MCMC Sampler for N-Mixture Models

Angela Zhu
Oregon State University
zhuang@oregonstate.edu

Brayden Edwards
Oregon State University
edwabray@oregonstate.edu

Abstract

The goal of this work is to understand the mechanistic behavior of Markov Chain Monte Carlo (MCMC) sampling algorithms used to estimate species abundance from repeated surveys with imperfect detection. In this exercise we implemented five methods of MCMC sampling. We evaluate our approaches against a out of box implementation, NIMBLE, across a range of data conditions.

1 Introduction

N-mixture models are a probabilistic modeling tool used to estimate animal population sizes over spatial contexts. This class of models is within the class of occupancy models used to estimate occupancy of species using a hierarchical formulation. The core idea behind N-mixture models is that the target information (true occupancy) is something that we do not have access to, therefore we need to estimate this target using its relationship to variables we can observe (observed occupancy). In other words, we estimate true occupancy using imperfect observation data. The difference in N-mixture models compared to traditional occupancy modeling is that the target is no longer binary (presence or absence of species). It, instead, becomes how many individuals (0 to n) are at a location. The input data that these models have are counts of observed individuals in repeated visits to a set of locations. Since animals are rare to observe, ecologists use a hidden latent variable formation to model true abundance through an observed detection variable that is expected to be imperfect.

In this work, we aim to implement a custom MCMC sampler as an exercise in applied probabilistic graphical modeling. By developing a sampler from scratch, we develop a deeper understanding of how these hierarchical models behave in practice.

2 Related Work

N-mixture models are a class of hierarchical models that model observed data as a mixture of multiple underlying processes. This work is based off Royle's work Royle [2004] introducing N-mixture models for estimating population size based on repeated count data. We mostly used this work to build the definitions for our problem space since it does not go into the implementation details of building the MCMC sampler of the N-mixture model. We defined our experimental platform based off of sections 6.2 through 6.9 of the Applied Hierarchical Modeling in Ecology textbook by Kéry and Royle [2016].

To study population ecology, researchers must define a graphical model structure of random variables that represent the ecological process and observation process and input their data. Then they can use packages to estimate model parameters, common packages do this in two ways: using maximum likelihood estimation or Bayesian methods. unmarked Fiske and Chandler [2011] is a common framework in R that uses maximum likelihood estimation to estimate parameters. From the Bayesian side, NIMBLE uses MCMC sampling and allows for custom samplers. Once they are fitted, ecologists will compare different models to identify the best supported models. Then parameter estimates and uncertainties are extracted from the fitted models to draw ecological inferences. The general standard in ecology for accessing MCMC convergence is to use traceplots and the Gelman-Rubin diagnostic. Another thing to note here is that these methods will give the user posterior distributions and uncertainty scores for the information from the model, it is not just a point estimate.

3 Methodology

Defining the Graphical Model The study designed to estimate the population of a single species of animal across all survey locations. Animals are counted during survey occasions t = 1, 2, ..., T and at locations (sites) i = 1, 2, ..., R.
Definitions:

- N_i is the number of individuals available for counting at location i, **True Abundance**
- $C_{i,t}$ is the number of distinct individuals counted at location i in time t, **Observed Data**
- λ is the expected abundance
- p is the detection probability

The hierarchical relationships of these variables are represented in a plate diagram in Figure 1.

What is MCMC? Markov chain Monte Carlo (MCMC) methods are a form of approximate inference through sampling.

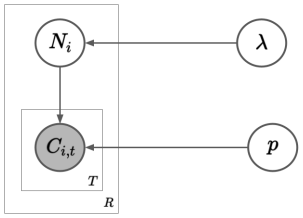


Figure 1: **Plate Diagram for Static p and lambda values**

MCMC is unique from other sampling methods in that samples are generated by defining a random process to transition from the current state to the next state, where a state corresponds to the values in that sample. MCMC is an algorithm that is used to compute the posterior distribution of some set of variables, allowing us to model the relationship between observations and our target variable. In our case, we are attempting to model the posterior on N:

$$P(N|C,\lambda,p) \propto \prod_{i=1}^{R}\left(\prod_{t=1}^{T} P(C_{it}|N_i,p)\right)P(N_i|\lambda) \quad (1)$$

Within the class of MCMC, we use a mixture of both Metropolis and Metropolis-Hastings (MH) sampling framework for our various methods. What makes MH different from other MCMC methods is that it relies on computing the acceptances score of the new state given the current state. If the score is large enough, the new state is accepted, otherwise, we reject the new sample. MH offers much flexibility because all it requires is a target distribution (the joint distribution in our case) and a proposal distribution (how we define the transition from one state to the next). With only these two pieces, we are able to perform approximate inference; this is the true power of MH methods. However, MH is slow to converge to the posterior, often sensitive to the proposal distribution, and has high autocorrelation between samples when not performing independent MCMC sampling.

