces from SECR data, using regression splines.

3. Package `secrgam` is an extension of package `secr` to implement some models available in the generalised additive model package `mvcv`. It accommodates density models that are arbitrarily flexible functions of spatially- and temporally-referenced variables. This includes one-dimensional and multi-dimensional smooths of covariates and smooths with interactions. The shape and smoothness of the fitted density surfaces is data-driven and can be determined using AIC or similar criteria. We illustrate use of the package by estimating the density surface from a simulated camera trap survey of leopards.

4. Package `secrgam` provides a flexible tool for species distribution modelling using SECR data.

Key-words: spatially explicit capture-recapture, generalised additive model, species distribution model, density estimation

Introduction

The first spatially explicit capture-recapture (SECR) software was program Density, developed by Efford *et al.* (2004) to implement the inverse prediction method of Efford (2004). The R (R Development Core Team, 2014) package `secr` (Efford, 2013) implementing the maximum likelihood estimation methods of Borchers & Efford (2008) was developed in 2008 and has since been extended substantially. Singh *et al.* (2010) developed the R package `SPACECAP` that implements a Bayesian estimation method.

Theory for modelling non-uniform density surfaces with SECR data using nonhomogeneous Poisson processes (NHPPs) was developed by Borchers & Efford (2008). Since the rate parameter of the NHPP at any point in space is the density of activity centres at the point and the rate parameter of an NHPP changes in space, this kind of model accommodates non-uniform distribution of activity centres. However, with the exception of package `secr`, all estimation software developed to date implements models with uniform distribution. This is frequently an unrealistic model but development of software that implements realistic non-uniform distributions is substantially more difficult than assuming a uniform distribution. This is because the form of dependence of density on spatially-referenced variables is generally unknown and may be non-monotonic (i.e. increasing in some parts of the variable space and decreasing in others), so that flexible, data-driven models for the dependence of density on spatially-referenced variables are required. And it is more difficult to implement such models than uniform density models or models with fixed parametric forms.

Efford (2013) addressed the problem to some extent by allowing density to depend on habitat, geographic and/or other spatially-referenced variables in package `secr`, using generalised linear model link functions to link these variables to density. This allows SECR models in which the rate parameter of the NHPP (i.e. the density) is a parametric function of explanatory variables. For example, if x is an explanatory variable, then with a log link, rate parameters of polynomial form are possible:

$$D(x) \;=\; \exp\left\{\beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_2 x^k\right\} \tag{1}$$

where k can be chosen using some model selection criterion. When density is modelled as a function of more than one explanatory variable, interaction terms can be included.

We extend the capabilities of `secr` by modelling $D(x)$ using regression splines. Further information, including help files and example data and analysis can be obtained by downloading and installing the `secrgam` package from github (https://github.com/david-borchers/secrgam). In what follows we assume some familiarity with the `R` packages `mgcv`, which implements generalized additive models (GAMs; Wood, 2014) and `secr` (Efford, 2013), which implements maximum likelihood estimation for SECR surveys.

Regression splines for density surface modelling

The linear predictor in a polynomial model of degree k can be viewed as a weighted sum of the basis functions x^0, x^1, $x^2, \ldots,$ x^k (in which the para-

meters $\beta_0, \ldots, \beta_k$ are the weights). When used as a means of representing unknown smooth functions, such polynomial basis functions have some undesirable properties (see Section 3.2.1 of Wood, 2006) and so in package `secrgam` we use regression splines instead of polynomials to represent unknown smooth functions.

