Supplementary Material A: Island Scrub-jay analysis

Mark V. Bravington
David L. Miller
Sharon L. Hedley

April 15, 2020

1 Introduction

This analysis is based on data from the paper:

Sillett, T. Scott, Richard B. Chandler, J. Andrew Royle, Marc Kery, and Scott A. Morrison. 'Hierarchical Distance-Sampling Models to Estimate Population Size and Habitat-Specific Abundance of an Island Endemic'. Ecological Applications 22, no. 7 (2012): 1997-2006

Data is available at: `https://figshare.com/articles/Supplement_1_R_code_data_and_grid_covariates_used_in_the_analyses_/3517754`. Script `get_issj_data.R` converts this to a `dsm`-compatible format (or use `issj.RData`).

The authors analyse 2 point transect surveys of Island Scrub-jays from Santa Cruz Island, CA. Each survey consists of 307 sample locations (point transects), one ocurring in fall 2008 and the other in spring 2009. Distances were binned into 3 intervals. Locations of the points were available, along with vegetation (proportion forest and proportion chaparral) and elevation. They dredged the data (see their table 2), but their two final models were:

Fall 2008 model: abundance was given as $\beta_0 + \beta_1\texttt{chaparrel}^2 + \beta_2\texttt{chaparrel} + \beta_3\texttt{elevation}$, detectability as a function of `chaparral`.

Spring 2009 model: abundance was given as $\beta_0 + \beta_1\texttt{chaparrel}^2 + \beta_2\texttt{chaparrel} + \beta_3\texttt{elevation}^2 + \beta_4\texttt{elevation}$, detectability as a function of `forest`.

In both cases, abundance was assumed negative binomially distributed and a half-normal detection function was fitted.

2 Analysis

Reproducing the analysis from Sillet et al. First loading the requisite data and R packages.

```
# load data
```

```
load("issj.RData")

# load packages
library(Distance)

## Loading required package:  mrds
## This is mrds 2.2.2
## Built:  R 3.6.2; ; 2020-04-02 15:55:48 UTC; unix
##
## Attaching package:  ’Distance’
## The following object is masked from ’package:mrds’:
##
##     create.bins
```

```
library(dsm)

## Loading required package:  mgcv
## Loading required package:  nlme
## This is mgcv 1.8-31.  For overview type ’help("mgcv-package")’.
## Loading required package:  numDeriv
## This is dsm 2.2.18
## Built:  R 3.6.2; ; 2020-03-23 09:57:26 UTC; unix
```

2.1 Fall 2008

Running models:

```
# fit detection function
```

```
df_fall <- ds(obs_fall, transect="point", formula=~chaparral)

## Columns "distbegin" and "distend" in data:  performing a binned analysis...
## Model contains covariate term(s):  no adjustment terms will be included.
## Fitting half-normal key function
## AIC= 330.658
## No survey area information supplied, only estimating detection function.

# fit spatial model
ll_fall <- dsm(count~chaparral +
                     I(chaparral^2) +
                     elevation,
               observation.data=obs_fall, segment.data=segs, ddf.obj=df_fall,
               transect="point", family=nb())

# check parameter estimates
summary(ll_fall)

##
## Family: Negative Binomial(0.343)
## Link function: log
##
## Formula:
## count ~ chaparral + I(chaparral^2) + elevation + offset(off.set)
##
## Parametric coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -11.7170     0.1682 -69.656  < 2e-16 ***
## chaparral        1.4288     0.1906   7.496 6.57e-14 ***
## I(chaparral^2)  -0.3789     0.1255  -3.019  0.00254 **
## elevation       -0.2260     0.1478  -1.528  0.12639
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =  0.026   Deviance explained = 25.9%
## -REML = 270.86  Scale est. = 1          n = 307
```