3.0.1 Acceptance Probability

The acceptance probability is a key part of the MH algorithm. The general formula to compute the acceptance probability is defined below:

$$\alpha = \min\left(1, \frac{\pi(x')\mathcal{T}(x' \to x)}{\pi(x)\mathcal{T}(x \to x')}\right) \quad (2)$$

where variable x' is the proposed new state and x is the current state. A proposed sample is accepted if $U \sim \text{Uniform}(0,1) \leq \alpha$. Along with this general formula, there are two special cases of the MH algorithms that our methods leverage: the Metropolis algorithm and the independent sampler.

Defining $\pi(x)$ $\pi(x)$ is used in the acceptance probability and is simply our joint probability distribution. In our case, our joint probability is:

$$\pi(x) = P(C|N,p)P(N|\lambda)P(\lambda)P(p) \quad (3)$$

$$= \left[\prod_{i=1}^{R} P(C_i|N_i,p)P(N_i|\lambda)\right]P(\lambda)P(p) \quad (4)$$

$$= \left[\prod_{i=1}^{R}\left(\prod_{t=1}^{T} P(C_{i,t}|N_i,p)\right)P(N_i|\lambda)\right]P(\lambda)P(p) \quad (5)$$

Metropolis Algorithm The Metropolis algorithm is a special case of the MH algorithm where the proposal distribution is symmetric, meaning that the probability of transitioning from state x to x' and vise versa is equal. As a result, our acceptance probability simplifies to just the ratio of the joint distributions:

$$\alpha = \min\left(1, \frac{\pi(x')\mathcal{T}(x' \to x)}{\pi(x)\mathcal{T}(x \to x')}\right) = \min\left(1, \frac{\pi(x')}{\pi(x)}\right) \quad (6)$$

The most simple proposal distribution is the uniform distribution, which is symmetric and falls under the Metropolis algorithm. This is also the case for the normal distribution, both of which are used in our various methods.

Independent Sampler Another special case is that of the independence sampler, which means that the proposed sample only depends on the new state such that $q(x' \mid x) = q(x')$, causing the acceptance probability to simply as:

$$\alpha = \min\left(1, \frac{\pi(x')\mathcal{T}(x' \to x)}{\pi(x)\mathcal{T}(x \to x')}\right) \quad (7)$$

$$= \min\left(1, \frac{\pi(x')q(x \mid x')}{\pi(x)q(x' \mid x)}\right) \quad (8)$$

$$= \min\left(1, \frac{\pi(x')q(x)}{\pi(x)q(x')}\right) \quad (9)$$

Our Method 1 utilizes both the Metropolis algorithm and independence sampling, while Method 2 only operates under the independence sampling framework. All three random walk methods (Methods 3, 4, and 5) follow a more standard MH approach.

3.1 Model Assumptions

Our fist major assumption is population closure. The population closure assumption means that the data-generation process assumed by the model has no change in site population at any visit. In other words, at a single site we assume that the true population is the same at the first visit and the 10th visit; this assumption is ubiquitous in ecological modeling because it allows us to simplify the relationships within the data. Without this assumption, we would need to model births/deaths and the traveling of species in and out of the site, which is beyond the scope of this project.

Another simplifying assumption we hold to in this work is that there are no false positives or duplicate sightings. In other words, if a visit reports a count of three, then we can assume that three unique individuals were truly spotted at that site during the visit.

3.2 Datasets

Simulated Data. The data generation process is based off examples in section 6.2 titled "An Exercise in Hierarchical modeling: Derivation of Binomial N-mixture Models from First Principles" Kéry and Royle [2016]. For each site i, the true abundance N_i is drawn from a Poisson distribution with parameter λ. Then we add repeated surveys for each site and each timestep, as an observation $C_{i,t}$ which is drawn from a Binomial distribution with N_i trials and detection probability p. The full parameter grid across five site counts i (5, 10, 20, 40, 100), four visits counts t (2,6,12,20), four expected abundance values λ (2, 5, 15, 30), and four detection probabilities p (0.1, 0.25, 0.5, 0.75). Most of our experiments are run on simulated data, but we do test against one real animal dataset in the experiments section 4.2.

