Ecological Archives **M081-021-A4**

Philip J. Harrison, Ilkka Hanski, and Otso Ovaskainen. 2011. Bayesian state-space modeling of metapopulation dynamics in the Glanville fritillary butterfly. *Ecological Monographs* 81:581–598.

Appendix D. MCMC (Markov chain Monte Carlo) implementation.

To fully follow this Appendix the reader is advised to first read Appendix A and especially to consult Tables A1 and A2 and Figs. A1 and A2. MCMC works by simulating from the target distribution (in this case the joint posterior distribution of all the parameters and states in the model) by successively and repeatedly simulating from the conditional distribution of each model component (block of parameters) given the other components in the model. After discarding an initial number of draws (the burn-in) the remaining draws can be viewed as a sample from the posterior distribution. To update the parameters and states we used the Metropolis Hastings (MH) algorithm (Gelman 2004). Below we give detailed examples of the updating algorithms used, including the full conditional likelihoods, the proposal distributions we implemented and the adaptive algorithms we used during the burn-in period to tune the variance parameters of the proposals to gain a close to optimal acceptance probability of approximately 0.44 (Gelman 2004). In some cases we used blocking of highly correlated parameter values to improve the mixing of the MCMC chains (see below). One novel approach we implemented for updating the states in the model (in order to improve the mixing of the MCMC chains) was to utilize some ideas from particle filtering (also known as sequential importance sampling; SIS) into the MCMC algorithm as detailed below.

Mark-recapture model

MCMC was used to fit the mark-recapture model in which we used an adaptive adjustment of the multinormal proposal distributions for the log- and logit-transformed parameters (see Ovaskainen et al. 2008) during the burn in period.

IBM (updater 1):

The full conditional likelihood for the updater for the 16 main outer parameters in the DAG shown in Fig. A1 is

$$P(\phi, e_\mu, e_\sigma, n_h^0, t_\mu^*, t_\sigma^*, p_\mu, p_\sigma, q_\mu, q_r, q_e, q_k, D_1, D_2, m, k_\mu \,|\, \bullet) \propto$$
$$P(\phi, e_\mu, e_\sigma, n_h^0, t_\mu^*, t_\sigma^*, p_\mu, p_\sigma, q_\mu, q_r, q_e, q_k, D_1, D_2, m, k_\mu) \times$$
$$\prod_i \left\{ g_{i,1}^8 \prod_{t=1}^{t=15} \{g_{i,t}^3 g_{i,t}^4 g_{i,t}^5\} \prod_{t=2}^{t=16} \{g_{i,t}^7\} \right\} \prod_t \{h_t^1\} \prod_{t=1}^{t=15} \{h_t^3\} \prod_{i,t \in \Psi^N} \{h_{i,t}^4\} \prod_c \prod_y \prod_s \{h_{c,y,s}^5\}$$

where *t* gives the subscript for time such that *t*=1 refers to the butterfly "year" which starts in the autumn of 1993 and ends in the summer of 1994, and the final time point *t*=16 refers to the year that starts in the autumn of 2008 and ends with the final piece of spring nest count data in the spring of 2009. The subscript *i* gives the patch, of which there were 49 in the network #61 to which the model was fitted. For the control monitoring data part of the models the subscripts are *c* for patch, *y* for year and *s* for survey pair.

To improve the mixing of the MCMC chains we used a SCAM (single component adaptive Metropolis; Haario et al. 2005, Ovaskainen et al. 2008) algorithm for these 16 parameters. The parameters were first defined on the real scale (using either a log or a logit transformation if required) and candidate values were generated using a normal distribution with sampling directions

determined by rotating the 16-dimensional parameter space during the burn-in period by the eigenvectors of the covariance matrix of the posterior distribution sampled so far (Haario et al. 2005).

