Finite mixture models in secr

Murray Efford

March 3, 2010

Variation in detection probability among individuals ('individual heterogeneity') is a persistent problem in capture–recapture studies. Ideally, such variation is removed 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.

Implementation in secr

Version 1.3 of **secr** introduced the 2-class finite mixture model as an option 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)
> data(captdata)
> 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 1.3.0, 12:46:50 11 Mar 2010

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        :  22.09 ha


Model            :  g0~h2 sigma~1 pmix~h2
Fixed (real)     :  none
```

```
Detection fn     :   halfnormal
N parameters     :   4
Log likelihood   :   -755.6554
AIC              :   1519.311
AICc             :   1519.874


Beta parameters (coefficients)
                beta     SE.beta         lcl         ucl
g0        -0.8108346  0.53679358   -1.862931   0.2412614
g0.h22    -0.8196927  1.34342244   -3.452752   1.8133669
sigma      3.3808856  0.04494949    3.292786   3.4689850
pmix.h22  -1.2827990  5.11429840  -11.306640   8.7410417


Variance-covariance matrix of beta parameters
                   g0        g0.h22        sigma      pmix.h22
g0        0.288147343  0.539084179  -0.001455749   2.58666132
g0.h22    0.539084179  1.804783856   0.004839211   6.09243471
sigma    -0.001455749  0.004839211   0.002020456   0.01916520
pmix.h22  2.586661324  6.092434709   0.019165195  26.15604813


Fitted (real) parameters evaluated at base levels of covariates

 session = 1, h2 = 1
        link    estimate SE.estimate         lcl         ucl
g0     logit   0.3077127   0.1143508   0.1343618   0.5600245
sigma    log  29.3967941   1.3220385  26.9177589  32.1041401
pmix   logit   0.7829258          NA          NA          NA


 session = 1, h2 = 2
        link    estimate SE.estimate         lcl         ucl
g0     logit   0.1637581  0.07350927  0.06400653   0.3592936
sigma    log  29.3967941  1.32203850  26.91775888  32.1041401
pmix   logit   0.2170742          NA          NA          NA
```

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. 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       AIC       AICc
model.0            g0~1 sigma~1 halfnormal    2  -755.9344  1515.869  1516.033
model.h2 g0~h2 sigma~1 pmix~h2 halfnormal    4  -755.6554  1519.311  1519.874
        dAICc   AICwt
```

```
model.0  0.000 0.8722
model.h2 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].

At this time you are advised not to fit more than 2 classes in **secr** because there are technical difficulties with the link function for `pmix`. This defaults to `mlogit` (after the 'mlogit' link in MARK), and in fact any attempt to change the link is ignored.

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).

Notes

It's worth mentioning a perennial issue of interpretation: Do the latent classes have biological reality? The answer is No. 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.
Pledger, S. (2000) Unified maximum likelihood estimates for closed capture–recapture models using mixtures. *Biometrics* **56**, 434–442.

---

[1]score tests (e.g. McCrea and Morgan *Biometrics* In press) are not appropriate because the models are not nested, at least that's how it seems to me