3.3 Methods: Markov Chain Monte Carlo Samplers

Many of our model assumptions, as they relate to our choice of parameters, were tested through ablation testing. These ablation tests, however, still required us to hold certain variables constant, which we refer to as our baseline parameters. These parameters are p, λ, R, and T, set at 0.25, 5, 20, and 6, respectively. We believe these baselines to be reasonable for a given real-world scenario. To minimize error, we fit all models using the same datasets that we generate in Python and save as CSVs. To reduce variance, we used a seed of 42 in Python and 123 in R.

3.3.1 Baseline: NIMBLE

To benchmark our custom MCMC sampler, we fitted an N-mixture model using NIMBLE [de Valpine et al., 2024], a well established R package for Bayesian hierarchical models. This general purpose package lets the user to define a state space model and then conduct inference on datasets. The model specifies the ecological process and observation process as follows:

$$N_i \sim \text{Poisson}(\lambda) \tag{10}$$

$$C_{i,t} \sim \text{Binomial}(N_i, p) \tag{11}$$

We assume that the average abundance is constant across all sites and that the detection probability is constant across all sites and visits. We specified normal priors for λ and p, The model would estimate average abundance λ, and detection probability p, which we then use to calculate total abundance across all sites, N. The model ran three MCMC chains for 40,000 iterations each, with a burn-in count of 10,000 iterations.

We do not specify any samplers, relying instead on the use the package's default. In our case, NIMBLE used Random-walk (RW) samplers using Metropolis-Hastings MCMC for estimating λ and p since they were defined to have normal priors, similar to our method 3.3.4. Then for each N_i, where i is from 1 to R sites, NIMBLE used a slice sampler. The sampler iteratively updates each parameter using its assigned sampler, proposing new values for each parameter. Accepted values then get saved to approximate the posterior distribution of variables of interest. The package translates the model from R into C++ code to speed up compilation time.

3.3.2 Method 1: Uniform Proposal

Method 1 is an independent sampler, following the Metropolis algorithm (see section 3.0.1) with the proposal distribution defined as follows:

$$p \sim \text{Uniform}(0, 1) \tag{12}$$

$$\lambda \sim \text{Uniform}(1, S) \tag{13}$$

$$N_i \sim \text{Uniform}(1, S) \tag{14}$$

where S is a tunable parameter. This is our most basic model. Under the uniform proposal, this independent sampler's acceptance probability takes on the same general form as (6) and is defined as follows:

$$\alpha = \min\left(1, \frac{\pi(x')}{\pi(x)}\right) \tag{15}$$

Because the uniform proposal distribution is symmetric and independent between states, the $P(p) = P(p')$, $P(\lambda) = P(\lambda')$, and $P(N_i) = P(N_i')$ causing these terms to cancel, resulting in an acceptance probability that is simply the ratio of the joint probability of the proposed and current state.

3.3.3 Method 2: Uniform Priors

Method 2 is an independent sampler, similar to Method 1, where p and λ are generated in the same way as (12) and (13), respectively. However, Method 2 samples N_i differently than Method 1; namely, Method 2 samples N_i from a Poisson distribution with mean λ:

$$N_i \sim \text{Poisson}(\lambda) \tag{16}$$

Method 2 capitalizes on the fact that abundance counts are often modeled using a Poisson distribution in ecology, and our synthetic data was generated by sampling N_i from a Poisson with mean λ. Additionally, this method was motivated by the idea that we could limit the search space by avoiding uniform proposals for all parameters.

Lastly, because we are no longer sampling all variables uniformly and Poisson is not a symmetric distribution, we must incorporate the probability of N_i back into our acceptance probability score. Under the uniform and non-symmetric Poisson proposal, this independent sampler's acceptance probability takes on the same general form as (9) and is defined as follows:

$$\alpha = \min\left(1, \frac{\pi(x')q(x)}{\pi(x)q(x')}\right) \tag{17}$$

$$= \min\left(1, \frac{\pi(x')P(N|\lambda)P(\lambda)P(p)}{\pi(x)P(N'|\lambda')P(\lambda')P(p')}\right) \tag{18}$$

$$= \min\left(1, \frac{\pi(x')\prod_i^R P(N_i|\lambda)P(\lambda)P(p)}{\pi(x)\prod_i^R P(N_i'|\lambda')P(\lambda')P(p')}\right) \tag{19}$$

$$= \min\left(1, \frac{\pi(x')\prod_i^R P(N_i|\lambda)}{\pi(x)\prod_i^R P(N_i'|\lambda')}\right) \tag{20}$$

which is the a special case of MH called an independent sampler outlined in section 3.0.1.