IBM (updater 2):

The second updater for the IBM merges ideas from MCMC and particle filtering such that to update the states we generate a state lineage associated with patch *i* in year *t* from the process model by first generating $M_{i,t-1}^{can}$ conditional on $F_{j,t-1}^{d-1}$ for all $j \neq i$ and then sequentially generating all the states for patch *i* up to $G_{i,t}^{can}$ (where *can* refers to the candidate value and $d-1$ to the previously accepted draw from the MCMC chain). Starting the generation at $M_{i,t-1}^{can}$ was done as an earlier exploratory analysis showed that when updating the states separately there were strong correlations between successive states. However, during the "movements and oviposition" sub-process section these correlations were broken down. The lineage of states is then updated as a block using the MH algorithm. In the first year the state lineage is generated starting from $N_{i,1}^{can}$, which is generated conditional on the prior for this state and $N_{i,1}^{o}$.

Given that we used the process model to generate the candidate values for the state lineages, most parts of the MH fraction associated with the likelihoods, priors and candidate generation cancel out. We then only need to consider the observation likelihoods for the current and candidate values and the final two process model likelihoods of the state lineages, such that the probability of accepting the candidate as the new state of the chain is $\min(1,\alpha)$, where:

$$\alpha = \begin{cases} \dfrac{f^1_{i,t}(N^o_{i,t}, N^{can}_{i,t}, p^{d-1}_{i,t}) g^5_{i,t}(Q^{d-1}_{i,t}, G^{can}_{i,t}, T^{d-1}_i) \prod_{j \neq i} g^4_{i,t}(M^{d-1}_{j,t}, F^{can}_{i,t}, R^{d-1}_{ji})}{f^1_{i,t}(N^o_{i,t}, N^{d-1}_{i,t}, p^{d-1}_{i,t}) g^5_{i,t}(Q^{d-1}_{i,t}, G^{d-1}_{i,t}, T^{d-1}_i) \prod_{j \neq i} g^4_{i,t}(M^{d-1}_{j,t}, F^{d-1}_{i,t}, R^{d-1}_{ji})} & \text{if } (i,t) \in \Psi^N \\ \dfrac{g^5_{i,t}(Q^{d-1}_{i,t}, G^{can}_{i,t}, T^{d-1}_i) \prod_{j \neq i} g^4_{i,t}(M^{d-1}_{j,t}, F^{can}_{i,t}, R^{d-1}_{ji})}{g^5_{i,t}(Q^{d-1}_{i,t}, G^{d-1}_{i,t}, T^{d-1}_i) \prod_{j \neq i} g^4_{i,t}(M^{d-1}_{j,t}, F^{d-1}_{i,t}, R^{d-1}_{ji})} & \text{if } (i,t) \notin \Psi^N \end{cases}$$

where Ψ^N refers to the group of all (i,t) combinations for which there are data for $N^o_{i,t}$ (see Appendix A for further details). The alternative fitting method to SSMs, particle filtering or SIS, works (in its basic form) by drawing a large number of "particles" from the joint distribution of prior parameters and initial states (Harrison et al. 2006). A particle in this context would be one possible realization for the entire metapopulation. This particle swarm is then projected forward one year using the state process model and each particle is given a weight according to that year's data. Particles are then resampled stochastically according to these weights. This process is iterated until the final year is reached. Hence, it can be seen that our updating algorithm for the states is to some extent merging the ideas from SIS with those from MCMC. However, we use SIS only for one part of the life-cycle, and only within one habitat patch in a turn. Starting from the number of females that visited the focal patch (either born there or immigrants), we used the process model to simulate the oviposition of eggs, their overwinter survival, and hatching of larvae. For this part we simply simulated the model forward, and accepted or rejected the "particle" based on its likelihood. If we would have used the same approach simultaneously for all model parameters (including all patches and all years), our MCMC algorithm would have become a conventional SIS algorithm. However, a straightforward SIS implementation for our model would have not been possible due to severe particle depletion. Almost all particles would have died out, as the likelihood of the data is zero whenever a generated value for $N^{can}_{i,t-1}$ is less than $N^o_{i,t-1}$ for any of the patch-year combinations.

