

Supplementary Material for
Species Density Models from Opportunistic Citizen Science Data
accepted to *Methods in Ecology and Evolution*, June, 2021

Jay M. Ver Hoef¹, Devin Johnson¹, Robyn Angliss¹, Matt Higham²

¹Marine Mammal Laboratory, NOAA-NMFS Alaska Fisheries Science Center, 7600 Sand Point Way NE, Seattle, WA 98115, jay.verhoef@noaa.gov

²Department of Statistics, St. Lawrence University, Canton, NY

APPENDIX: Details on MCMC Sampling Methods

Here, we provide more details on our MCMC sampling methods.

Updating Spatial Random Effects Sequentially

It would be most natural to set up the Metropolis-Hastings (MH) to sample spatial random effects sequentially. First consider sites with observed counts. The MH acceptance probability for z_i at a sampled location i is

$$\min \left(1, \frac{[y_i|\beta_0, z_i^*, \nu][\mathbf{y}_{-i}|\beta_0, \mathbf{z}_{-i}, \nu][z_i^*|\mathbf{z}_{-i}, \sigma, \rho][\mathbf{z}_{-i}|\sigma, \rho][\beta_0][\sigma][\rho]q(z_i|z_i^*)}{[y_i|\beta_0, z_i, \nu][\mathbf{y}_{-i}|\beta_0, \mathbf{z}_{-i}, \nu][z_i|\mathbf{z}_{-i}, \sigma, \rho][\mathbf{z}_{-i}|\sigma, \rho][\beta_0][\sigma][\rho]q(z_i^*|z_i)} \right),$$

where z_i^* is the proposed new value of z_i , \mathbf{z}_{-i} is the vector \mathbf{z} minus the i th element, and $q(z_i|z_i^*)$ is the proposal distribution, which, if symmetric, can be eliminated from the acceptance probability, and recall that $[\mathbf{y}|\beta_0, \mathbf{z}, \nu] = [y_i|\beta_0, z_i, \nu][\mathbf{y}_{-i}|\beta_0, \mathbf{z}_{-i}, \nu]$ because of conditional independence. Of course, many parts of the above equation cancel, and, assuming a symmetric proposal distribution, yields the acceptance ratio

$$\min \left(1, \frac{[y_i|\beta_0, z_i^*, \nu][z_i^*|\mathbf{z}_{-i}, \sigma, \rho]}{[y_i|\beta_0, z_i, \nu][z_i|\mathbf{z}_{-i}, \sigma, \rho]} \right).$$

Due to the special nature of the way we use the CAR model with row-standardization, $[z_i^*|\mathbf{z}_{-i}, \sigma, \rho] = N(\rho\bar{z}_i, \sigma^2/n_i)$, where \bar{z}_i is the mean of z -values that are neighbors of site i , and n_i is the number of neighbors. This comes directly from the definition of a CAR model. No inversion of matrices are required for one-at-a-time updating of z 's at sites with observed counts. For z 's at sites with missing count values, we can do Gibbs sampling straight from the definition. Again, this is simple, and the only limitation is that looping one-at-a-time might be slow, especially in an interpreted language like R.

Block Updating for Spatial Random Effects

We used block updating, which was faster than sequential sampling. We split the sites into those with observed counts \mathbf{z}_o , and those with missing counts \mathbf{z}_m . First consider sampling for \mathbf{z}_o . The MH acceptance probability is very similar to above,

$$\min \left(1, \frac{[\mathbf{y}|\beta_0, \mathbf{z}_o^*, \nu][\mathbf{z}_o^*|\mathbf{z}_m, \sigma, \rho][\mathbf{z}_m|\sigma, \rho][\beta_0][\sigma][\rho]q(\mathbf{z}_o|\mathbf{z}_o^*)}{[\mathbf{y}|\beta_0, \mathbf{z}_o, \nu][\mathbf{z}_o|\mathbf{z}_m, \sigma, \rho][\mathbf{z}_m|\sigma, \rho][\beta_0][\sigma][\rho]q(\mathbf{z}_o^*|\mathbf{z}_o)} \right), \quad (\text{A.1})$$

