Uncertainty estimation in DSM

Aim:

This document aims to shed some light on the divergence in results between the bootstrap and variance propagation methods for quantifying uncertainty in density surface models. We want to quantify uncertainty in the DSM, this has two sources: spatial uncertainty, detection function uncertainty. In particular, we see large discrepancies in the CV and plots of the CV over the prediction grid.

What are the desirable properties for a procedure to find the uncertainty?

- want CVs to be estimated accurately
- probably don't mind if they are a little conservative (i.e. if there is some overcoverage)
- *but* they should also have good coverage (it's easy to make them huge, we don't really want that)
- would like the method to be fast so that we can produce CV maps for all candidate models in reasonable time

Conventions

- DSM refers to the density surface modelling approach of Hedley and Buckland (2004).
- `dsm` refers to the R package.
- MVB refers to Mark Bravington

Problem deconstruction

In `dsm` as it stands, we bootstrap the residuals of the model (optionally, using moving blocks), then "add in" the uncertainty from the detection function via the delta method. The "new" bootstrap regenerates new distances from the fitted detection function and then uses them to re-fit the model.

MVB's variance propagation (Williams et al, 2011) method combines an estimate of the GAM uncertainty (for the spatial model, using standard results from GAM theory) and the detection function uncertainty by calculating the variance from an extended model which includes an extra term which accounts for the uncertainty in the detection function (the Hessian from the detection function fitting).

Rather than thinking about how to combine the detection function and GAM uncertainties, let's think just about the GAM since this is the larger conceptual difference between the two methods. First a more detailed outline of the two methods.

Bootstrap

Having created a set of blocks of segments, denote the fitted values for the model to be $\hat{\eta}$. For $b = 1, \ldots, B$ (where B is the number of resamples required):

1. Resample (with replacement) the residuals in a block-wise fashion, store the values in $\mathbf{r}_b$.
2. Refit the model but with the response set to $\hat{\eta} + \mathbf{r}_b$ (where $\hat{\eta}$ are the fitted values from the orginal model).
3. Take the predicted values (over the region you're interested in) for the new model and store them.

Summaries can then be found, to create CV maps, or percentile intervals. Outlier removal can be done using the Tukey method (outliers from a boxplot), but this method of outlier detection appears unsatisfactory in some cases.

Issues with the bootstrap are as follows:

- *breakdown of spatial structure* – moving around residuals (even in blocks) does not properly account for the spatial structure. Intuitively this seems like it would over-inflate the variances
- *belief in the response distribution* – we assume that the response distribution is "correct" to move the residuals around
- *slow* – obviously, this is a bootstrap so it's bound to take a long time

Using some GAM theory

When we have the identity link, finding the variance is relatively easy. Need to use the "`lpmatrix`" (Wood, 2006; page 245), that is the matrix $\mathbf{X}_p$ such that:

$$\hat{\eta}_p = \mathbf{X}_p \hat{\beta}$$

ie. it maps the model parameters to the linear predictor. This is useful for calculating variance estimates. Since we can obtain $\mathbf{V}_\beta$, parameter covariance matrix, we can then use $\mathbf{X}_p$ to find the covariance matrix for the linear predictor using:

$$\mathbf{V}_{\hat{\eta}_p} = X_p \mathbf{V}_{\hat{\beta}} X_p^{\mathrm{T}}$$

linear functions of the linear predictor can be calculated using this method (just changing the pre- and post-multipier). In the non-linear case, it's a bit more tricky. First note that the posterior for the parameters (given the data) is multivariate normal with its mean being the parameter estimates and the covariance matrix of the parameters. (i.e. $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$).

The following algorithm is suggested by Wood (2006, page 246) (and appears to work, eg Marra, Miller and Zanin (2011)):

1. For $b = 1, \ldots, N_b$ do the following:
   a. Simulate from $\beta \sim N(\hat{\beta}, \mathbf{V}_{\hat{\beta}})$, to obtain β_b.
   b. Calculate $\hat{\eta}_b = \exp(\mathbf{X}_p \beta_b)$ (e.g. if we are using the log-link)
   c. Sum over the survey area
2. Calculate the appropriate summary statistics, e.g. median, 95% quantiles etc over b.

MVB, though suggests that this is unneccessary, since:

"to deal with nonlinearity in the link [...]. Could be done by simulation instead, and that would be more accurate (if you did enough). However, in my limited experience: once you've got a CV so big that the delta-method doesn't work, then your estimate is officially Crap and there is not much point in expending extra effort to work out exactly how Crap!"

So he simply computes

$$\left(\frac{\partial^2 \log_e \eta}{\partial \eta^2}\Big|_{\eta=\hat{\eta}} \circledast \mathbf{X}_p \right) \mathbf{V}_p \left(\frac{\partial^2 \log_e \eta}{\partial \eta^2}\Big|_{\eta=\hat{\eta}} \circledast \mathbf{X}_p \right)^{\mathrm{T}}$$

where $\frac{\partial^2 \log_e \eta}{\partial \eta^2}\Big|_{\eta=\hat{\eta}}$ is the vector of second derivatives of the link evaluated at the values of the linear predictor and $\circledast$ denotes `R`-style matrix-vector multiplication. So it then beefs up the variance based (roughly) on the uncertainty in the linear predictor.

Issues here:

- *bias* – smoothers are biased over small areas, this could be an issue with the CV map, but this effects the bootstrap too!
- *complicated* – most people understand the bootstrap well enough, where as this is more mathsy

- *does the last bit work?* – the last "trick" here with the derivatives we just have to take MVB's word for it. (We don't have to use it – instead we could use the simulation approach.)

Univariate vs multivariate smooths?

As a side note, the difference between the two methods is most pronounced when bivariate smooths of space (`~s(x,y)` in `mgcv` notation) rather than two separate functions of space (`s(x)+s(y)`), I think this is down to the former having more flexibility, so more complicated functions can be fitted. I've only thoroughly looked into this with the dolphin data in practise and that is a very small data set of 47 observations, which is pushing the limits for bivariate smoothing, but I think the above arguments hold in any case.

What do I think?

I think that these more standard methods using standard GAM theory make much more sense than using the bootstrap. It seems from the outside like the bootstrap doesn't make as many assumptions but actually we are making some rather strong claims about what is going on in the model (i.e. the error distribution and the correlation structure – or lack thereof). These "new" GAM intervals are not only quick to compute, they are comparable between models and have a better theoretical basis.

References

Hedley, S.L. & Buckland, S.T. (2004) Spatial models for line transect sampling. Journal of Agricultural, Biological, and Environmental Statistics, 9, 181–199.

Marra, G., Miller, D.L. & Zanin, L. (2011) Modelling the spatiotemporal distribution of the incidence of resident foreign population. Statistica Neerlandica.

Williams, R., Hedley, S.L., Branch, T.A., Bravington, M.V., Zerbini, A.N. & Findlay, K.P. (2011) Chilean Blue Whales as a Case Study to Illustrate Methods to Estimate Abundance and Evaluate Conservation Status of Rare Species. Conservation Biology, 25, 526–535.

Wood, S.N. (2006) Generalized Additive Models: an Introduction with R. Chapman & Hall/CRC.