Finite mixture models in secr

Murray Efford

June 7, 2013

Variation in detection probability among individuals ('individual heterogeneity') is a persistent problem in capture–recapture studies. Ideally, such variation is modelled by grouping individuals into homogeneous classes (males and females) or including continuous predictors such as body weight. Finite mixture models are an option when unmodelled heterogeneity remains (Pledger 2000; Borchers and Efford 2008). The population is assumed to comprise 2 or more latent classes differing in detection parameters, with an unknown proportion in each class. The likelihood is a weighted sum over the classes.

Mixture models are prone to fitting problems caused by a multimodal likelihood. Some comments are offered below, but a fuller investigation is needed.

The distinction between a finite mixture model and one in which the classes of individuals are known is removed in a hybrid model added in secr 2.6.0 and outlined here.

Implementation in secr

secr allows 2- or 3-class finite mixture models for any 'real' detection parameter (e.g., g0 or sigma of a halfnormal detection function). Consider a simple example, using conditional likelihood and `trace = FALSE` for brevity:

```
> library(secr)
> model.0 <- secr.fit(captdata, model = g0~1, CL = TRUE, trace = FALSE)
```

Specify a 2-class mixture by adding the predictor `h2` to the model formula:

```
> model.h2 <- secr.fit(captdata, model = g0~h2, CL = TRUE, trace = FALSE)
> model.h2

secr.fit(capthist = captdata, model = g0 ~ h2, CL = TRUE, trace = FALSE)
secr 2.6.0, 16:44:02 07 Jun 2013
```

1

```
Detector type      single
Detector number    100
Average spacing    30 m
x-range            365 635 m
y-range            365 635 m
N animals      :   76
N detections   :   235
N occasions    :   5
Mask area      :   21.22711 ha


Model          :   g0~h2 sigma~1 pmix~h2
Fixed (real)   :   none
Detection fn   :   halfnormal
N parameters   :   4
Log likelihood :   -755.6614
AIC            :   1519.323
AICc           :   1519.886


Beta parameters (coefficients)
              beta     SE.beta        lcl        ucl
g0       -0.8108839 0.53576728  -1.860968 0.2392007
g0.h22   -0.8201234 1.34477436  -3.455833 1.8155859
sigma     3.3807520 0.04491538   3.292720 3.4687846
pmix.h22 -1.2846358 5.10870053 -11.297505 8.7282332


Variance-covariance matrix of beta parameters
                  g0        g0.h22        sigma     pmix.h22
g0        0.287046575 0.538478767 -0.001463082   2.57844357
g0.h22    0.538478767 1.808418091  0.004835138   6.09198098
sigma    -0.001463082 0.004835138  0.002017391   0.01908746
pmix.h22  2.578443568 6.091980978  0.019087455  26.09882106


Fitted (real) parameters evaluated at base levels of covariates

 session = 1, h2 = 1
        link    estimate SE.estimate          lcl          ucl
g0     logit  0.3077022   0.1141300 1.345902e-01    0.5595167
sigma    log 29.3928675   1.3208579 2.691596e+01   32.0977061
pmix   logit  0.7832379   0.8673363 1.619221e-04    0.9999876


 session = 1, h2 = 2
```

```
      link   estimate SE.estimate        lcl        ucl
g0   logit  0.1636924   0.2438317 5.928978e-03  0.8652902
sigma  log 29.3928675   1.3208579 2.691596e+01 32.0977061
pmix logit  0.2167621   0.8673363 1.240368e-05  0.9998381
```

From the output you can see that `secr.fit` has expanded the model to include an extra 'real' parameter, `pmix` for the proportions in the respective latent classes. You could specify this yourself as part of the `model` argument, but `secr.fit` knows to add it. The link function for `pmix` defaults to `mlogit` (after the 'mlogit' link in MARK), and in fact any attempt to change the link is ignored.