We can then do the variance propagation via the `dsm_varprop` function and compare to the delta methd (implemented in the `dsm.var.gam` function):

```
# varprop
```

```
pp_fall <- dsm_varprop(ll_fall, pred)
pp_fall

## Summary of uncertainty in a density surface model calculated
##  by variance propagation.
##
## Probability of detection in fitted model and variance model
##    chaparral Fitted.model Fitted.model.se Refitted.model
## 1 -0.8613374    0.3345638      0.04953141      0.3383015
## 2  0.1733685    0.2350608      0.02187221      0.2361407
## 3  2.0242947    0.1166973      0.01900894      0.1154638
##
## Approximate asymptotic confidence interval:
##     2.5%     Mean    97.5%
## 1625.431 2271.870 3175.400
## (Using log-Normal approximation)
##
## Detection function CV          : 0.08974614
##
## Point estimate                 : 2271.87
## Standard error                 : 390.9639
## Coefficient of variation       : 0.1721

# delta method for comparison
vg_fall <- dsm.var.gam(ll_fall, pred, off.set=pred$off.set)
vg_fall

## Summary of uncertainty in a density surface model calculated
##  analytically for GAM, with delta method
##
## Approximate asymptotic confidence interval:
##     2.5%     Mean    97.5%
## 1614.289 2270.436 3193.281
## (Using log-Normal approximation)
##
## Point estimate                 : 2270.436
## CV of detection function       : 0.08974614
## CV from GAM                    : 0.1506
## Total standard error           : 398.1164
## Total coefficient of variation : 0.1753
```

2.2 Spring 2009

Again fitting models:

```
# fit detection function
```

```
df_spring <- ds(obs_spring, transect="point", formula=~forest)

## Columns "distbegin" and "distend" in data:  performing a binned analysis...
## Model contains covariate term(s):  no adjustment terms will be included.
## Fitting half-normal key function
## AIC= 303.83
## No survey area information supplied, only estimating detection function.

# fit spatial model
ll_spring <- dsm(count~chaparral +
                       I(chaparral^2) +
                       elevation +
                       I(elevation^2),
          observation.data=obs_spring, segment.data=segs, ddf.obj=df_spring,
          transect="point", family=nb())

# check parameter estimates
summary(ll_spring)

##
## Family: Negative Binomial(0.672)
## Link function: log
##
## Formula:
## count ~ chaparral + I(chaparral^2) + elevation + I(elevation^2) +
##     offset(off.set)
##
## Parametric coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -11.43571    0.17886 -63.936  < 2e-16 ***
## chaparral        0.68155    0.15642   4.357 1.32e-05 ***
## I(chaparral^2)  -0.31515    0.10730  -2.937  0.00331 **
## elevation       -0.08065    0.15006  -0.537  0.59098
## I(elevation^2)  -0.33865    0.15125  -2.239  0.02515 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =  0.0727   Deviance explained = 12.3%
## -REML = 276.52  Scale est. = 1         n = 307
```

Compare variance calculations:

```
# varprop
```

```
pp_spring <- dsm_varprop(ll_spring, pred, trace=TRUE)
pp_spring

## Summary of uncertainty in a density surface model calculated
##  by variance propagation.
##
## Probability of detection in fitted model and variance model
##       forest Fitted.model Fitted.model.se Refitted.model
## 1 -0.4633652    0.2710221      0.02760400      0.2514265
## 2 -0.3424751    0.2604019      0.02555611      0.2468799
## 3  2.5274876    0.0933240      0.03014374      0.1569675
##
## Approximate asymptotic confidence interval:
##     2.5%     Mean    97.5%
## 1262.702 1684.184 2246.355
## (Using log-Normal approximation)
##
## Detection function CV          : 0.1420396
##
## Point estimate                 : 1684.184
## Standard error                 : 248.8427
## Coefficient of variation       : 0.1478

# delta method for comparison
vg_spring <- dsm.var.gam(ll_spring, pred, off.set=pred$off.set)
vg_spring

## Summary of uncertainty in a density surface model calculated
##  analytically for GAM, with delta method
##
## Approximate asymptotic confidence interval:
##     2.5%     Mean    97.5%
## 1187.849 1697.548 2425.956
## (Using log-Normal approximation)
##
## Point estimate                 : 1697.548
## CV of detection function       : 0.1420396
## CV from GAM                    : 0.1165
## Total standard error           : 311.8206
## Total coefficient of variation : 0.1837
```

3 Comparing to results in Sillet et al.

Table 1 compares the parameter estimates from Sillet et al. (their Table 3) with the results from using the DSMs described above.