where \mathbf{z}_o^* is a proposal for sites with observed counts, $[\mathbf{y}|\beta_0, \mathbf{z}_o, \nu]$ is simply the product of the count distribution model for all sites with counts, and $q(\mathbf{z}_o|\mathbf{z}_o^*)$ is the proposal distribution. Assuming the proposal distribution is symmetric, simplifying the MH acceptance probability becomes

$$\min \left(1, \frac{[\mathbf{y}|\beta_0, \mathbf{z}_o^*, \nu][\mathbf{z}_o^*|\mathbf{z}_m, \sigma, \rho]}{[\mathbf{y}|\beta_0, \mathbf{z}_o, \nu][\mathbf{z}_o|\mathbf{z}_m, \sigma, \rho]} \right). \quad (\text{A.2})$$

It turns out that can we easily and rapidly draw an MH sample using Equation (A.2).

Note that

$$\frac{[\mathbf{y}|\beta_0, \mathbf{z}_o^*, \nu][\mathbf{z}_o^*|\mathbf{z}_m, \sigma, \rho]}{[\mathbf{y}|\beta_0, \mathbf{z}_o, \nu][\mathbf{z}_o|\mathbf{z}_m, \sigma, \rho]} = \frac{[\mathbf{y}|\beta_0, \mathbf{z}_o^*, \nu][\mathbf{z}_o^*, \mathbf{z}_m|\sigma, \rho]}{[\mathbf{y}|\beta_0, \mathbf{z}_o, \nu][\mathbf{z}_o, \mathbf{z}_m|\sigma, \rho]}.$$

So, for a block proposal, \mathbf{z}_o^* , we can form a MH probability acceptance ratio as

$$\min \left(1, \frac{[\mathbf{y}|\beta_0, \mathbf{z}_o^*] \exp \left(-\frac{1}{2}(\mathbf{z}_m, \mathbf{z}_o^*)\Sigma^{-1} \begin{pmatrix} \mathbf{z}_m \\ \mathbf{z}_o^* \end{pmatrix} \right)}{[\mathbf{y}|\beta_0, \mathbf{z}_o] \exp \left(-\frac{1}{2}(\mathbf{z}_m, \mathbf{z}_o)\Sigma^{-1} \begin{pmatrix} \mathbf{z}_m \\ \mathbf{z}_o \end{pmatrix} \right)} \right). \quad (\text{A.3})$$

Recall that Σ^{-1} is given directly to us in the CAR model (that is, this is what we are actually modeling), and it is sparse. Equation (A.3) contains a simple, fast matrix multiplication because we have the full, sparse Σ^{-1} , and allows for a block update of all \mathbf{z}_o . By the same logic, but without the count distribution, we can do block updates of \mathbf{z}_m as well,

$$\min \left(1, \frac{\exp \left(-\frac{1}{2}(\mathbf{z}_m^*, \mathbf{z}_o)\Sigma^{-1} \begin{pmatrix} \mathbf{z}_m^* \\ \mathbf{z}_o \end{pmatrix} \right)}{\exp \left(-\frac{1}{2}(\mathbf{z}_m, \mathbf{z}_o)\Sigma^{-1} \begin{pmatrix} \mathbf{z}_m \\ \mathbf{z}_o \end{pmatrix} \right)} \right) \quad (\text{A.4})$$

where \mathbf{z}_m^* is the proposal for \mathbf{z}_m .

The Problem of Many Zeros, Missing Data, and High Dispersion

We ran into a problem because there were many zeros, plus some very high counts, and missing values were unconstrained by observed counts. The \mathbf{z}_o for sites with observed counts will naturally be sampled so that

$$\exp(\beta_0 + z_i)$$

will be near the observed $y_i > 0$. However, for $y_i = 0$, the “fit” will get better and better as z_i goes toward negative infinity. This drives up the variance of the latent CAR model while the intercept goes down, and when we predict z_i at sites without missing counts, we can obtain large values. When they are exponentiated, they become even larger, which leads to unrealistically large values on the exponentiated scale. We considered several solutions, including constraining the CAR variance, but that had limited success because the individual z_i still became very large. Instead we treated \mathbf{z} as a truncated normal distribution to keep any value from becoming too large. So long as the truncation was not too limiting, it still allowed z_i to fit the 0’s very well. The truncation helped to stabilize both the mean and CAR variance parameter as well. A truncated multivariate normal distribution simply changes the normalizing constant, which cancels in Equations (A.3) and (A.4), so we only needed to make proposals that stayed within the truncation limits. We used uniform proposal distributions for each z_i , where the median of the uniform distribution was centered on z_i . If z_i was near a truncation value, so that the uniform distribution exceeded the truncation limit, then the uniform distribution was simply adjusted so that its endpoint was the truncation limit. The proposal distribution was symmetric, because both the current values and proposed values have the same probability density from the uniform distribution.

