Adjustment for varying effort in **secr**

Murray Efford

January 23, 2013

When sampling effort varies between detectors or over time in a capture–recapture study we expect a commensurate change in the number of detections. It makes sense to allow for known variation in effort when modelling detections. This simultaneously removes one source of un-modelled heterogeneity in detection probability and relates detection parameters to a consistent unit of effort (e.g. one trap set for one day).

Borchers and Efford (2008) allowed the duration of exposure to vary between sampling occasions in their competing-hazard model for multi-catch traps. The duration (T_s) was a measure of occasion-specific effort. A range of detector types is now acknowledged, each with its own probability model for detections (Efford et al. 2009a,b). We generalise the method for effort to allow joint variation in effort over detectors and over time (occasions), and indicate how effort may be included in models for other detector types. Adjustment for effort is equivalent to the use of an offset variable to allow for varying exposure in generalized linear modelling of counts (McCullagh and Nelder 1989).

Contents

1 Theory

In what follows we use T_{sk} for the effort on occasion s at detector k. It is expected that for small T_{sk} the number of detections increases linearly with T_{sk} (saturation may occur at higher effort, depending on the detector type) and that there are no detections when $T_{sk} = 0$. Examples of possible effort variables are the number of days that each automatic camera was operated in a tiger study, or the number of rub trees sampled for DNA in each grid cell of a grizzly bear study.

The observations to be modelled are either binary (represented by δ_{sk}, an indicator variable for the presence of an animal on occasion s at binary detector k), or integer (represented by y_{sk}, the number of detections on occasion s at count detector k). We assume the probability of detecting an individual declines with the distance $d_k(X)$ between a detector k and the animal's range centre at coordinates $X = (x, y)$. The binary relationship is described by a spatial detection function $g(d_k(X); \theta)$, where θ is a vector of parameters. We define $g(.)$ so that its intercept when $d_k = 0$ is a non-spatial scale parameter g_0 $(0 < g_0 \leq 1)$. For a concrete example, the half-normal detection function uses $g(d_k(X); g_0, \sigma) = g_0 \exp(-d_k(X)^2/(2\sigma^2))$.

If the data are counts rather than binary observations, we may choose to define the spatial detection function as the decline in expected count with distance $\lambda(d_k(X); \theta')$. We use the symbol λ_0 for the intercept $(\lambda_0 > 0)$. For a particular distribution of the counts we can switch back and forth between the binary and expected-count representations (e.g., $g(X) = 1 - e^{-\lambda(X)}$ when the counts are Poisson-distributed). The transformation is non-linear so, for example, a half-normal form for $g(.)$ does not correspond to half-normal form for $\lambda(.)$. Other count models such as the negative binomial may sometimes be required, but we know of no examples of their use. Further details are given by Efford and Borchers (in review).

1.1 Previous methods

Two options were provided for effort adjustment in earlier versions of **secr**:

i. If only a subset of detectors is used on any occasion s, and there is no other variation in effort, T_{sk} is a binary indicator taking the values 0 (detector not used) or 1 (detector used). This case is handled simply by setting log-likelihood components for occasion s and detector k to 0 whenever $T_{sk} = 0$ (via the 'usage' attribute of 'traps' objects in **secr**).

ii. The parameter g_0 or λ_0 may be modelled on an appropriate link scale (logit or log) as a linear function of T_{sk} or other time-varying detector-level covariates.

The first is effective for binary use vs non-use of detectors, but does not encompass other gradations of effort. The second is suboptimal because varying effort is not expected to have a linear additive effect on either of the default link scales, and the estimation of additional parameters is an unnecessary burden.

1.2 Linear hazard models

A more comprehensive approach to effort adjustment follows from the hazard model of Borchers and Efford (2008). We assume detections are independent of each other except as allowed by the competing hazard model for multi-catch traps. The variables to be modelled are δ_{sk}, an indicator variable for the presence of an animal on occasion s at binary detector k, and y_{sk}, the number of detections on occasion s at detector k, if k is a 'count' detector (i.e., one that can record multiple independent occurrences of an animal).

Given a measure of effort on a ratio scale (T_{sk}), it is simple to include effort directly in the formulae for p_{sk} or λ_{sk}. By allowing the instantaneous hazard of detection to increase linearly with T_{sk} we avoid the need to estimate additional parameters (the coefficient is merely g_0 or λ_0 as already fitted, corresponding to $T_{sk} = 1$).