3.3.4 Method 3: Random Walk

Unlike the previous two methods, Method 3 moves away from entirely symmetric or independence sampling towards a more standard MCMC approach. Specifically, Method 3 deploys a random walk method over each parameter λ, p, and N_i. In this method, λ and p are both drawn from a normal distribution with mean being the previous state for each variable with some small variance. N_i is then drawn from a Poisson distribution with mean being the previous state of N_i. See the equations below:

$$p' \sim \mathcal{N}(p, \sigma_p^2) \tag{21}$$

$$\lambda' \sim \mathcal{N}(\lambda, \sigma_\lambda^2) \tag{22}$$

$$N_i' \sim \text{Poisson}(N_i) \tag{23}$$

The intuition behind this method is to explore the space more slowly by setting σ^2 to be some small, tunable value. Thus, this method explores the Markov chain in a more systematic way, taking smaller steps, which is much different from the shotgun-like approach of uniform sampling in Methods 1 and 2.

Because the normal distribution is symmetric, in that if $x \sim \mathcal{N}(x', \sigma^2)$ and $x' \sim \mathcal{N}(x, \sigma^2)$ then $P(x|x') = P(x'|x)$, the acceptance probability resembles Method 2's, but no longer are the two states independent, resulting in the following formulation:

$$\alpha = \min\left(1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right) \tag{24}$$

$$= \min\left(1, \frac{\pi(x')P(N|N_i')P(\lambda)P(p)}{\pi(x)P(N'|N_i)P(\lambda')P(p')}\right) \tag{25}$$

$$= \min\left(1, \frac{\pi(x')\prod_i^R P(N_i|N_i')P(\lambda)P(p)}{\pi(x)\prod_i^R P(N_i'|N_i)P(\lambda')P(p')}\right) \tag{26}$$

$$= \min\left(1, \frac{\pi(x')\prod_i^R P(N_i|N_i')\mathcal{N}(\lambda|\lambda', \sigma_\lambda^2)\mathcal{N}(p|p', \sigma_p^2)}{\pi(x)\prod_i^R P(N_i'|N_i)\mathcal{N}(\lambda'|\lambda, \sigma_\lambda^2)\mathcal{N}(p'|p, \sigma_p^2)}\right) \tag{27}$$

$$= \min\left(1, \frac{\pi(x')\prod_i^R P(N_i|N_i')}{\pi(x)\prod_i^R P(N_i' \mid N_i)}\right) \tag{28}$$

Biased Acceptance Probability One key characteristic of this method is that it was intentionally left biased in its computation of the acceptance probability. Namely, the normal distribution is such that negative values can be produced from the draw. However, we cannot allow p or λ to be negative. Additionally, N_i can produce values lower than our counts. To counter this, we continue to draw the proposal from the normal and Poisson distribution until the value is within the allowed range (i.e., non-negative for λ and p and $N_i \geq C_{i,\max}$). In doing so, we should normalize the probability of obtaining the proposed sample by the probability of obtaining an in-domain sample. However, for this method we do not do this, and simply compute the probability without the normalization.

3.3.5 Method 4: Random Walk with Component-wise Updates

Method 4 is nearly identical to Method 3, except that we update the parameters component-wise. In other words, instead of proposing all new variables and then computing the acceptance probability, we propose and compute acceptance probabilities one variable at a time. If the proposal is accepted, it becomes part of the new state. If no acceptance is reached after n number of proposals, we retain the old state and then iterate on this process for the rest of the variables, starting with λ, then p, and then $N_i \; \forall i = 1, ..., R$. Note that n can be set to any value (we set $n = 5$) and acts as an early stopping mechanism.