Using Sparse Matrices for Determinants, and Creating a Lookup Table

In the above sections, we showed how to sample \mathbf{z} quickly using block updates. Because there are only 2 samplings, one for sites with observed counts, and one for sites with missing counts, it scales linearly with sample size, as only sparse matrix multiplications are involved. Updates for β_0 and σ proceed as usual. They are very fast because no inverses or determinants are necessary. However, ρ presents a new challenge because it is involved in the determinant $|\mathbf{R}_\rho|$,

$$\log[\rho | \dots] \propto (1/2)\log(|\mathbf{R}_\rho^{-1}|) - \mathbf{z}'\mathbf{R}_\rho^{-1}\mathbf{z}/(2\sigma^2),$$

where $\mathbf{R}_\rho^{-1} = \mathbf{M}^{-1}(\mathbf{I} - \rho\mathbf{W})$ is sparse. Fortunately, determinants for sparse matrices are quite fast. For example, using the sparse matrix representation for a $11,424 \times 11,424$ CAR model, we were able to do 81 determinants in several minutes. These determinants can then be used as a lookup table when doing MCMC sampling. That is, rather than sampling from ρ continuously, we sample

from a grid of values. The MCMC chain will mix over the discrete values, so very little is lost. The determinant is known from the lookup table, and we may as well store the sparse matrices $\mathbf{R}_\rho^{-1} = \mathbf{M}^{-1}(\mathbf{I} - \rho\mathbf{W})$ for each ρ value as well, so they won't need to be computed during MCMC. There are some issues with creating the proposals. Our strategy was to take the nearest 3 indexes on each side of the index of the current ρ value. For example, we used a grid with 81 ρ values, ordered from lowest to smallest. They were evenly spaced on the logit scale, so values were compressed near 0 and 1 on the expit scale. Say the index for the current ρ value is 41. Then, we choose a proposal from indexes 38 to 40, and 42 to 44, with a discrete uniform probability of 1/6. As long as the current index is not near the ends, the proposal distribution will be symmetric. However, say the current value has index 81. Then, the proposal will come from indexes 78:80, with a discrete uniform probability of 1/3 for choosing an index. Suppose 78 is chosen. Then a proposal from index 78 would come from indexes 75 to 77 and 79 to 81, with a discrete uniform probability of 1/6 for choosing index 81. Hence, in general, we used Hastings and included the proposal probabilities $q(\rho|\rho^*)$ and $q(\rho^*|\rho)$,

$$\min \left(1, \frac{(|\mathbf{R}_{\rho^*}^{-1}|)^{1/2} \exp[-\mathbf{z}'\mathbf{R}_{\rho^*}^{-1}\mathbf{z}/(2\sigma^2)]q(\rho|\rho^*)}{(|\mathbf{R}_\rho^{-1}|)^{1/2} \exp[-\mathbf{z}'\mathbf{R}_\rho^{-1}\mathbf{z}/(2\sigma^2)]q(\rho^*|\rho)} \right)$$

An Overall Sampling Strategy

When block sampling \mathbf{z}_o and \mathbf{z}_m , it seems unlikely that, when each vector has 1000's of elements, we can create autocorrelation by independently drawing each of the elements of \mathbf{z}_o and \mathbf{z}_m . This does require literally millions of draws. So, a good strategy is to draw \mathbf{z}_o and \mathbf{z}_m many times before advancing to other parameters. Both of these draws are very fast, simply requiring the proposal, and a sparse matrix multiplication. We sampled both \mathbf{z}_o and \mathbf{z}_m twenty times before advancing to other parameters.