In general we assume the hazard of detection for an individual located at X is related linearly to effort: $h_{sk} = -T_{sk}\ln[1 - g(d_k(X))]$.

If an animal can be detected at most once on any occasion then detectors 'compete' for animals and we require a competing hazard model that uses the summed hazard across all K detectors: $h_{\cdot s}(X) = \sum_{k=1}^{K} h_{sk}(X)$.

The properties of various detector types and the expressions for p_{sk} or λ_{sk} as a function of effort T_{sk} are given in Table 1. The expression $1-[1-g(d_k(X))]^{T_{sk}}$ results from expanding and simplifying $1 - e^{-h_{sk}(X)}$. The expression for binary proximity detectors simplifies to $p_{sk}(X) = g(d_k(X))$ when $T_{sk} = 1$. Only in the Poisson case is the expected number of detections linear on effort. For binomial count detectors we propose a formulation not based directly on instantaneous hazard that is explained more fully below.

Table 1: Including effort in SECR models for various detector types

Detector type	Model	
Multi-catch trap	$\delta_{sk} \sim \text{Bernoulli}(p_{sk})$	$p_{sk}(X) = [1 - e^{-h_{.s}(X)}]h_{sk}(X)/h_{.s}(X)$
Binary proximity	$\delta_{sk} \sim \text{Bernoulli}(p_{sk})$	$p_{sk}(X) = 1 - [1 - g(d_k(X))]^{T_{sk}}$
Poisson count	$y_{sk} \sim \text{Poisson}(\lambda_{sk})$	$\lambda_{sk}(X) = \lambda_0 T_{sk} g(d_k(X))$
Binomial count	$y_{sk} \sim \text{Binomial}(N_{sk}, p_{sk})$	$N_{sk} = T_{sk}$, $p_{sk} = g(d_k(X))$

1.3 Binomial counts

Counts y_{sk} are sometimes modelled as binomial with size N. This arises, for example, when data have been aggregated across a known number of occasions, each representing a binary (Bernoulli-distributed) opportunity of detection (Efford et al. 2009b). N is the aggregate number of opportunities for detection. If the original effort matrix is binary but contains zeros, the 1-D or 2-D aggregate is also likely to vary (i.e., N_{sk} is a non-negative integer specific to the occasion and the detector). Here N_{sk} substitutes for T_{sk} as the measure of effort.

The need to allow for varying effort in a binomial model also arises when data are aggregated across space from binary proximity detectors each used on a different set of occasions. Aggregation across space may be justified and efficient when several detectors are close together, relative to the spatial scale of animal movement and detection. When the original detectors are binary, the number of detections at each aggregated detector cannot exceed the sum of the binary 'usage' values (as in option (i) above).

2 Implementation in secr

The 'usage' attribute of a 'traps' object in **secr** is a K x S matrix recording the effort (T_{sk}) at each detector k ($k = 1...K$) and occasion s ($s = 1...S$). If the attribute is missing (NULL) it will be treated as all ones. Extraction and replacement functions are provided (usage() and usage<-(), as demonstrated below). All detector types accept usage data in the same format, except

- polygon — usage matrix has one row for each polygon
- transect — usage matrix has one row for each transect
- signal strength — usage is not considered when fitting acoustic models
- binomial counts — N_{sk} is determined by `secr.fit` from usage, rounded to an integer, when `binomN = 1`, or equivalently `binomN = 'usage'`

2.1 Data entry

Usage data may be read as extra columns in the text file of detector coordinates (see `read.traps` and secr-datainput.pdf). When only binary (0/1) codes are used, and the `read.traps` argument `binary.usage = TRUE`, separation with white space is optional. This means that '01000' and '0 1 0 0 0' are equivalent. For non-binary values always set `binary.usage = FALSE` and separate with spaces.

The input file for polygons and transects has multiple rows for each unit (one row for each vertex). Usage data are taken from the first vertex for each polygon or transect.