3.3.6 Method 5: Random Walk with Truncated Normals + Unbiased Transitions

Method 5 follows the same methodology as Method 3, utilizes a truncated normal distribution instead of the repeated draws to ensure proper bounding. Additionally, the acceptance probabilities are properly computed by correctly normalizing for each variable. Specifically, the acceptance score is computed as follows:

$$\alpha = \min\left(1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right) \tag{29}$$

$$= \min\left(1, \frac{\pi(x')P(N|N_i', N \geq C_{i,\max})P(\lambda)P(p)}{\pi(x)P(N'|N_i, N \geq C_{i,\max})P(\lambda')P(p')}\right) \tag{30}$$

$$= \min\left(1, \frac{\pi(x')\prod_i^R \frac{P(N_i|N_i')}{P(N_i \geq C_{i,\max})}P(\lambda)P(p)}{\pi(x)\prod_i^R \frac{P(N_i'|N_i)}{P(N_i' \geq C_{i,\max})}P(\lambda')P(p')}\right) \tag{31}$$

$$= \min\left(1, \frac{\pi(x')\prod_i^R \frac{P(N_i|N_i')}{P(N_i \geq C_{i,\max})}\psi(\lambda|\lambda', \sigma_\lambda^2)\psi(p|p', \sigma_p^2)}{\pi(x)\prod_i^R \frac{P(N_i'|N_i)}{P(N_i' \geq C_{i,\max})}\psi(\lambda'|\lambda, \sigma_\lambda^2)\psi(p'|p, \sigma_p^2)}\right) \tag{32}$$

$$= \min\left(1, \frac{\pi(x')\prod_i^R \frac{P(N_i|N_i')}{P(N_i \geq C_{i,\max})}\psi(\lambda|\lambda', \sigma_\lambda^2)\psi(p|p', \sigma_p^2)}{\pi(x)\prod_i^R \frac{P(N_i'|N_i)}{P(N_i' \geq C_{i,\max})}\psi(\lambda'|\lambda, \sigma_\lambda^2)\psi(p'|p, \sigma_p^2)}\right) \tag{33}$$

where $\psi(.)$ is the truncated normal distribution. As you can see, unlike Method 3 and 4, we are properly incorporating the truncation into the acceptance score calculations. As a result, Method 5 is an unbiased version of Method 3, in that the acceptance probabilities are being correctly computed.

Implementation The code for this project is stored on GitHub.

4 Results

4.1 Experiments on Simulated Data

We first tested our model on our synthetic dataset (see section 3.2 for more information on our simulated data) with baseline parameters of $R = 20$, $T = 6$, $\lambda = 5$, and $p = 0.25$. Estimation accuracy is measured using absolute error (AE), defined as $|\hat{\theta} - \theta|$. We compare methods based on AE in estimating total population abundance, as well as AE for λ and p.

Table 1 summarizes the results of NIMBLE and all five of our methods on our synthetic dataset. Method 4 obtains superior results in terms

Table 1: Results on Simulated Data

	Estimated Values		
Method	λ	p	Population
NIMBLE	6.96	**0.22**	139.72
1	7.74	0.20	154.42
2	10.48	0.13	213.23
3	6.66	0.20	138.12
4	**3.94**	0.38	**77.80**
5	8.97	0.15	179.64
Ground Truth	**5.00**	**0.25**	**107.00**

of AE in population abundance (AE=29.2), followed by Method 3 (AE=31.12) and NIMBLE (AE=32.72). Additionally, Method 4 was able to achieve the smallest absolute error in estimating λ (AE=1.06) over the other models, while NIMBLE had the closest estimate of p (AE=0.03).

Simulated Data Results Discussion Overall, we anticipated Method 4 to produce better results than the other methods. However, we did not anticipate that Method 3 would perform better than Method 5. As noted in the discussion on these models, Method 3 is biased in that we are not properly normalizing after truncation, unlike Method 5. We believe the discrepancy in our expectations and the experiment results may be due to how the model is being biased. When we normalize by $P(N_i \geq C_{i,\max})$, we are automatically accepting more proposals. As a result, we expect Method 5 to be accepting more proposals than Method 3, in general. However, our results indicate that the difference in number of accepted samples between the two methods is minuscule. This is likely due to the changes in N between the current and proposed sample being negligible, resulting in the two normalization terms canceling. See equation 33 and consider $P(N_i \geq C_{i,\max}) \approx P(N_i' \geq C_{i,\max})$. We believe this is why our model is not actually accepting noticeably more samples as the equation would suggest.

Additionally, including this normalization term should cause our model to bias towards exploring smaller values of N, but this intuition is not clear from our results. While the equation would suggest that Method 3 is biased towards larger N values, it is difficult to quantify this as the estimates are coupled with C, p, and λ in a manner difficult to sift through for assessing local bias on a given N_i.