There are also two extra 'beta' parameters: `g0.h22` which is the difference in g0 between the classes on the link (logit) scale, and `pmix.h22` which is the proportion in the second class, also on the logit scale. Fitted (real) parameter values are reported separately for each mixture class (`h2 = 1` and `h2 = 2`).

We can compare a 2-class finite mixture model to the null (constant) model using AIC:

```
> AIC (model.0, model.h2)


                        model   detectfn npar    logLik
model.0          g0~1 sigma~1 halfnormal    2 -755.9403
model.h2 g0~h2 sigma~1 pmix~h2 halfnormal    4 -755.6614
             AIC    AICc dAICc  AICwt
model.0   1515.881 1516.045 0.000 0.8722
model.h2 1519.323 1519.886 3.841 0.1278
```

In this case there is no reason to prefer the mixture model.

More complex models are allowed. For example, one might, somewhat outlandishly, fit a learned response to capture that differs between two latent classes, while also allowing sigma to differ between classes:

```
> model.h2xbh2s <- secr.fit(captdata, model = list(g0~h2*b, sigma~h2),
    CL = FALSE)
```

Number of classes

The theory of finite mixture models in capture–recapture (Pledger 2000) allows an indefinite number of classes – 2, 3 or perhaps more. Programmatically, the extension to more classes is obvious (e.g., h3 for a 3-class mixture). The

appropriate number of latent classes may be determined by comparing AIC for the fitted models[1].

Looking on the bright side, it is unlikely that you will ever have enough data to support more than 2 classes. For the data in the example above, the 2-class and 3-class models have identical log likelihood to 4 decimal places, while the latter requires 2 extra parameters to be estimated (this is to be expected as the data were simulated from a null model with no heterogeneity).

Multimodality

The likelihood of a finite mixture model may have multiple modes (e.g. Brooks et al. 1997, Pledger 2000). The risk is ever-present that the numerical maximization algorithm will get stuck on a local peak, and in this case the estimates are simply wrong. Slight differences in starting values or numerical method may result in wildly different answers.

The problem has not been explored fully for SECR models, and care is needed. Pledger (2000) recommended fitting a model with more classes as a check in the non-spatial case, but this is not proven to work with SECR models. It is desirable to try different starting values. This can be done simply using another model fit. For example:

```
> model.h2.2 <- secr.fit(captdata, model = g0~h2, start = model.0,
                    CL = TRUE, trace = FALSE)
```

A more time consuming, but illuminating, check on a 2-class model is to plot the profile log likelihood for a range of mixture proportions (Brooks et al. 1997). For this we use the 'fixedbeta' feature of `secr.fit`:

```
> ## fit CL model with range of fixed beta values for mixing proportion
> pmixProfileLL <- function (CH, mask, pmvals = seq(0.01, 0.99, 0.01)) {
    npm <- length(pmvals)
    outCL <- vector('list', npm)
    for (pm in 1:npm) {
        outCL[[pm]] <- secr.fit(CH, CL = TRUE, model = list(g0~h2, sigma~h2),
            details = list(fixedbeta = c(rep(NA,4), logit(pmvals[pm]))),
            mask = mask, trace = FALSE)
    }
    outCL
 }
```