Usage codes may be added to an existing traps object, even after it has been included in a capthist object. For example, the traps object in the demonstration dataset `captdata` starts with no usage attribute:

```
> library(secr, quietly = TRUE)
> usage(traps(captdata))
```

```
NULL
```

Suppose that we knew that traps 14 and 15 caught no animals on occasions 1–3 because they were not set. We could construct and assign a binary usage matrix to indicate this:

```
> mat <- matrix(1, nrow = 100, ncol = 5)
> mat[14:15,1:3] <- 0
> usage(traps(captdata)) <- mat
```

2.2 Model fitting

Following on from the preceding example, we can confirm our assignment and fit a new model:

```
> summary(traps(captdata))
```

```
Object class      traps
Detector type     single
Detector number   100
Average spacing   30 m
x-range           365 635 m
y-range           365 635 m
Usage range by occasion
```

```
    1 2 3 4 5
min 0 0 0 1 1
max 1 1 1 1 1
```

```
> fit <- secr.fit(captdata, trace = FALSE)
> predict(fit)
```

```
       link   estimate SE.estimate       lcl        ucl
D       log  5.4664915  0.64518607  4.341025  6.8837489
g0    logit  0.2766626  0.02734643  0.226373  0.3333109
sigma   log 29.3975886  1.30918234 26.941598 32.0774662
```

The result in this case is only subtly different from the model with uniform usage (compare `predict(secrdemo.0)`).

Usage is 'hardwired' into the traps object, and will be applied (in the sense of Table 1) when a model is fitted with `secr.fit`. There are two ways to suppress this. The first is to remove or replace the usage attribute. For example,

```
> usage(traps(captdata)) <- NULL
```

returns our demonstration dataset to its original state (this would happen in any case when we started a new R session). The second is to bypass the attribute for a single model fit by calling `secr.fit` with `details = list(ignoreusage = TRUE)`.

For a more interesting example, we simulate data from an array of proximity detectors (such as automatic cameras) operated over 5 occasions, using the default density (5/ha) and detection parameters (g0 = 0.2, sigma = 25 m) of `sim.capthist`. We choose to expose all detectors twice as long on occasions 2 and 3 as on occasion 1, and three times as long on occasions 4 and 5:

```
> simgrid <- make.grid(nx = 10, ny = 10, detector = 'proximity')
> usage(simgrid) <- matrix(c(1,2,2,3,3), byrow = TRUE, nrow = 100, ncol = 5)
> simCH <- sim.capthist(simgrid)
> summary(simCH)
```

```
Object class      capthist
Detector type     proximity
Detector number   100
Average spacing   20 m
x-range           0 180 m
y-range           0 180 m
Usage range by occasion
```

```
    1 2 3 4 5
min 1 2 2 3 3
max 1 2 2 3 3
Counts by occasion
                    1   2   3   4   5 Total
n                  21  27  28  24  30   130
u                  21   9   3   1   2    36
f                   8   1   3   9  15    36
M(t+1)             21  30  33  34  36    36
losses              0   0   0   0   0     0
detections         28  65  75  98 103   369
detectors visited  25  44  54  60  65   248
detectors used    100 100 100 100 100   500
```

Now we fit three models with a half-normal detection function. The first implicitly adjusts for effort. The second has no adjustment because we wipe the usage information. The third allows for occasion-to-occasion variation by fitting a separate g0 each time. We use `trace = FALSE` to suppress output from each likelihood evaluation, and drop columns 1 and 2 (model and detectfn) from the AIC table to save space.

```
> fit.usage <- secr.fit(simCH, trace = FALSE)
> usage(traps(simCH)) <- NULL
> fit.null <- secr.fit(simCH, trace = FALSE)
> fit.t <- secr.fit(simCH, model = g0 ~ t, trace = FALSE)
> AIC(fit.usage, fit.null, fit.t)[,-(1:2)]
```

```
          npar    logLik      AIC     AICc  dAICc  AICwt
fit.usage    3 -1052.398 2110.796 2111.546  0.000 0.9383
fit.t        7 -1049.495 2112.990 2116.990  5.444 0.0617
fit.null     3 -1085.161 2176.323 2177.073 65.527 0.0000
```