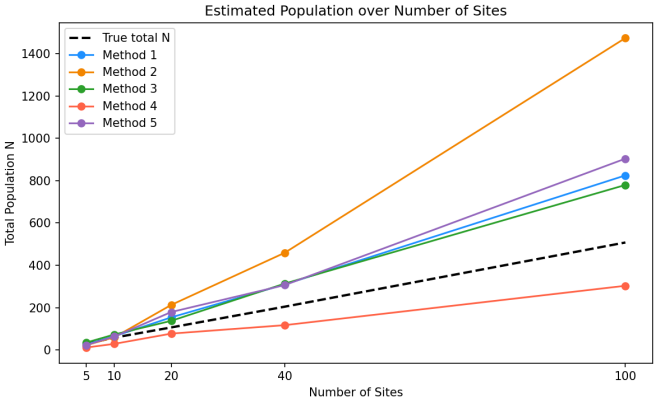
Moreover, we note that low AE estimation of λ is a proxy for low AE estimation in total population. This trend also emerged in our ablation tests over λ; the better the model estimated λ, the better we did at estimating total abundance. Estimations with low AE of p, however, did not directly correlate with low AE of total population.

Finally, we note the consistent tradeoff between estimations of λ and p. You can see from our results that our models were consistent in the following behavior: over-estimating λ resulted in under-estimating p or vise versa. This behavior was prevalent throughout all experiments. This is a known limitation from the model, wherein estimates in abundance, controlled by λ, and p, compete to find the correct combination that best explains the observed counts of C.

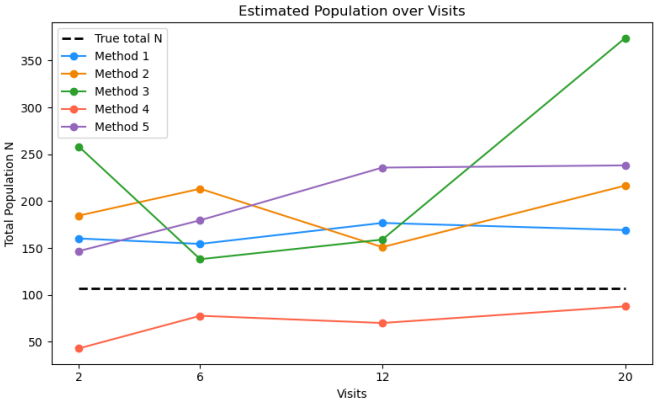
4.1.1 State Changes

Changing Number of visits vs sites This scenario is particularly interesting for ecological research because the number of sites surveyed R and the number of visits to each site T are factors that researchers can control when designing a study. In this experiment, we set the baseline parameters of expected abundance per site λ to 5, and detection probability p to 0.25, while varying R and T. These varied conditions can be seen as changes in the observation data C that we input into the model.

As we increase the number of sites surveyed, the total population size also increases, because λ remains constant. We find that the absolute error (AE) in all estimates increases with the number of sites. At 5 sites, Method 1 has the lowest AE of 4.6 individuals, which is -12.00% from the true population N of 30. At 100 sites, Method 4 has the lowest AE of 204.4 individuals, which is -40.23% from the true N of 508. The increase in relative error could be due to the increased



(a) Population as number of sites increases



(b) Population as number of visits increases

Figure 2: **Estimating population under different survey design choices.** Both figures use baseline parameters for $\lambda = 5$, and $p = 0.25$. The varied parameter for Figure 2a is the number of sites R = 5, 10, 20, 40, 100. Showing the effect of increasing the number of survey sites on each method's estimate of total population. Similarly, Figure 2b varies the number of visits T = 2, 6, 12, 20. Showing the effect on estimation population size as we increase the number of visits to a site. The true total population is plotted as the black dotted line for both plots. The distance from this line represents how far each model's estimate is from the true value. Lines under the dotted line represent underestimates of total population and lines above are overestimates.

number of parameters to estimate. This may be due to the cumulative effect of the estimation error across more sites. This suggests that increasing the number of sites may not be sufficient to improve the accuracy of population estimates.

We hypothesized that as we increase the number of visits to each site, the population estimates would become more accurate. The rationale behind this is that repeatedly visiting a site should reveal more information about the true abundance N_i at that location and make it easier to estimate the detection probability p, assuming these parameters remain constant throughout our study. We expect that as the number of visits to a site T go to infinity, the estimated population count would converge to the true value.