To obtain uncertainty estimates of detection function parameters we require the following function to recompute the Hessian for that model component.

```
# wee function to get a "corrected" detection function
```

Submodel and		Fall 2008			Spring 2009	
coefficient	Sillet et al.	Pre-varprop	Post-varprop	Sillet et al.	Pre-varprop	Post-varprop
Abundance						
Chaparral	1.43 (0.229)	1.43 (0.191)	1.44 (0.209)	0.67 (0.667)	0.68 (0.156)	0.67 (0.156)
Chaparral^2	-0.38 (0.115)	-0.38 (0.126)	-0.38 (0.126)	-0.29 (-0.291)	-0.32 (0.107)	-0.29 (0.106)
Elevation	-0.23 (0.146)	-0.23 (0.148)	-0.23 (0.148)	-0.11 (-0.107)	-0.08 (0.15)	-0.11 (0.148)
Elevation^2				-0.34 (0.148)	-0.34 (0.151)	-0.34 (0.15)
Dispersion	0.36 (0.0777)	0.34	0.34	0.78 (0.239)	0.71	0.71
Detection						
Intercept	4.68 (0.0658)	4.67 (0.055)	4.68 (0.055)	4.63 (0.540)	4.63 (0.051)	4.64 (0.05)
Chaparral	-0.20 (0.060)	-0.19 (0.052)	-0.2 (0.052)			
Forest				-0.09 (0.043)	-0.18 (0.06)	-0.08 (0.062)

Table 1: Comparison of parameter estimates from Sillet et al. to those using DSM (before and after our variance propagation procedure). Note that `mgcv` (and hence the `dsm` package) does not provide uncertainty around the negative binomial parameter. Abundance intercept parameters are not comparable and therefore omitted (due to differences in how the offset is calculated in each model).

```r
# note this only works for half-normal models
# the function's argument object is the result from dsm_varprop
fix_ddf_vp <- function(object){
  # get the model from the varprop object
  fix_ddf <- object$old_model$ddf
  # get parameters
  which.names <- !(names(coef(object$refit)) %in% names(coef(object$old_model)))
  newpar <- fix_ddf$par + coef(object$refit)[which.names]

  # setup a new fitting call
  fix_call <- fix_ddf$call
  fix_ddf$control$initial <- list()
  fix_ddf$control$initial$scale <- newpar

  # ensure that the model is not fitted!
  # Just want to calculate the Hessian!
  fix_ddf$control$maxiter <- 0
  fix_ddf$control$nrefits <- 0
  fix_ddf$control$refit <- FALSE
  fix_ddf$control$nofit <- TRUE

  # run the model and return it
  fix_ddf <- eval(fix_call, fix_ddf)
  invisible(fix_ddf)
}


# corrected for fall
vp_ddf_fall <- fix_ddf_vp(pp_fall)

# corrected for spring
vp_ddf_spring <- fix_ddf_vp(pp_spring)
```

Table 2 compares the abundance estimates from from Sillet et al. to the estimates here.

	Fall 2008	Spring 2009
Sillet et al.	2267 (1613–3007)	1705 (1212–2369)
DSM	2272 (1625-3175)	1684 (1263-2246)

Table 2: Comparison of abundance estimates between Sillet et al. and the above analysis.

