Ecography

E5646

Gorresen, P. M., McMillan, G. P., Camp, R. J. and Pratt, T. K. 2009. A spatial model of bird abundance as adjusted for detection probability. – Ecography 32: 291–298.

Supplementary material

General outline of species-habitat model

We present a two-step, spatially explicit, Bayesian hierarchical regression model of species-habitat relationships. The first step models abundance given habitat covariates and higher-level temporal and spatial effects using methods presented by Link and Sauer (2002) and Thogmartin et al. (2004). In the second step, the effective sample area estimated by distance analysis is used to correct model predictions for undetected individuals.

Preliminary model steps

- Generate habitat variable values for all point observations within study area. Standardize habitat variables to improve model performance. Standardize by subtracing variable values by the mean, then dividing by the square-root of the variance.

- Assess correlation among explanatory variables. Exclude one of a pair of variables with correlation coefficients >|0.80|.

- Select habitat variables associated with counts using a generalized linear model with a Poisson distribution and a log link. Add variables in a step-wise manner and assess variable significance at each step with a likelihood ratio test.

- Calculate seasonal effect values with a unimodal sine and a cosine function for all dates (converted to Julian days) associated with survey observations. Functions are calculated as "=SIN(2 * ACOS(–1) * [input Julian day here]/365)" and "=COS(2 * ACOS(–1) * [input Julian day here]/365)".

- Calculate spatial dependence. In ArcGIS, develop grid cells of desired size (note that a small size may result in a large number of cells that slow computation; ideally grid cell size should match the scale ultimately used to calculate density; e.g. 100 × 100 m for 1-ha density values). Clip cells by study area boundary. Assign unique cell identity numbers. Develop an adjacency matrix for the spatial neighborhood with the GeoBUGS module in WinBUGS (details at <www.biostat.umn.edu/~brad/yuecui/index.html>). Generate habitat variable values for all cells within study area.

Bayesian model formulation

- Develop hierarchical models incorporating habitat and latent variables in WinBUGS.

- Specify a model in the BUGS language. Shown here is an example of model specification code in WinBugs 1.4. Syntax in later versions of WinBugs may differ. Code as follows:

```
# Model example includes 3 habitat variables (X1, X2, X3) and incorporates
# the seasonal effect as a random effect represented by the terms SINE and COSINE.

Model name;        #add descriptive name before running
{          # open model
#### counts and overdispersion effects ####
           ##priors##
mu ~ dflat()
```

```
beta1 ~ dnorm(0.0, 1.0E-6)#specify desired distribution, mean and precision
beta2 ~ dnorm(0.0, 1.0E-6)
beta3 ~ dnorm(0.0, 1.0E-6)
beta4 ~ dnorm(0.0, 1.0E-6)
beta5 ~ dnorm(0.0, 1.0E-6)

for(k in 1: ncounts)
{log(lambda[k]) <– mu + spatial[grid[k]] + yeareffect[year[k]] + beta1*X1[k] + beta2*X2[k] + beta3*X3[k] +
beta4*SINE[k] + beta5*COSINE[k]
count[k] ~ dpois(lambda[k])}

#### spatial (grid) effects ######
spatial[1:ngrids] ~ car.normal(adj[], weights[], num[], taugrid)
for(j in 1:sumNumNeigh)
{weights[j] <– 1}                # assigns weight of 1 if cells are adjacent, zero otherwise
taugrid <– 1/(sdgrid*sdgrid)
sdgrid ~ dunif(0,10)

#### year effects ####
yeareffect[1] <– 0  # specify year to be treated as fixed effect
for(y in 2:nyears)
{yeareffect[y] ~ dnorm(0.0, 0.0001)}
}           #close model
```

- The MCMC sampler must be given initial values for each variable and for each chain. These can be arbitrary values, although in practice, convergence can be poor if wildly inappropriate values are chosen. Refer to the WinBugs User Manual on guidance in generating initial values. Initial values may be appended to the code for model specification.

- Monitor parameter values and check convergence of model chains. After discarding parameter values from the part of the iterations (a "burn-in" period; e.g. 25 000 iterations), assess model convergence from the posterior distribution of the remaining part (e.g. 10 000 iterations). Following convergence, re-run model with a single chain for another 1000 iterations to compute the posterior samples of model coefficients. In the Sample Monitor Tool, select "Coda" to generate an ASCII file of model coefficients to be used for predicting densities.