IBM (updater 3):

Candidate generation for the random effect parameters (e_t, t_t^*, $p_{i,t}$, $p_{c,y,s}$ and ε_i) was done by defining the parameters on the real scale (using either a log or a logit transformation if required) and then using a normal distribution centered on the current value. The variances of these normal distributions were modified during the burn-in period to get an acceptance probability of approximately 0.44. The full conditional distribution for each e_t is given by

$$P(e_t|\bullet) \propto h_t^1 \prod_i \{g_{i,t}^1\} \prod_{\{i|(i,t)\in\Psi^S\}} \{g_{i,t}^2\}, \qquad \text{for } t = 1,...,16.$$

The full conditional distribution for each t_t^* is given by

$$P(t_t^*|\bullet) \propto h_t^3 \prod_i g_{i,t}^6, \qquad \text{for } t = 1,...,15.$$

The full conditional distribution for each $p_{i,t}$ is given by

$$P(p_{i,t}|\bullet) \propto h_{i,t}^4 f_{i,t}^1, \qquad \text{for } (i,t) \in \Psi^N$$

The full conditional distribution for each $p_{c,y,s}$ for the first set of control data is given by

$$P(p_{c,y,s}|\bullet) \propto h_{c,y,s}^5 f_{c,y,s}^3, \qquad \text{for } c = 1,...,4 \text{ and } s = 1,...,8 \text{ and } y = 1,...,3$$

The full conditional distribution for each $p_{c,y,s}$ for the second set of control data is given by

$$P(p_{c,4,1}|\bullet) \propto h_{c,4,1}^5 f_{c,4,1}^3, \qquad \text{for } c = 1,...,71.$$

The full conditional distribution for each ε_i is given by

$$P(\varepsilon_i \mid \bullet) \propto P(\varepsilon_i) g_{i,1}^{8} \prod_{t=1}^{t=15} \{g_{i,t}^{3}, g_{i,t}^{4}, g_{i,t}^{5}\}, \quad \text{for } i = 1,...,49.$$

SPOM (updater 1):

The full conditional likelihood for the updater for the 16 main outer parameters in the DAG shown in Fig. A2 is

$$P(\phi, e_\mu, e_\sigma, n_h^0, t_\mu^*, t_\sigma^*, z, \rho, q_\mu, q_r, q_e, q_k, D_1, D_2, m, k_\mu \mid \bullet) \propto$$
$$P(\phi, e_\mu, e_\sigma, n_h^0, t_\mu^*, t_\sigma^*, z\rho,, q_\mu, q_r, q_e, q_k, D_1, D_2, m, k_\mu) \times$$
$$\prod_{i,t \in \Psi^O} \{f_{i,t}^1\} \prod_c \prod_y \prod_s \{f_{c,y,s}^3\} \prod_i \prod_{t=1}^{t=15} \{g_{i,t}^1 h_t^1 h_t^3\}$$

Candidate generation for these 16 parameters were done in the same manner as was applied to updater 1 for the IBM, using the SCAM algorithm.

SPOM (updater 2):

State updating for the SPOM only needed to be performed when $O_{i,t}^o = 0$ or the data for patch *i* in year *t* was missing. The full conditional distribution for each occupancy state in the first year, $O_{i,1}$, is given by

$$P(O_{i,1} \mid \bullet) \propto \begin{cases} P(O_{i,1}) f_{i,1}^1 \prod_i g_{i,1}^1 & \text{if } (i,1) \in \Psi^O \\ P(O_{i,1}) \prod_i g_{i,1}^1 & \text{if } (i,1) \notin \Psi^O \end{cases} \qquad \text{for } i = 1,...,49.$$

where $P\left(O_{i,t}\right)$ is the prior for the initial states (i.e. with probability ½ $O_{i,1}=0$ and with probability ½ $O_{i,1}=1$). The full conditional distribution for the rest of the occupancy states, apart from those in the final year is given by

$$P(O_{i,t+1}|\bullet) \propto \begin{cases} f^1_{i,t+1} g^1_{i,t} \prod_i g^1_{i,t+1} & \text{if } (i,t+1) \in \Psi^O \\ g^1_{i,t} \prod_i g^1_{i,t+1} & \text{if } (i,t+1) \notin \Psi^O \end{cases} \qquad \text{for } i=1,...,49 \text{ and } t=1,...,14.$$