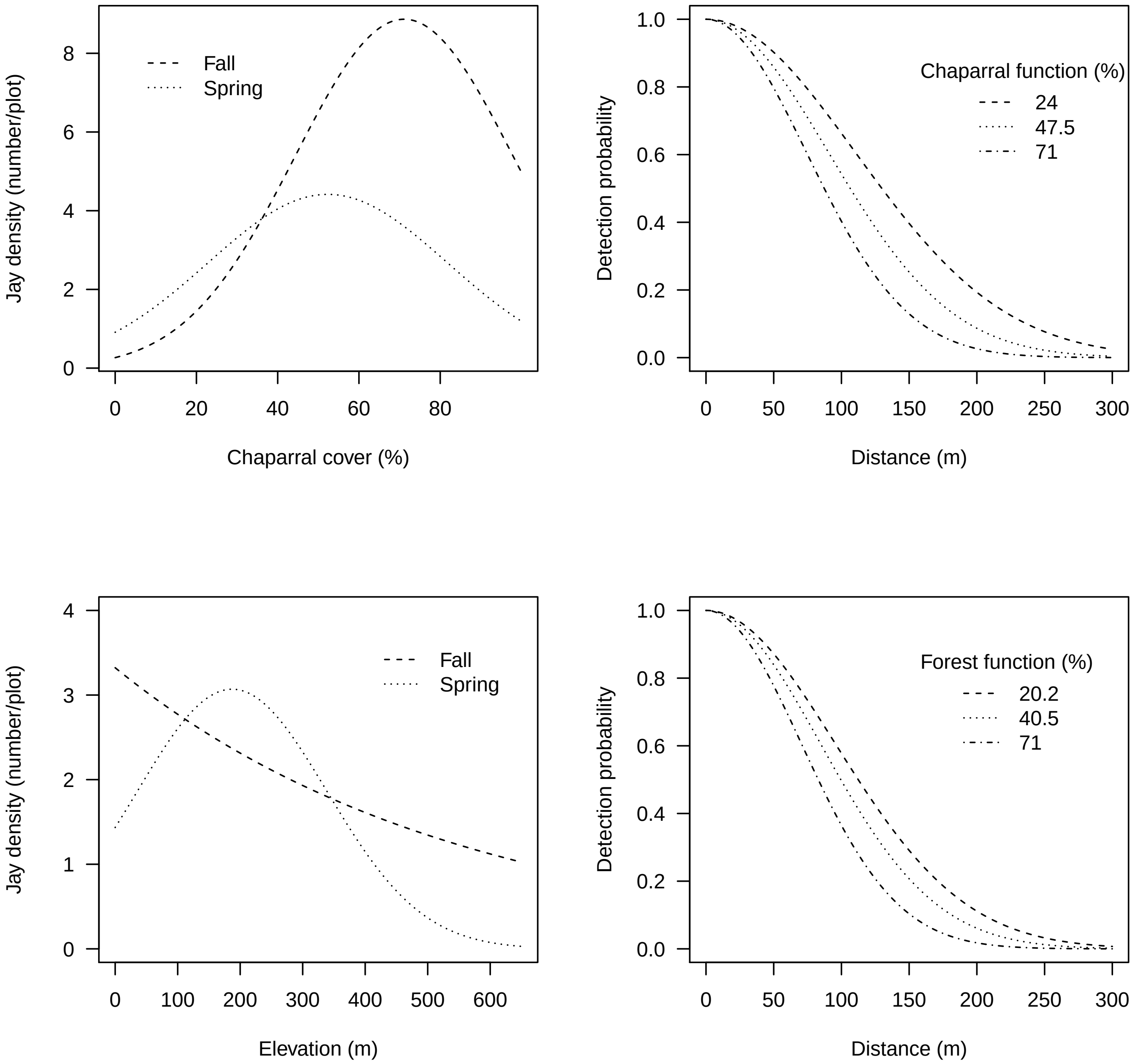

Figure 1: Reproduction of Figure 4 from Sillet et al. using the models specified above. Left side shows abundance per site as a function of the covariates in each time period. Right side shows the detection function at selected covariate levels for fall (top) and spring (bottom). These are indistinguishable from those in the article.

4 Unbalanced data

Since the coverage of both covariate and geographical space is extremely good, we decided to subsample the data to see what effect that had on our models. We used the fall data only for this experiment and randomly selected 100 points. We then removed points where where chaparral cover was greater than the mean chaparral proportion (with mean calcualted over all points).

```
# reproducible subsample
set.seed(1889)

# take 100 sites at random
sub_segs <- segs[sample(1:nrow(segs), 100), ]
# only the observations at those sites are to be included
sub_obs <- obs_fall[obs_fall$Sample.Label %in% sub_segs$Sample.Label, ]

# since the data were z-transformed, mean=0
# so chaparral<=0 gives values at or lower than the mean
sub_segs <- sub_segs[sub_segs$chaparral<=0, ]
sub_obs <- sub_obs[sub_obs$chaparral<=0, ]
```