Regression splines involve a set of locally-defined basis functions ("local" in the sense that each basis function is non-zero only on some relatively small interval of the explanatory variable range). The weighted sum of these functions is the unknown smooth function representation. Specifying spline basis functions involves specifying the kind of basis function (e.g. cubic splines, thin plate splines, B-splines), the number of basis functions to use (k), and the range of each basis function (the covariate values at which it is non-zero). The range is determined by the spline "knot" locations. (see Wood, 2014, Section 3.2.1 for an explanation of knots), the number of which is determined by the kind of basis function and k. As with polynomial regression, in which k is the highest power of the explanatory variable, the greater k is in the case of regression splines, the greater the flexibility of the resulting function. A regression spline representation of $D(x)$ of Equation (1) would be

$$D(x) \quad = \quad \exp\left\{\beta_0 b_0(x) + \beta_1 b_0(x) + \beta_2 b_2(x) + \ldots + \beta_2 b_k(x)\right\} \qquad (2)$$

where $b_0(x), \ldots, b_k(x)$ are k spline basis functions evaluated at x and $\beta_0, \ldots, \beta_k$ are the weights. Three forms for $b_j(x)$ are implemented in package `secrgam`: thin plate regression splines, cubic regression splines and tensor product re-

4

gression splines. Each of these can be functions of single or multiple variables. For example if x and y are explanatory variables one can have

$$D(x, y) = \exp\left\{\sum_j \beta_{x,j} b_{x,j}(x) + \sum_j \beta_{y,j} b_{y,j}(y)\right\} \tag{3}$$

$$\text{or } D(x, y) = \exp\left\{\sum_j \beta_j b_j(x, y)\right\} \tag{4}$$

where $b_{x,j}(x)$ and $b_{y,j}(y)$ are univariate spline basis functions of x and y, and $b_j(x, y)$ is a bivariate spline basis function of both x and y. Equation (4) implicitly includes interaction terms, whereas (4) does not. See Wood (2006) for details of univariate and multivariate spline bases and related issues.

Density surface modelling with `secrgam`

To fit regression spline models of density to SECR data, `secrgam` relies heavily on package `secr`, which does maximum likelihood estimation for a wide range of SECR models, and on the generalised additive model (GAM) package `mvcv`. Package `mvcv` can't be used directly to fit SECR models, because SECR likelihoods are not in the exponential family, but it can readily be used to construct spline basis functions. Having done this, `secr` can be used (after some R object manipulation) to fit regression spline models to SECR data.

The key to fitting regression spline models using `secr` is representing them as linear models (on the link function scale), since `secr` can fit this kind of model. The first step in the process is to evaluate the basis functions at all values of x that are used in fitting the model. (We use x to represent spatially-

referenced explanatory variables, which may be scalar or vector.) Because activity centre locations are unobserved (latent) variables in SECR models, spatially referenced variables at all possible activity centre locations for any detected animal must be considered (see Borchers & Efford, 2008; Efford *et al.*, 2009). A grid of points representing all such locations is called the "mask". Evaluation of the basis functions at every mask point is done using `mvcv`, which returns a design matrix containing $b_j(x_u)$ $(j = 0, \ldots, k, u = 1, \ldots, U)$ for all x_u on the mask, where $x = (x_1, \ldots, x_U)$ is a vector of U explanatory variables. Package `secr` can then be used to fit an SECR model with the design matrix returned by `mvcv`.

So `secrgam` comprises a set of wrapper functions around `secr` that (1) interprets regression spline model specifications and does some R object manipulation (using `mvcv`) to formulate a regression spline model as a linear model on the link function scale (this includes knot placement, which is done by `mvcv`), (2) uses `secr` to fit the model, and (3) manipulates the objects output by `secr` (e.g. prediction, plotting and summaries of point and interval estimates of density). Using `secr` to fit the model has the substantial advantage that it gives users access to the extensive range of SECR models and associated functions implemented in `secr`.

Note that the regression spline models fitted with `secrgam` are similar to GAM models fitted with `mvcv` using the option `fx=TRUE` to fix the smoothness of the fitted model, but they are different from GAM models fitted with `mvcv` using `fx=FALSE` (the `mgcv` default), which implements *penalised* regression splines with automatic smoothness selection. The key differences between the two methods is that in the latter case the smoothness of the fitted function

is determined automatically (by penalising "wiggliness" and using a model selection criterion to find an optimal penalty), whereas it must be specified explicitly via the argument `k` (see below) in `secrgam`, and the user must manually select an appropriate `k` using a model selection criterion. Penalised regression splines are not implemented in `secrgam`.