The full conditional distribution for the occupancy states in the final year is given by

$$P(O_{i,16}|\bullet) \propto \begin{matrix} f^1_{i,16} g^1_{i,15} & \text{if } (i,16) \in \Psi^O \\ g^1_{i,15} & \text{if } (i,16) \notin \Psi^O \end{matrix} \qquad \text{for } i=1,...,49.$$

We use individual MCMC updates for each $O_{i,t}$ such that with probability ½ $O^{can}_{i,t}=0$ and with probability ½ $O^{can}_{i,t}=1$. No updating was required when $O^{o}_{i,t}=O_{i,t}=1$.

SPOM (updater 3):

Candidate generation for random effect parameters (e_t, t^*_t and ε_i) was based on the same method as was used for the random effect parameters in updater 3 for the IBM. The full conditional distribution for each e_t is given by

$$P(e_t|\bullet) \propto h^1_t \prod_i \{g^1_{i,t}\}, \qquad \text{for } t=1,...,15.$$

The full conditional distribution for each t^*_t is given by

$$P(t^*_t|\bullet) \propto h^3_t \prod_i \{g^1_{i,t}\}, \qquad \text{for } t=1,...,15.$$

The full conditional distribution for each ε_i is given by

$$P(\varepsilon_i \mid \bullet) \propto P(\varepsilon_i) \prod_i \prod_{t=1}^{t=15} \{g_{i,t}^1\}$$

Note on our model development strategy:

When developing the MCMC algorithm for the IBM, one major difficulty was the often long computational time needed before an assessment could be made of whether or not the chains were converging fast enough for the method to be useful. Our general approach was to (i) first fit the model to a small simulated data set; (ii) then fit the model to simulated data of the same dimension as the real data; and finally (iii) fit the model to the real data. In most cases (i) was a very quick and efficient way to let us know if a particular method had a chance of succeeding. However, for many earlier incarnations of the models, specifically those without the added flexibility of the random effects models, the chains mixed adequately for (i) and (ii) but not for (iii). More generally, the most difficult aspects of developing MCMC algorithms for models with several unknown variables are the strong correlations in the high-dimensional posterior distribution (Newman et al. 2009). In our model there were very strong correlations, as one might expect, between successive states in the model and we thus found that the greatest improvement in the mixing of the chains came from updating subsets of the states (which we called state lineages above) together in a block. This worked well because we used in the burn-in phase SCAM with eigenvector rotation to identify the correlations among the main parameters and to account for them in the proposal distributions, and because we used in part of the procedure the process model as the proposal distribution. As a final note, we direct the interested reader to the paper by Newman et al. (2009) which gives a very thorough comparison of MCMC and SIS methods for fitting state-space models to animal

population data and how to approach the challenges faced when developing them for complex high-dimensional problems.

LITERATURE CITED

Gelman, A. 2004. Bayesian data analysis. Second edition. Chapman & Hall/CRC, Boca Raton, Fla. London.

Haario, H., E. Saksman, and J. Tamminen. 2005. Componentwise adaptation for high dimensional MCMC. Computational Statistics 20:265–273.

Harrison, P. J., S. T. Buckland, L. Thomas, R. Harris, P. P. Pomeroy, and J. Harwood. 2006. Incorporating movement into models of grey seal population dynamics. Journal of Animal Ecology 75:634–645.

Newman, K. B., C. Fernandez, L. Thomas, and S. T. Buckland. 2009. Monte carlo inference for state-space models of wild animal populations. Biometrics 65:572–583.

Ovaskainen, O., H. Rekola, E. Meyke, and E. Arjas. 2008. Bayesian methods for analyzing movements in heterogeneous landscapes from mark-recapture data. Ecology 89:542–554.