This hypothesis holds true for Method 4, which is also our best performing method in this scenario. As we increase the number of visits from 2 to 20, the estimated population approaches the true total population N. However, the other models do not follow this trend. For Methods 1, 2 and 5 the estimations do not substantially change the AE as we add more visits to each site. Method 3 exhibits unexpected behavior, with the AE initially decreasing when visits increase from 2 to 6, then increases significantly as visits increase to 20.

4.2 Additional Element: Experimenting on Real Data

We chose to use mallard data from Kéry et al. [2005] that is available via the unmarked library in R. The dataset consists of 239 sites with 3 repeated visits. This data was collected by the Swiss national monitoring program on common breeding bird species in Switzerland in 2002. As common with ecological data, there is missing data, such as missing some repeat visits and some sites that don't have data at all. For our purposes, we removed all sites with any missing data, leaving us with 191 locations with 3 repeat visits each. It is worth noting that this approach may limit the generalizability of our model, as ecological data are characteristically sparse and prone to missing data due to survey logistics.

Additionally, this real world dataset exhibits substantial zeros; the probability that a given site visit leads to no mallard observation is 85.69%. While this may seem extreme, low per-survey detection probabilities are characteristic of ecological count data. This is also the main motivation to use N-mixture models, that explicitly separate the latent abundance process form the imperfect observation process[Royle, 2004].

Real Data Results Discussion On the real-world datasets, our models performed significantly worse than NIMBLE; see the results in Table 2. This result was expected as NIMBLE is a geneal purpose approximator, while our Methods were designed with very specific modeling assumptions. Namely, our model assumed assumed a global λ and p across all sites, making our parameter estimation highly susceptible to outliers in the data.

Overall, Method 1 performed the best on the real-world dataset when comparing to the NIMBLE results. This was anticipated behavior because Method 1 consistently provided lower AE in total abundance on sparse data across our ablation tests. When running ablation tests, we found that Method 1 estimated total abundance with an $AE = 1$ while the next closest method, Method 3, produced an $AE = 17.34$ with true abundance of 40. Our intuition for why Method 1 works better under sparse data conditions lies in the fact that the vast sweeps in the parameter space, because of the Uniform distributions, allows the sampler to better explore the distributions compared with our other methods. Also, it may be that the random walk based methods get stuck in low density, high volume regions and struggle to explore higher density areas of the posterior distribution.

Table 2: Results on Mallard Dataset

	Estimated Values		
Method	λ	p	Population
NIMBLE	**0.35**	**0.62**	**67.23**
1	2.50	0.10	473.0
2	5.34	0.04	983.09
3	5.77	0.30	948.0
4	3.80	0.06	720.46
5	4.80	0.37	907.09

The real-world data was highly sparse, containing 85.69% of zeros in the entries. While we did test various levels of sparsity in our synthetic data, such extreme conditions made it difficult for our model obtain enough informative samples to explore the space.

To estimate how sensitive our methods were to initialization, we performed an additional experiment on Method 1. By simply decreasing the upper bound of the Uniform distribution over λ (parameter S in equation (13)), we obtained results from Method 1 that were nearly five times closer to NIMBLE. While these results are not included in our table due to data leakage, they serve as a useful case study how sensitive the MH algorithm is to hyperparameter choice.

5 Conclusions

5.1 How do our experiments connect with real life ecology

Our experiments show that our methods perform better as observation data becomes less sparse. This is not surprising from a statistical perspective, since with more observed detections our models receive a stronger signal about the detection probability and true abundance. And when the data is sparse, meaning our observed counts are dominated by zeros, the models struggle to determine whether the site has low true abundance or whether the individuals are present but go undetected. This issue reveals the tensions for practitioners. Models do better with more data, but collecting more structured survey data comes at a substantial cost and is often done manually. Depending on the species, in real ecological surveys detection probabilities are likely lower than our baseline of 0.25. This is especially true for species with low population counts, often species of conservation interest. In the occupancy framework, a detection is a much stronger signal than a non-detection since a negative could be true absence or a false negative, assuming no false positives. Citizen science platforms like eBird or iNaturalist do offer the possibility of many repeat visits to a location over time, but these visits are not structure the way formal surveys are to reduce bias and control or at least document survey conditions. This makes ecologists hesitant to use citizen science data, since without bias reduction the data may reflect where species are most likely to be found rather than true occupancy Guillera-Arroita et al. [2015]. This highlights the necessity to see ecological modeling as a structured model whose components of abundance and detection as separate variables that both contribute to an observation.