MODEL SPECIFICATION IN `secrgam`

The syntax for fitting models in `secrgam` is identical to that of `secr` except that (1) function `secrgam.fit` is used in place of `secr.fit` and (2) the GAM model specification syntax of `mvcv` can be used to specify the density model. (A note on terminology: although these models of animal density are not strictly GAMs because the SECR likelihood is not in the exponential family, we refer to them as GAM density models for convenience.)

For example, the code below fits a GAM density model of altitude (variable `alt`) to the SECR capture history object `Boland.CH1` with the mask `Boland.mask1`, using thin plate spline basis functions with $k = 4$.(Thin plate splines are the `mgcv` default when using the "`s( )`" option for smooths; tensor product splines are also available in `secrgam` by using the "`te( )`" option.)

```
model = list(D~s(alt, k=4)) # (specifies the GAM density model)
fit = secrgam.fit(capthist=Boland.CH1, model=model, mask=Boland.mask1)
```

(The structure of masks is described in the help files of `secr`.) The fitted regression spline model can be viewed using `secrgam` (see below).

Camera-trap survey analysis

We illustrate `secrgam` by using it to model density as a function of spatially-referenced variables from a simulated camera-trap survey of leopards. The simulated leopard data are based on a camera-trap survey conducted by the Cape Leopard Trust Boland Leopard Project. Data were simulated from a model fitted to the real data using `secrgam`, but the simulated density was higher than that of the fitted model – to give more power to detect spatial trend and so better illustrate the density modelling capabilities of `secrgam`.
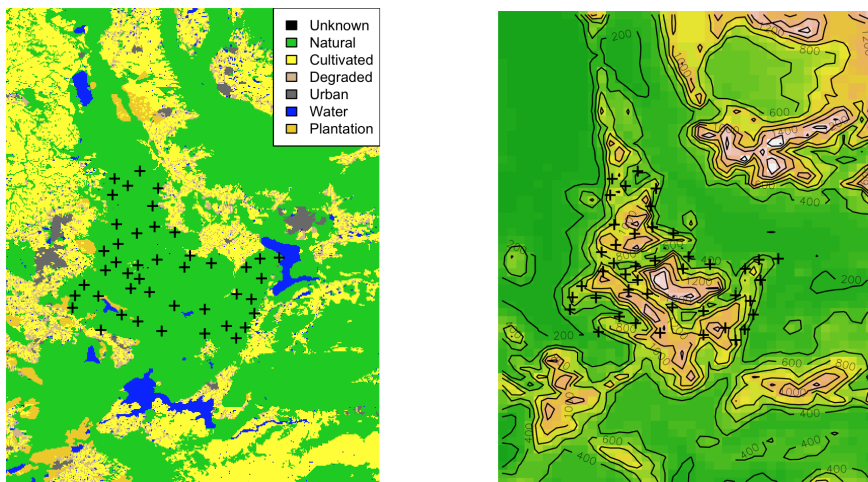

Fig. 1: Landuse class (left) and altitude (right) plots of the Boland camera trap survey area. Black crosses are camera trap locations.

Figure 1 contains image plots of altitude and landuse type in the area, with camera traps overlaid. Two additional variables, the shortest distance to water and shortest distance to urban area, were calculated from the data in Figure 1. Figure 2 shows these variables on the mask (called `Boland.mask1`) used to fit the SECR models below. Recall that the mask contains all activity centre locations of animals that could be detected on the survey; the detection

