Ecography

E5798
Bierman, S. M., Butler, A., Marion, G. and Kühn, I. 2010. Bayesian image restoration models for combining expert knowledge on recording activity with species distribution data. – Ecography 33: xxx–xxx.

Supplementary material

Appendix 1. WinBugs code to implement the autologistic model within a Bayesian Image Restoration framework

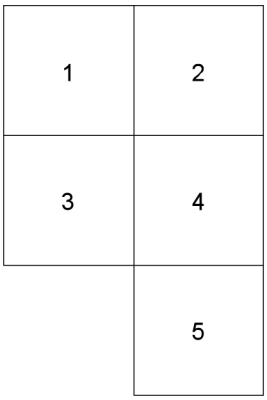
All text after the # character are comments. We explain how to compute the neighborhood weight in detail below. In this example, 3 environmental covariates are used as regressors to predict probabilities of species presence/absence: precipitation (precip[i]), area of forest (forest[i]), and temperature (temp[i]). The index $i = (1, 2, ..., N)$ is used to loop through all the grid cells.

```
model{

# occ: unobserved truth (2=present, 1=absent)
# obs: recorded state (2=present, 1=absent)

        for(i in 1:N){

        occ[i] ~ dcat(p.occ[i,]);

        obs[i] ~ dcat( p.obs[control[i],occ[i],] );

# compute the neighborhood weighting, based on the unobserved true
# distribution of the species: further explanation below

        weight[i] <- ( ( occ[adj[start[i]]] + occ[adj[start[i]+1]] + occ[adj[start[i]+2]] + occ[adj[start[i]+3]] +
        occ[adj[start[i]+4]] + occ[adj[start[i]+5]] + occ[adj[start[i]+6]] + occ[adj[start[i]+7]] - ( (8-num[i]) *
        occ[1]) - num[i] ) / num[i] ) - 0.5;

# the presence/absence model: the autologistic model

        logit(p.occ[i,2]) <- alpha + b1*precip[i] + b2*temp[i] + b3 * forest[i] + tau*weight[i];

        p.occ[i,1] <- 1-p.occ[i,2];

}

# the image restoration model
# k =1...9: levels of the control variable (9 categories of grid cells)
# q[k] is the detection probability of a grid cell of category k

for ( k in 1:9 ) {
        p.obs[k,1,1] <- 1;
        p.obs[k,1,2] <- 0;
        p.obs[k,2,1] <- 1-q[k];
```

```
        p.obs[k,2,2] <- q[k];
        }

# priors

        for ( j in 1:9 ) {
        q[j] ~ dbeta(1,1);
        }

        alpha ~ dflat();
        b1 ~ dflat();
        b2 ~ dflat();
        b3 ~ dflat();
        tau ~ dflat();

}
```

How to compute the neighborhood weight

Here, we explain how to compute the neighborhood weighting (variable "weight[i]" in the code above) based on the latent true distribution of the species in each of the iterations of the Markov chain, using an example of a grid with just 5 cells:



1) Create a vector "num[i]" of the number of neighbors for each of the grid cells. In our example of the grid with 5 cells: num = {3,3,3,4,2}: grid cells 1,2 and 3 have 3 neighbors each, grid cell 4 has 4 neighbors, and grid cell 5 has 2 neighbors.

2) Create a vector "adj" of the identifiers of all neighbors of each grid cell. The list of neighbors for each grid cell are: grid cell 1: {2,3,4}, grid cell 2: {1,3,4}, grid cell 3: {1,2,4}, grid cell 4: { 1,2,3,5}, grid cell 5: {3,4}. For convenience, since the maximum number of potential neighbors is 8, we fill the individual neighborhood vectors with the identifier of a known but arbitrary grid cell number, say 1. Thus, the vectors of neighborhood indices now are: grid cell 1: {2,3,4,1,1,1,1,1}, grid cell 2: {1,3,4,1,1,1,1,1} , grid cell 3: {1,2,4,1,1,1,1,1}, grid cell 4: { 1,2,3,5,1,1,1,1}, grid cell 5: {3,4,1,1,1,1,1,1}. The vector adj is simply created by combining all these vectors: adj = {2,3,4,1,1,1,1,1, 1,3,4,1,1,1,1,1, 1,2,4,1,1,1,1,1, 1,2,3,5,1,1,1,1, 3,4,1,1,1,1,1,1}