5.2 Bayesian/Frequentest Methods

One key consideration when starting this project was in modeling using Frequentest or Bayesian methods. MH utilizes Bayesian methods for inference, allowing us to create priors on the parameters of our distributions, treating them as random variables themselves and modeling them with some uncertainty. While we did not take full

advantage of such methods in this work, we did attempt to implement a Gamma and Beta prior on λ and p, respectively. The results we obtained were not great, however, and we lacked the time to dive into this deeper. Nevertheless, modeling parameters using prior distributions has precedence in the ecology space and is a natural next step to this work, along with comparing our current implementation to a Frequentest method, like MLE.

5.3 Limitations with MCMC

We also note some key limitations of MCMC that we experienced throughout this project. MCMC is sensitive to hyperparameters. As we discussed in section 4.2, our Method 1 model was able to achieve five times better results on the real-world dataset simply by changing one hyperparameter in the model. We also found through experimentation that different initializations of our parameters caused the sampler to behave differently. Overall, the amount of hand-tuning needed in MH, including selecting the proposal distributions, parameters on the proposal distributions, number of iterations, burn-in period, etc., makes MH difficult to rely on; MH is much more about fine-tuning than algorithmic stability.

5.4 Future Directions

There are two natural extensions of this work. The first is incorporating covariates, which is standard practice in ecological occupancy modeling. In this framework, state covariates are environmental or site level variables used to predict the expected abundance, λ and detection covariates capture conditions that vary across sites and visits that can be used to predicted the detection probability p. Introducing covariates allows each site to have its own expected abundance λ_i and a detection probability, $p_{i,j}$ at every site and visit. This would enable the model to capture more complex ecological relationships. The usual set up is that we introduce a vector of weights for the state model $\beta = [\beta_0, \beta_1, ..., \beta_n]$ is the vector of weights state model and a vector of state/occupancy covariates $x = [x_1, ..., x_n]$. Such that $log(\lambda_i) = \beta_0 + \beta_1 * x_1 + ... + \beta_n * x_n$. Similarly for the detection model, $logit(p_{i,t}) = \alpha_0\alpha_1 * w_1 + ... + \alpha_n * w_n$ where alpha is the vector of weights for the detection model $\alpha = [\alpha_0, \alpha_1, ..., \alpha_n]$ and w is the vector of detection covariates $w = [w_1, ..., w_n]$.

The second extension is investigating which MCMC sampling strategies are most effective for ecological occupancy models under sparse data conditions. We could compare samplers we learned in class like Hamiltonian MCMC or using an adaptive MCMC that tunes the step size based on the acceptance rate. It would be interesting to see what samplers work best for realistic sparsity levels.

References

Perry de Valpine, Christopher Paciorek, Daniel Turek, Nick Michaud, Clifford Anderson-Bergman, Fritz Obermeyer, Claudia Wehrhahn Cortes, Abel Rodríguez, Duncan Temple Lang, and Sally Paganin. *NIMBLE: MCMC, Particle Filtering, and Programmable Hierarchical Modeling*, 2024. URL `https://cran.r-project.org/package=nimble`. R package version 1.3.0.

Ian Fiske and Richard Chandler. unmarked: An R package for fitting hierarchical models of wildlife occurrence and abundance. *Journal of Statistical Software*, 43(10):1–23, 2011. URL `https://www.jstatsoft.org/v43/i10/`.

Gurutzeta Guillera-Arroita, José J Lahoz-Monfort, Jane Elith, Ascelin Gordon, Heini Kujala, Pia E Lentini, Michael A McCarthy, Reid Tingley, and Brendan A Wintle. Is my species distribution model fit for purpose? matching data and models to applications. *Global ecology and biogeography*, 24(3):276–292, 2015.

Marc Kéry and J. Andrew Royle. *Applied Hierarchical Modeling in Ecology: Modeling Distribution, Abundance and Species Richness Using R and BUGS*, volume 1. Academic Press, 2016.

Marc Kéry, J. Andrew Royle, and Hans Schmid. Modeling avian abundance from replicated counts using binomial mixture models. *Ecological Applications*, 15(4):1450–1461, 2005. doi: https://doi.org/10.1890/04-1120. URL `https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1890/04-1120`.

J. Andrew Royle. N-mixture models for estimating population size from spatially replicated counts. *Biometrics*, 60(1):108–115, 2004. doi: https://doi.org/10.1111/j.0006-341X.2004.00142.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0006-341X.2004.00142.x`.