From the likelihoods we can see that failure to allow for effort (model fit.null) dramatically reduces model fit. The fully time-varying model (fit.t) captures the variation in detection probability, but at the cost of fitting $S-1$ additional parameters. The model with built-in adjustment for effort (fit.usage) has the lowest AIC, but how do the estimates compare? This is a task for the **secr** function `collate`.

```
> collate(fit.usage, fit.null, fit.t, newdata = data.frame(t =
     factor(1:5)))[,,'estimate','g0']
```

```
    fit.usage fit.null     fit.t
t=1 0.2044331 0.397495 0.1511274
t=2 0.2044331 0.397495 0.3479662
t=3 0.2044331 0.397495 0.4044378
t=4 0.2044331 0.397495 0.5285369
t=5 0.2044331 0.397495 0.5710666
```

The null model fits a single 'average' g0 across all occasions that is approximately twice the true rate on occasion 1 (0.2). The estimates of g0 from fit.t mirror the variation in effort. The effort-adjusted model estimates the fundamental rate for one unit of effort (0.2).

```
> collate(fit.usage,fit.null,fit.t)[,,,'D']
```

```
          estimate SE.estimate      lcl      ucl
fit.usage 4.536700   0.7673420 3.264232 6.305203
fit.null  4.527430   0.7660154 3.257229 6.292963
fit.t     4.528942   0.7661376 3.258501 6.294709
```

The density estimates themselves are almost entirely unaffected by the choice of model for g0. This is not unusual. Nevertheless, the example shows how unbalanced data may be analysed with a minimum of fuss.

Adjustment for varying usage will be more critical in analyses where (i) the variation is confounded with temporal (between-session) or spatial variation in density, and (ii) it is important to estimate the temporal or spatial pattern. For example, if detector usage was consistently high in one part of a landscape, while true density was constant, failure to allow for varying usage might produce a spurious density pattern.

2.3 Data manipulation and checking

The various functions in **secr** for manipulating traps and capthist objects (`subset`, `split.traps`, `rbind.capthist`, `MS.capthist`, `join` etc.) attempt to deal with usage intelligently.

When occasions are collapsed or detectors are lumped with the `reduce` method for capthist objects, usage is summed for each aggregated units.

The function `usagePlot` displays a bubble plot of spatially varying detector usage on one occasion. The arguments `markused` and `markvarying` of `plot.traps` may also be useful.

2.4 Polygons and transects

Binary or count data from searches of polygons or transects (Efford 2011) do not raise any new issues for including effort, at least when effort is homogeneous across each polygon or transect. Effects of varying polygon or transect size are automatically accommodated in the models of Efford (2011). Models for varying effort within polygons or transects have not been needed for problems encountered to date. Such variation might in any case be accommodated by splitting the searched areas or transects into smaller units that were more nearly homogeneous (see the `snip()` function for splitting transects).

2.5 Miscellaneous

The units of usage determine the units of g_0 or λ_0 in the fitted model. This must be considered when choosing starting values for likelihood maximisation. Ordinarily one relies on `secr.fit` to determine starting values automatically (via `autoini`), and a simple linear adjustment for usage, averaged across nonzero detectors and occasions, is applied to the value of g0 from `autoini`.

Usage values other than 0 and 1 require significant additional computation because the adjustment is re-computed for each combination of detector x occasion x mask point x detection history x finite mixture. Execution speed may be improved in future versions.

It should be obvious that absolute duration does not always equate with effort. Consider trapping an animal that is most active in the early part of the evening. For example, brushtail possums *Trichosurus vulpecula* are generally caught soon after emerging from their daytime dens at dusk (Cowan and Forrester 2012). Traps set late afternoon and checked early in the morning can be expected to catch at least as many animals as those set in the middle of one day and checked in the middle of the next, despite being open for fewer hours.

3 References

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

Cowan, P. and Forrester, G. (2012) Behavioural responses of brushtail possums to live trapping and implications for trap-catch correction. *Wildlife Research* **39**, 343–349.

Efford, M. G. (2011) Estimation of population density by spatially explicit capture–recapture analysis of data from area searches. *Ecology* **92**, 2202–

2207.

Efford, M. G. and Borchers D. L. Varying effort in capture–recapture studies. In review *Methods in Ecology and Evolution*.

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

Efford, M. G., Dawson, D. K. and Borchers, D. L. (2009) Population density estimated from locations of individuals on a passive detector array. *Ecology* **90**, 2676–2682.

McCullagh, P. and Nelder, J. A. (1989) *Generalized linear models*. Second edition. Chapman & Hall.