Having subsampled the data, we can then fit models:

```
# fall detection function on subsampled data
```

```
sub_df <- ds(sub_obs, transect="point", formula=~chaparral)

## Columns "distbegin" and "distend" in data:  performing a binned analysis...
## Model contains covariate term(s):  no adjustment terms will be included.
## Fitting half-normal key function
## AIC= 31.544
## No survey area information supplied, only estimating detection function.

# fall GAM
sub_ll <- dsm(count~chaparral +
              I(chaparral^2) +
              elevation,
              observation.data=sub_obs, segment.data=sub_segs, ddf.obj=sub_df,
              transect="point", family=nb())

# inspect coefficient estimates
summary(sub_ll)

##
## Family: Negative Binomial(0.222)
## Link function: log
##
## Formula:
## count ~ chaparral + I(chaparral^2) + elevation + offset(off.set)
##
## Parametric coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -10.8052     1.6186  -6.676 2.46e-11 ***
## chaparral        3.7674     6.4030   0.588    0.556
## I(chaparral^2)  -0.9836     5.1426  -0.191    0.848
## elevation       -0.6331     0.4901  -1.292    0.196
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =  0.167   Deviance explained = 50.5%
## -REML = 26.603  Scale est. = 1         n = 65
```

Compare variance calculations:

```
# variance propagation:
```

	Pre-varprop	Post-varprop
Full data	2270 (1614-3193)	2272 (1625-3175)
Subsample	26434 (209-3348966)	2831 (39-206407)

Table 3: Comparison of abundance estimates before and after subsampling.

```
sub_pp <- dsm_varprop(sub_ll, pred)
sub_pp

## Summary of uncertainty in a density surface model calculated
##  by variance propagation.
##
## Probability of detection in fitted model and variance model
##    chaparral Fitted.model Fitted.model.se Refitted.model
## 1 -0.9059225    0.7806223       0.5482305      0.6638134
## 2 -0.3043294    0.2713778       0.1688496      0.2792179
## 3 -0.1975517    0.1967457       0.1929185      0.2233795
##
## Approximate asymptotic confidence interval:
##          2.5%         Mean        97.5%
##     38.84062   2831.42634 206406.98577
## (Using log-Normal approximation)
##
## Detection function CV          : 0.7106491
##
## Point estimate                 : 2831.426
## Standard error                 : 30908.15
## Coefficient of variation       : 10.9161

# delta method
sub_vg <- dsm.var.gam(sub_ll, pred, off.set=pred$off.set)
sub_vg

## Summary of uncertainty in a density surface model calculated
##  analytically for GAM, with delta method
##
## Approximate asymptotic confidence interval:
##        2.5%        Mean       97.5%
##     208.649   26434.038 3348965.752
## (Using log-Normal approximation)
##
## Point estimate                 : 26434.04
## CV of detection function       : 0.7106491
## CV from GAM                    : 21.1064
## Total standard error           : 558242.5
## Total coefficient of variation : 21.1183
```

Get corrected detection function:

```
sub_vp_ddf <- fix_ddf_vp(sub_pp)
```

A summary of the abundance results before and after variance propagation with those from the full data are given in Table 3. Comparison of model coefficients is given in Table 4.

Submodel and	Full data		Subsample	
coefficient	Sillet et al	DSM	Pre-varprop	Post-varprop
Abundance				
Chaparral	1.43 (0.229)	1.44 (0.209)	3.77 (6.403)	2.56 (9.17)
Chaparral^2	-0.38 (0.115)	-0.38 (0.126)	-0.98 (5.143)	-1.74 (6.541)
Elevation	-0.23 (0.146)	-0.23 (0.148)	-0.63 (0.49)	-0.63 (0.49)
Dispersion	0.36 (0.0777)	0.34	0.22	0.22
Detection				
Intercept	4.68 (0.0658)	4.68 (0.055)	4.23 (0.955)	4.39 (0.867)
Chaparral	-0.20 (0.060)	-0.2 (0.052)	-1.61 (2.587)	-1.14 (2.071)

Table 4: Comparison of parameter estimates from the full fall data from Sillet et al. and those using DSM, along with estimates for the subsampled data before and after our variance propagation procedure. Note that `mgcv` (and hence the `dsm` package) does not provide uncertainty around the negative binomial parameter. Intercept parameters are not comparable and therefore omitted (due to differences in how the offset is calculated in each model).