range on this survey is such that there is negligible probability of detecting an animal with activity centre outside the mask shown in Figure 2. These plots were created using the `image` function of `secrgam`. The plot of distance to water (variable name `dist2.Water`), for example, was created as follows (`asp=1` retains the original aspect ratio):

```
image(Boland.mask1,covariate="dist2.Water",asp=1,main="Distance to Water")
```
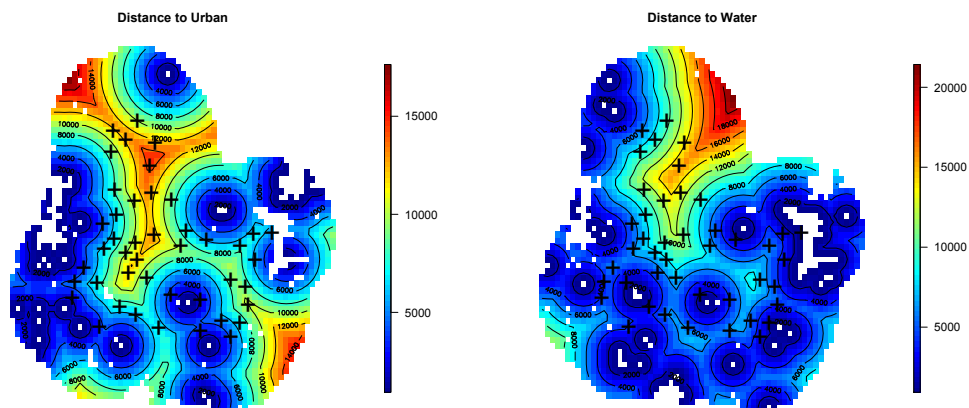

Fig. 2: Distance to urban area (left plot) and distance to water (right plot), both in metres. Black crosses are camera trap locations. Distances are shown for the `secrgam` mask. "Holes" in the mask are urban areas or water, neither of which are suitable leopard habitat and were therefore excluded.

Simulated survey data were created by fitting a GAM density model to the original survey data, using the `secrgam` function `sim.popn.secrgam` to generate a population from this GAM density model (but with higher density than the original model), and the `secr` function `sim.capthist` to generate capture histories from this population using detection function parameters from the original fit. The commands below do this (with 150 activity centres in the mask), taking the `secrgam` fitted object `fit.ex` as the true density

model. The last two commands below plot the capture histories and overlay the locations of the camera traps on this to create a plot like Figure 3. (Some plotting options have been omitted from the last two commands, for brevity. Functions `coef` and `sim.capthist` are `secr` functions, while the two `plot` commands below call the `secr` functions `plot.capthist` and `plot.traps` .)

```
cameras=traps(fit.ex$capthist) # extract the secr traps object
g0=coef(fit.ex)["g0",1] # extract det. fn. parameter g0
sigma=coef(fit.ex)["sigma",1] # extract det. fn. parameter sigma
# simulate the population on the mask:
popn=sim.popn.secrgam(fit.ex,N=150,mask=Boland.mask1)
# simulate a capture history:
Boland.CH1=sim.capthist(cameras,n.occasions=13,
          detecpar=list(g0=g0,sigma=sigma)
# plot the capture histories (rad offsets dots from cameras)
plot(Boland.CH1,rad=750,tracks=TRUE,gridlines=FALSE)
plot(cameras,add=TRUE) # add the camera locations
```

GAM models for density with various combinations of landuse category (`LUfactor`) and smooths of altitude (`alt`), shortest distance to urban area (`dist2.Urban`) and shortest distance to water (`dist2.Water`) were fitted, including linear models and models with k up to to 4. The best model, chosen by AIC, was fitted using the commands below. Also shown below are the commands used to create Figure 4 and the right hand plot in Figure 5.

```
model = list(D~s(alt,k=3)+s(dist2.Water,k=3))
fit1.a3.dW3= secrgam.fit(Boland.CH1,model,Boland.mask1)
plot(fit1.a3.dW3,type="smooth") # (plots of smooths of alt and dist2.Water)
plot(fit1.a3.dW3,type="density",asp=1) # (image plot of est. density on mask)
```
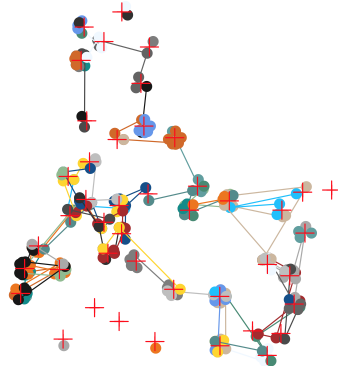
10

Fig. 3: Simulated leopard camera trap data. Different animals are represented by different coloured dots; recaptures of the same animal are linked by lines and camera traps are represented by red crosses.