3) Create a vector "start[i]", which gives the starting point in the vector adj where the set of identifiers for the neighbors of each grid cell starts. This vector can be computed using the num variable as follows: start[i] = $1 + \sum_{j=1}^{i-1} \text{num}[j]$. For example: start[1] = 1, start[2] = 1 + num[1], start[3] = 1 + num[1] + num[2], etcetera. Thus, in our 5-grid cell example, start={1,4,7,10,14}

4) Now, using the vectors num, adj and start, we can compute the neighbourhood weight for each grid cell by adding up the latent states of the neighboring grid cells occ[i] (occ[i] = 2 for occupied, occ[i] = 1 for unoccupied). The identifiers of the neighbors of grid cell i are the elements start[i], start[i]+1, …,start[i]+7 in the vector "adj". For grid cells with fewer than 8 neighbors, we need to subtract the occupancy states of the grid cell (a known grid cell, "1" in our case), that were added in the list: (8 – num[i]) × occ[i]. Finally, we

divide by the number of neighbors, num[i] to obtain a weighting between 0 and 1, and subtract a value 0f 0.5 to approximately center the variable on 0:

weight[i] <– ((occ[adj[start[i]]] + occ[adj[start[i]+1]] + occ[adj[start[i]+2]] + occ[adj[start[i]+3]] + occ[adj[start[i]+4]] + occ[adj[start[i]+5]] + occ[adj[start[i]+6]] + occ[adj[start[i]+7]] – ((8–num[i]) × occ[1]) – num[i]) / num[i]) – 0.5;

Appendix 2. Further details on the McMC algorithm to implement the autologistic model within a Bayesian Image Restoration framework

We take a Bayesian approach to the estimation of all unknowns, ν, which in the BIR case consist of the distribution model parameters, θ, the non-detection probabilities, q_i, and the underlying presence/absence status M_i of all grid cells with no recorded presence. Estimation of ν is based on the posterior distribution formed by combining prior knowledge about these unknowns with information provided by the data (Y) using Bayes theorem (see for example Brooks 2003):

$$f(\nu \mid Y) \propto f(Y \mid \nu) \times \pi(\nu),$$

where $\pi(\nu)$ denotes the prior knowledge about θ, and $f(Y \mid \nu)$ the likelihood of observing the data given the parameters. The function $f(\nu \mid Y)$ is of most interest, and denotes the joint posterior distribution of all the unknowns.
Markov chain Monte Carlo (MCMC) algorithms can be used to draw a set of values from the joint posterior distribution $f(\nu \mid Y)$ (Brooks 2003). The Metropolis-Hasting algorithm provides a general means by which samples can be generated from the posterior, by repeated proposal and rejection, even though it is known only up to a constant of proportionality. Gibbs sampling is a special case of the Metropolis-Hasting algorithm which enables samples to be drawn directly from the posterior, and is most commonly applied where a given marginal posterior distribution can be written in terms of a standard distributional form. These sets of values can be used to obtain summary statistics (such as the mean and standard deviation) of the posterior distribution of each individual parameter. In summary, drawing samples from the posterior distribution will proceed in three separate steps (these will be iterated many times): step 1: simulating values of the posterior distribution of the parameters θ, conditional upon a set of values for the states $\underline{M}$ and a set of values for q_i. Step 2: simulating values from the posterior distribution of q_i, conditional upon a set of states $\underline{M}$ and a set of parameter values θ (updated in step 1). Step 3: simulating values of the posterior distribution of the states $\underline{M}$, conditional upon a set of values for q_i (updated in step 2), and a set of parameter values θ (updated in step 1). We explain in detail what algorithms we used to simulate the draws from the posterior distribution of these unknowns below.

Step 1: Simulating from the posterior distribution of q_i
The detection probabilities may be estimated using the set of temporary true occupancy states M_i, and the set of recorded presences and absences O_i, if we assume that detection probabilities are independent across sites. Conditional on the underlying set of true occupancies the total number of observed occupancies follows a binomial distribution:

$$P(\Sigma O_i \mid \Sigma M_i, q_i) = \begin{pmatrix} \Sigma M_i \\ \Sigma O_i \end{pmatrix} \times (1 - q_i)^{\Sigma O_i} \times (q_i)^{\Sigma M_i - \Sigma O_i},$$

with q_i the probability of non-detection.