---

[1]score tests (e.g. McCrea, R. S. and Morgan, B. J. T. (2011) Multistate mark-recapture model selection using score tests *Biometrics* 67, 234–241) are not appropriate because the models are not nested, at least that's how it seems to me

```
> ## run function for one year of ovenbird data and plot results
> pmvals <- seq(0.01,0.99,0.01)
> mask <- make.mask(traps(ovenCH[[1]]), nx = 32, buffer = 100)
> outCL <- pmixProfileLL(ovenCH[[1]], mask, pmvals) ## slow!
> plot(pmvals, sapply(outCL, logLik), xlim = c(0,1),
      xlab = 'Fixed pmix', ylab = 'Profile log-likelihood')
```
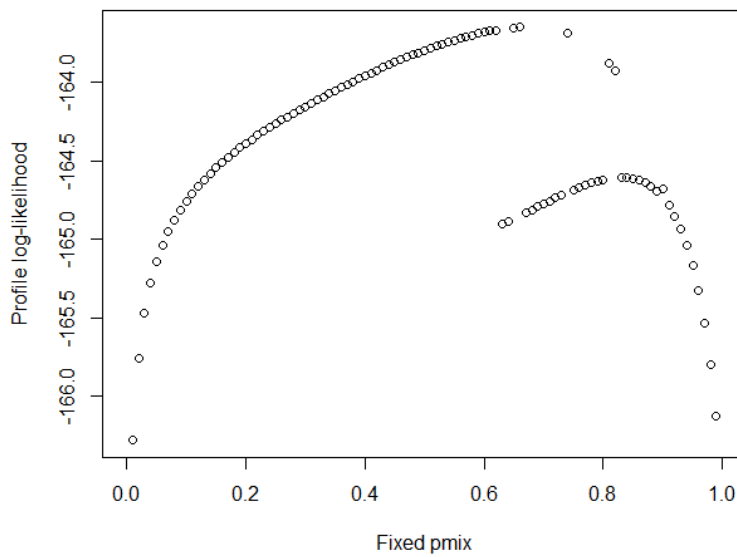

Figure 1: Profile log-likelihood for mixing proportion between 0.01 and 0.99 in a 2-class finite mixture model (ovenbird data 2005).

Multimodality is likely to show up as multiple rounded peaks in the profile likelihood. Label switching (e.g., Stephens 2000) may cause some ghost reflections about pmix = 0.5 that can be ignored. If multimodality is found one should accept only estimates for which the maximized likelihood matches that from the highest peak. In the ovenbird example, the maximized log likelihood of the fitted h2 model was -163.6 and the estimated mixing proportion was 0.67, so the correct maximum was found.

Maximization algorithms (argument 'method' of secr.fit) differ in their tendency to settle on local maxima; 'Nelder-Mead' is probably better than the default 'Newton-Raphson'. Simulated annealing is sometimes advocated, but it is slow and has not been tried with SECR models.

Hybrid 'hcov' model

The hybrid mixture model accepts a categorical (factor) individual covariate for class membership that may be missing (NA) for any fraction of animals. The covariate is specified as argument 'hcov' in secr.fit. If the covariate is missing for all individuals then a full finite mixture model will be fitted (i.e. mixture as a random effect). Otherwise, the random effect applies only to animals of unknown class, and others are modelled with detection parameter values appropriate to their known class. If class is known for all individuals the model is equivalent to a covariate (CL = TRUE) or grouped (CL = FALSE) model. When many or all animals are of known class the mixing parameter may be treated as an estimate of population proportions (probability a randomly selected individual belongs to class m). This is obviously useful for estimating sex ratio free of detection bias.

Notes

It's worth mentioning a perennial issue of interpretation: Do the latent classes in a finite mixture model have biological reality? The answer is 'Probably not' (although the hybrid model blurs this issue). Fitting a finite mixture model does not require or imply that there is a matching structure in the population (discrete types of animal). A mixture model is merely a convenient way to capture heterogeneity.

When more than one real parameter is modelled as a mixture, there is an ambiguity: is the population split once into latent classes common to all real parameters, or is the population split separately for each real parameter? The second option would require a distinct level of the mixing parameter for each real parameter. **secr** implements only the 'common classes' option, which saves one parameter.

References

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

Brooks, S.P., Morgan, B.J.T., Ridout, M.S. and Pack, S.E. (1997) Finite mixture models for proportions. *Biometrics* **53**, 1097–1115.

Pledger, S. (2000) Unified maximum likelihood estimates for closed capture–recapture models using mixtures. *Biometrics* **56**, 434–442.

Stephens, M. (2000) Dealing with label switching in mixture models. *Journal of the Royal Statistical Society. Series B* **62**, 795–809.