Plots of smooths of altitude and distance to water, together with 95% confidence limits are shown in Figure 4.

The estimated density surface from the selected model is shown in Figure 5. It can be seen that the estimated density surface is a fairly good representation of the surface from which the data were generated. This does not of course imply that one will reproduce the true density surface this well in practice: our model has the considerable advantage that the true model was in the set of models fitted to the data (in fact it was a model just like the selected model, with `D s(alt,k=3)+s(dist2.Water,k=3)`). It does, however, demonstrate that with the right model and adequate data, it is possible to get a good representation of the underlying non-uniform density model by using `secrgam`. The point estimate of abundance on the mask from the fitted model, obtained using the `secrgam` function `region.ND` (which is a minor extension of the `secr` function `region.N`) is 117 animals with 95% confidence interval (88, 155); recall that simulated abundance is 150.
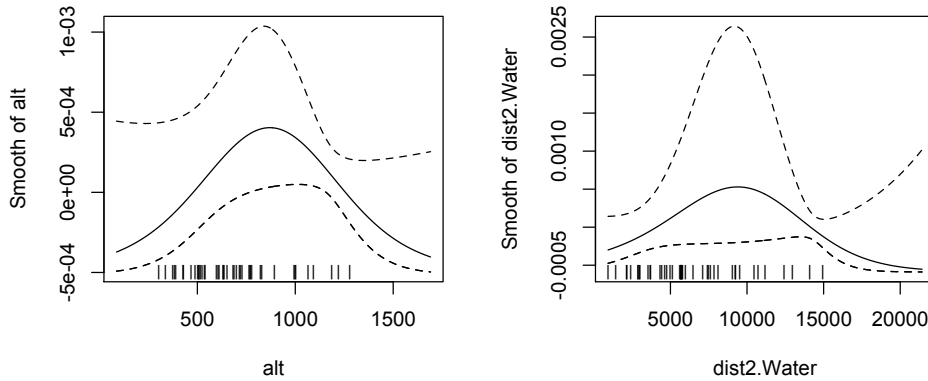
11

Fig. 4: Estimated smooth functions of altitude (left) and distance to water (right). Dashed lines are 95% confidence limits; tick marks on the horizontal axis show locations of the camera traps.

The ability of SECR GAM density models to capture the true distribution of animals in an area of interest will depend on sample size and survey design. The real data sample size (number of animals detected) for this survey was 20, whereas our simulated sample size is 66. With a simulated sample size of 20, a uniform density model was preferred (ΔAIC=10) on the basis of AIC - despite the very considerable non-uniformity in the true density model (shown in the left panel of Figure 5).

The ability of SECR GAM density models to capture the true distribution of animals will also depend on where detectors are located. One can see from the right panel of Figure 1 and the left panel of 4 that there are no camera traps at the highest and lowest altitudes contained in the mask. (The range of altitudes in Figure 4 corresponds to the range in the mask.) Extrapolating from the range of altitudes covered by the cameras to those outside this range is a matter of faith (that the form of the relationship
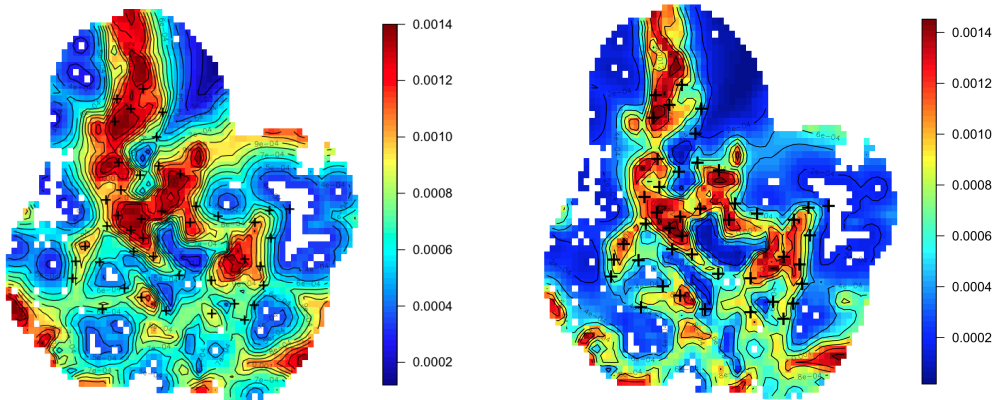
Fig. 5: Plots of true (left) and estimated (right) density in the Boland camera trap survey mask. Black crosses are camera trap locations.