A flexible prior distribution on the probability of non-detection is a Beta distribution, with shape and scale parameters α and β:

$$P(q_i \mid \alpha, \beta) \propto \times (1 - q_i)^{\alpha - 1} \times q_i^{\beta - 1}$$

In this paper, we use a Beta(1,1) prior distribution for q_i (which is equal to a uniform distribution with values between 0 and 1) indicating that we believe that each value of this parameter between 0 and 1 is equally likely prior to fitting the model to the data. Using this prior specification, the posterior distribution of q_i is proportional to:

$$P(q_i \mid \Sigma O_i, \Sigma M_i) \propto (1 - q_i)^{\Sigma O_i + 1} \times (q_i)^{\Sigma M_i - \Sigma O_i + 1}$$

This is the standard form of a binomial distribution and therefore a Gibbs sampling step can be used to draw samples of the q_i conditional on the other unkowns.

Step 2: simulating values from the posterior distribution of θ
We used the Metropolis-Hastings algorithm to simulate draws from the posterior distributions of the parameters of the presence/absence model (Hastings 1970, Gilks et al. 1996). We used location invariant (uniform) prior distributions for all parameters of the presence/absence model, indicating that we believed that each value of these parameters equally likely prior to fitting the model to the data. Compute the likelihood, L_θ, of the parameters of the autologistic model, given the set of true underlying presences and absences M:

$$L_\theta = \prod_{i=1}^{N} \left| M_i - \frac{\exp\left(a_0 + \sum_{j=1}^{l} a_j X_i^j\right)}{1 + \exp\left(a_0 + \sum_{j=1}^{l} a_j X_i^j\right)} \right|,$$

Where N is the number of grid cells, a_o the intercept and a_j the slopes of the autologistic model (with covariates X_i^j, one of which is the spatial weight).

Then, sample a candidate set of parameter values, θ_{new}, from a symmetric jumping distribution. As a jumping distribution, we chose to use a multivariate normal distribution with mean $\vec{\theta}_j$ and an $n \times n$ variance-covariance matrix Σ: $\vec{\theta}_{new} \sim N\left(\vec{\theta}_j, \Sigma\right)$, with all off-diagonal elements set to zero. The choice of suitable values for the variances is important for the efficiency of the algorithm and requires tuning (but does not affect the final results). For a more detailed explanation of the choice of jumping distributions and the efficiency of Markov chain simulations see for example Gilks et al. (1996). Calculate the likelihood, $L_{\theta_{new}}$, of the parameters of the autologistic model, given the set of true underlying presences and absences M (using the same formula as above). Finally, values from the posterior distribution of θ are simulated as follows:

Accept the proposed parameter vector θ_{new} with probability $\min\left(1, \alpha\right)$, where $\alpha = \frac{L_{\theta_{new}}}{L_\theta}$, otherwise use the "old" set of parameters θ.

Step 3: simulating values from the posterior distribution of the states $\overline{M}$

The probability that a grid cell M_i is occupied by the species, given that the species was not recorded in this grid cell and the probability that it is present in the grid cell (computed using the autologistic model), are simulated by setting: $M_i = 1$ with probability

$$P\left(M_i = 1 \mid O_i = 0, \theta\right) = \frac{q_k}{q_k + P\left(M_i = 0 \mid \theta\right) / P\left(M_i = 1 \mid \theta\right)}$$ (this is a Gibbs sampling step).

Monitoring the convergence of the Markov chain

McMC algorithms have to be initialised with user-specified starting values, and a number of draws will be necessary before the Markov chains settle down to stationary behavior (the "burn-in period", which has to be discarded before the analysis of the results). Furthermore, users will have to determine how many iterations the McMC algorithm has to be run for, before it is safe to assume that the set of samples can be used to represent the posterior distribution. Many heuristic diagnostic and graphical techniques have been developed to assess this, but experience and a good understanding of McMC techniques remains crucial since in general no formal proof of convergence exists (Cowles and Carlin 1996, Gilks et al. 1996, Brooks and Roberts 1998).

References

Brooks, S. P. and Roberts, G. O. 1998. Convergence assessment techniques for Markov Chain Monte Carlo. – Stat. Computi. 8: 319–335.

Cowles, M. K. and Carlin, B. P. 1996. Markov chain Monte Carlo convergence diagnostics: a comparative review. – J. Am. Stat. Assoc. 91: 883–904.

Gilks, W. R. et al. (eds).1996. Markov Chain Monte Carlo in practice. – Chapman and Hall

Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. – Biometrika 67: 97–109.