Distance analysis of detection probability and effective area sampled

- Apply a current program of Distance (Thomas et al. 2005) to develop distance-based models of animal detectability for the purpose of calculating the **effective area** (EA) sampled for each species. Model selection methods are described in detail by Buckland et al. (2001, 2004). General steps as follows:

  - Plot histograms of recorded distances. Use Model Definition with arbitrary model and manually define large number of intervals in the Detection Function Model Diagnostics. Look for: heaping, evasive movement, outliers and possible gross errors.
  - Determine suitable truncation distances (e.g. levels of g(0.1), 10%, etc.).
  - Select candidate models and run as individual analyses.
  - Select levels for estimating parameters.
  - Run analyses and select "best-fit" models.
  - Proceed with Multiple Covariate Distance Sampling analysis. Use model without adjustments and build up the number of adjustments for "best-fit". Default in MCDS is zero adjustments.
  - Combine stratum estimates to produce global estimates. Use Bootstrap variance estimates and place all models in one Model Definition. This approach will account for the variance due to uncertainty in model selection, among other sources of variance.

Prediction of densities

- Apply custom SAS script to integrate the posterior samples from each of the 1000 iterations output in the WinBugs CODA with the values of the habitat and latent variables for all grid cells in the study area. SAS script references a macro (`BUGtoSAS.sas`) that converts BUGS output files to a SAS dataset (macro detailed below).

- Values to be input are indicated with brackets ("[ ]") in the following SAS script.

```
%let area_co = 1/[set "EA" value here];   **** "EA" is effective area
sampled value generated by Distance Analysis to adjust for unobserved
birds and for obtaining 1-ha abundance estimates.  Set this value for
each species. ****
%let datapath = C:\[set pathname]Public\bugs\iiwi;    **enter the path
to the excel spreadsheet and to the index and chain files ***;
%let iterations = 1000; *** can be as big as 1000, but this will take a
long time!! ***;

%include "C:\[set pathname]\BUGtoSAS.sas";  *** Set path to the macro
BUGtoSAS.sas ***;

%BUGtoSAS(full,&datapath.\,[set file name].index.txt, [set file
name].chain.txt);  *** References the two files in the CODA output
generated by WinBugs.  Set filename. ***
libname outdat "&datapath";
data outdat.iiwi5b2b;
      set full;
run;

/*
data init;
      set full;
            seed=floor(1000000000*(sqrt(time())-floor(sqrt(time()))));
            k=rannor(seed);
run;
proc sort data = init; by k;
run;
*/
data iterdat;
      set full;
            iter = _n_;
            if _n_ <= &iterations.;
run;

*** create a macro variable with the spatial effect name list ***;
proc sql;
      select name
     into :varlist separated by ' '
     from dictionary.columns
    where libname = "WORK" and
          memname = "FULL"
                  and substr(name,1,3) = "spa";
quit;

***** import the habitat data for grids into which predictions will be
made ***;
PROC IMPORT OUT= WORK.preddat1
```

```
            DATAFILE= "&datapath.\ [set file name].xls"   *** Set file
name of habitat data in Excel spreadsheet. ***
            DBMS=EXCEL2000 REPLACE;
     GETNAMES=YES;
RUN;


%macro prediter;
proc delete data = mastercount; run;
proc delete data = predict; run;
proc delete data = outsum; run;
proc delete data = sumdata; run;

data mastercount;
      retain iterset grid_id id count_hat .;
run;
data sumdata;
      retain iterset total_birds .;
run;


%do i = 1 %to &iterations.;
proc sql;
create table predinput as
      select p.*, i.*
            from preddat1 as p, iterdat(where = (iter=&i.)) as i;
quit;

************* predictions for yeareffect***change as needed*****;
data predict(keep = iterset grid_id id count_hat);
      set predinput;
      iterset = &i.;
      seed=floor(1000000000*(sqrt(time())-floor(sqrt(time()))));
      array sp{*} &varlist.;
      lambda = &area_co.*exp(mu + yeareffect_X_ + beta1*X1 + beta2*X2 +
beta3*X3 + beta6*SINE + beta7*COSINE + sp[grid_id]);   *** Example
model statement.  Define model here. Specify year to be treated as
fixed effect.***
      count_hat = ranpoi(seed, lambda);
run;

proc summary data = predict sum nway;
      var count_hat;
      id iterset;
      output out = outsum sum = t;
run;

**** Put the sum of all predicted counts (=total birds) into the
sumdata dataset;
proc sql; insert into sumdata select iterset, sum(t) as total_birds
from outsum; quit;

***** Append the grid level predicted counts to the mastercount
dataset. *****;
proc append base = mastercount data = predict; run;
%end;
%mend prediter;
```

```
%prediter;

PROC PRINTTO PRINT='C:\[set pathname]\[set filename].lst' NEW;  *** Set
pathname and filename for list file.  ***
RUN;
************** Get posterior summaries for each hectare *************;
proc summary data = mastercount mean nway std;
      var count_hat;
      class id;
      where iterset ne .;
      output out = mapdat mean = predcount std = st_dev;
run;

********* Get posterior summaries for the total count ****;
proc univariate data = sumdata;
      var total_birds;
      where iterset ne .;
run;

PROC PRINTTO PRINT=PRINT;
RUN;

PROC EXPORT DATA= WORK.Mapdat
            OUTFILE= "C:\[set pathname]\[set filename].xls" *** Set
pathname and filename for output spreadsheet file.  ***
            DBMS=EXCEL2000 REPLACE;
RUN;
```

