Multiple detection functions in density surface models

David L Miller

September 10, 2019

CREEM, University of St Andrews, Scotland

Abstract

We often want to combine data from multiple surveys into one spatial model. In this case we usually want to include multiple detection functions and use them with a single GAM. Here I show how to do that.

1 Introduction

Generally for a DSM we have:

$$\mathbb{E}\left[n_i|\boldsymbol{\beta}, \boldsymbol{\lambda}, p(\hat{\boldsymbol{\theta}}; \mathbf{z}_i)\right] = a_i p(\hat{\boldsymbol{\theta}}; \mathbf{z}_i) \exp\left(\beta_0 + \sum_m f_m(x_{im})\right), \tag{1}$$

where the number of individuals per segment (of area a_i), n_i and follows some count distribution such as quasi-Poisson, Tweedie or negative binomial (where above we assume a log link). The f_m are smooth functions of environmental covariates, x_{im}, represented by a basis expansion (i.e., $f_m(x) = \sum_j \beta_j b_j(x)$ for some basis functions b_j) penalized by a (sum of) quadratic penalty (or penalties); β_0 is an intercept term, included in parameter vector $\boldsymbol{\beta}$; $\boldsymbol{\lambda}$ is a vector of smoothing parameters which control the wiggliness of the smooth components of the model.

When we have multiple detection functions, we write $p(\hat{\boldsymbol{\theta}}; \mathbf{z}_i)$ as a piecewise function:

$$p(\hat{\boldsymbol{\theta}}; \mathbf{z}_i) = \begin{cases} p(\hat{\boldsymbol{\theta}}_1; \mathbf{z}_i) & \text{observation } i \text{ in detection function 1} \\ \vdots & \\ p(\hat{\boldsymbol{\theta}}_2; \mathbf{z}_i) & \text{observation } i \text{ in detection function 2} \\ \vdots & \\ p(\hat{\boldsymbol{\theta}}_K; \mathbf{z}_i) & \text{observation } i \text{ in detection function } K \end{cases},$$

so that a given observation is related to one detection function only (indexed by $k = 1, \ldots, K$). No assumption is made about the specific form of the detection function. We can concatenate all of the detection function parameters $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2, \ldots, \hat{\boldsymbol{\theta}}_K)$. We can then fit model (1) as usual.

1.1 Variance estimation

The above only addresses mean effects, what about variance?

1.1.1 Delta method

Calculate $\text{CV}(\hat{N}) = \sqrt{\text{CV}_{\text{GAM}}(\hat{N})^2 + \sum_k \text{CV}(\hat{p}_k)^2}$. There, the $\text{Var}_{\text{GAM}}(\hat{N})$ (hence $\text{CV}_{\text{GAM}}(\hat{N})$) is calculated using the usual GAM estimator (see the varprop paper for details).

1.1.2 Variance via variance propagation

Thinking about the variance propagation method of Bravington, Miller and Hedley (2018)[1], we need not only the $p(\hat{\boldsymbol{\theta}}; \mathbf{z}_i)$s but also their derivatives wrt $\hat{\boldsymbol{\theta}}$ and the Hessian corresponding to the detection functions. Following from the varprop paper, we fit the following model to estimate variance:

$$\log \mathbb{E}\left[n_i | \boldsymbol{\beta}, \boldsymbol{\lambda}, \hat{p}_i\right] = \log a_i \hat{p}_i + X_i \boldsymbol{\beta} + \kappa_i \boldsymbol{\delta},$$

defining the vectors $\boldsymbol{\delta} \triangleq \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}_0$ and $\kappa_i \triangleq \left.\frac{d \log p(\boldsymbol{\theta}, z_i)}{d\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}$. Extending this to our case of multiple detection functions, the definition of $\boldsymbol{\delta}$ follows simply and $\kappa_i \triangleq \left.\frac{d \log p(\boldsymbol{\theta}, z_i)}{d\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}$, where entries where i does not belong to detection function k are 0.

We then also need to form the covariance matrix for the detection functions. We assume no covariance between the detection functions[2], so we have:

$$\mathbf{V}_{\boldsymbol{\theta}} = \begin{pmatrix} \mathbf{V}_{\boldsymbol{\theta_1}} & 0 & \ldots & 0 \\ 0 & \mathbf{V}_{\boldsymbol{\theta}_2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & \mathbf{V}_{\boldsymbol{\theta}_K} \end{pmatrix}.$$

This can then be used as the covariance matrix for the random effect $\boldsymbol{\delta}$.

1.2 Differing density by platform/observation type

Predictions can be made as normal but we may want to include a factor to account for the different underlying densities for each detection function mode

[1] https://arxiv.org/abs/1807.07996

[2] There might be cases where we have some information about covariance between the detection functions, but we ignore that for now.

(for example if there are flying vs. on water birds). In that case we might formulate our model as:

$$\mathbb{E}\left[n_i|\boldsymbol{\beta}, \boldsymbol{\lambda}, p(\hat{\boldsymbol{\theta}}; \mathbf{z}_i)\right] = a_i p(\hat{\boldsymbol{\theta}}; \mathbf{z}_i) \exp\left(\beta_0 + \beta_{\text{state}}\texttt{state}_i + \sum_m f_m(x_{im})\right), \quad (2)$$

in which case we need to make predictions for each of the levels of`state` and appropriately combine them. In this case an effort-weighted sum seems appropriate, since we want an "average" over the multiple platforms (since we are effectively treating the different types/platforms as separate effort in the model). So then:

$$\hat{N} = \sum_{k=1}^{K} \frac{L_k}{\sum_{k=1}^{K} L_k} \hat{N}_k,$$

if L_k is the total effort in platform k (of which there are K) and $\hat{N}_k$ is the abundance estimate for platform k. To get to the variance we have:

$$\text{Var}\left(\hat{N}\right) = \sum_{k=1}^{K} \left(\frac{L_k}{\sum_{k=1}^{K} L_k}\right)^2 \text{Var}\left(\hat{N}_k\right),$$

where $\text{Var}\left(\hat{N}_k\right)$ is the variance estimate for platform k. Notably, these estimates are really simple when all K platforms have the same effort (they are simple averages).

2 Implementation

To implement this in `dsm` you need to be able to identify each observation as being from one detection function. You also need to be able to know which segments relate to a given detection function. This leads to some implementation differences in dsm.

Detection function: One detection function for each data subset (e.g., per cruise etc).

Observation table: The observation table is unchanged (it seems easiest to concatenate the tables used to fit the detection function) *but* you must ensure that the object IDs (column `object`) are unique and match those used to fit the detection functions.

Segment table: Additional column `ddfobj`, which refers to which detection function is used for each set of segments.

The call to `dsm` now lets you use a list of detection functions, the order of the detection functions in the list relates to the numbering in the `ddfobj` column in the segment table.

3 Some special cases and changes to dsm

This approach allows us to do some other stuff that we couldn't do before in dsm.

3.1 Strip transects

A new function (dummy_ddf) is used to construct dummy detection functions for use with dsm. This takes the object IDs, group sizes, truncation and transect type and constructs a model object that can be used with dsm. This replaces the strip.width way of specifying strip transects. This also works with whatever one might call the point transect equivalent ("circle transects"?).

3.2 Changes

- No longer need to supply the transect= argument, this is determined from the (dummy) detection functions. This means you can mix points and line and strips and circles.
- No strip.width argument, as this is handled by dummy detection functions.
- No transect argument as this can always be determined from the detection functions (dummy or not).