between density and altitude persists beyond the range covered by cameras). If there is interest in density at the more extreme altitudes, a better design would be to locate some cameras at these altitudes. Even if these cameras generate no detections, this is informative about the relationship between altitude and density. In general, surveys that wish to model the dependence of density on explanatory variables should have detectors that span the range of all explanatory variables of interest.

Summary

The R package secrgam provides an extension to the maximum likelihood SECR estimation package secr, to allow flexible data-driven modelling of density. While our example analysis above considers only modelling change in density over space, secrgam also implements smoothing over time

We anticipate that secrgam will prove useful for modelling and under-standing relationships between habitat and other spatially-referenced vari-

ables and density, using SECR survey data.

Although `secrgam` relies on the widely-used GAM package `mvcv` to set up basis functions, the smoothing methods implemented in `secrgam` are not as general as those in `mvcv`. In particular `secrgam` does not use penalised likelihood methods and automated selection of the "wiggliness" of the smooths. This means that the user must determine the wiggliness by trying a variety of values of the smoothness parameter `k` and choosing between models manually using model selection criteria such as AIC. When there are complex relationships between explanatory variables and density (i.e. when the optimal `k`s are large), the model selection process may be very time-consuming and it may not be feasible to consider all plausible models. Although we have analysed only a few datasets using `secrgam`, we have found `k`< 5 to be adequate in all these cases, and manual model selection to be quite feasible. It would nevertheless be useful to have software that implemented a penalised likelihood approach with automated model selection, as does `mvcv`.

Finally, note that `secrgam` does not implement density surface models for conditional likelihoods (see Borchers & Efford, 2008, Section 3.2) because these likelihoods do not involve any model for density.

Acknowledgements

in Figure 1 were obtained from the South African National Biodiversity Institute's site `<http://bgis.sanbi.org/landcover/project.asp>`; the altitude data shown in Figure 1 were obtained from the US National Oceanic and Atmospheric Administration site `<www.ngdc.noaa.gov/mgg/global/>`; we are grateful to both organisations for making these data freely available.

References

Borchers, D.L. & Efford, M.G. (2008) Spatially explicit maximum likelihood methods for capture-recapture studies. *Biometrics*, **64**, 377–385.

Efford, M.G. (2004) Density estimation in live-trapping studies. *Oikos*, **106**, 598–610.

Efford, M.G., Borchers, D.L. & Byrom, A. (2009) *Modeling Demographic Processes in Marked Populations. D.L. Thompson, E.G. Cooch, M.J. Conroy (eds)*, chapter Density estimation by spatially explicit capture-recapture: likelihood-based methods., pp. 255–269. Springer, New York.

Efford, M.G., Dawson, D.K. & Robbins, C.S. (2004) Density: software for analysing capture-recapture data from passive detector arrays. *Animal Biodiversity and Conservation*, **27**, 217–228.

Efford, M. (2013) *secr: Spatially explicit capture-recapture. Version 2.7.0.* Univeristy of Otago, http://www.otago.ac.nz/density.

R Development Core Team (2014) *R: A language and Environment for Statistical Computing. Version 3.1.0.* R Foundation for Statistical Computing, Vienna, Austria.

Singh, P., Gopalaswamy, A., Royle, A., Kumar, N. & Karanth, K. (2010) *SPACE-CAP: A Program to Estimate Animal Abundance and Density using Bayesian Spatially-Explicit Capture-Recapture Models. Version 1.0.*

Wood, S.N. (2006) *Generalized Additive Models.* Chapman & Hall.

Wood, S. (2014) *mgcv: Mixed GAM computation vehicle with GCV/AIC/REML smoothness estimation. Version 1.8.1.*