- Copy into the working directory the following SAS macro for converting BUGS output files to a SAS dataset.

```
%macro BUGtoSAS(newdata,direc,ind,out,stats) ;

    %put ;
    %put Written by Matthew Hayat, Division of Biostatistics, Medical
College of Wisconsin ;
    %put Last Updated 4/27/99 ;
    %put ;

    %let ind = "%cmpres(&direc&ind)" ;
    %let out = "%cmpres(&direc&out)" ;

    data info ind (keep=ratio) ;
      length varname $ 20;
     infile &ind end=lstrec  delimiter='09'x;
     input varname $ first last ;
      output info ;
      ratio = last / _n_ ;
      if lstrec then output ind ;
      call symput("numrow",_n_) ;

    data out (keep=value) ;
     infile &out delimiter='09'x;
     input sampnum value ;

    proc transpose data=info out=&newdata ;
```

```
     var first ;
     id varname ;

    data &newdata (drop=_name_) ;
     set &newdata ;
     if _n_ < 1 ;

    data all

   %if &stats = stats %then %do ;

      temp

   %end ; ;

     if _n_ < 1 ;

    %do i = 1 %to &numrow ;

      data v&i ;
       if _n_ = 1 then set ind ;
       set out ;
       if (ratio*%eval(&i)-ratio+1) le _n_ le (ratio*%eval(&i)) ;
       v&i = value ;

      data all ;
       merge all v&i ;

        %if &stats = stats %then %do ;

          proc univariate noprint data=v&i ;
           var v&i ;
           output out=v&i n=n mean=mean var=var std=std min=min max=max
                          range=range pctlpts=2.5 median=median
pctlpts=97.5
                          kurtosis=kurtosis skewness=skewness pctlpre=p
;

          data temp ;
           set temp v&i ;

        %end ;

    %end ;

    data &newdata ;
     set all (keep=v1-v%cmpres(&numrow)) &newdata ;

    data &newdata (drop=i v1-v%cmpres(&numrow)) ;
     set &newdata ;
     array numer(*) _numeric_ ;
     array vvec(*) v1-v%cmpres(&numrow) ;
     do i = 1 to dim(vvec) ;
      numer(%eval(&numrow)+i)= vvec(i) ;
     end ;

      %if &stats = stats %then %do ;
```

```
        data temp ;
         merge info (keep=varname) temp ;

        proc print label noobs uniform data=temp ;
         title "Summary Statistics" ;
         var n mean var std min max range p2_5 median p97_5 kurtosis
skewness ;
         id varname ;
         label varname  = 'Variable'
               n         = 'N'
               mean      = 'Mean'
               var       = 'Variance'
               std       = 'Standard Deviation'
               min       = 'Minimum'
               max       = 'Maximum'
               range     = 'Range'
               p2_5      = '2.5%'
               median    = '50%'
               p97_5     = '97.5%'
               kurtosis = 'Kurtosis'
               skewness = 'Skewness' ;

      %end ;

    * proc datasets ;
    * delete v1-v%cmpres(&numrow) all out ind info temp ;

    title " " ;

    run ;

 %mend ;
```