6-22-2022 1:30 PM

The Analysis of Mark-recapture Data with Individual Heterogeneity via the H-likelihood

Han-na Kim, *The University of Western Ontario*

Supervisor: Bonner, Simon, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Statistics and Actuarial Sciences

Recommended Citation

Kim, Han-na, "The Analysis of Mark-recapture Data with Individual Heterogeneity via the H-likelihood" (2022). *Electronic Thesis and Dissertation Repository*. 8716.
https://ir.lib.uwo.ca/etd/8716

Abstract

Mark-recapture methods have played a key role in ecological studies monitoring populations of wild animals, including those threatened by human disturbance. One consideration in the analysis of mark-recapture data is individual variations in the rate of detecting individuals. Failure to account for a variation can lead to biased inference, but classical methods for modelling heterogeneity require numerical integration and can be computationally intensive or numerically unstable. This thesis develops a novel approach based on the h-likelihood, which can remedy such difficulties by avoiding any numerical integration.

In the first project, I present my h-likelihood for fitting the fundamental model describing individual heterogeneity in mark-recapture studies. The conditional likelihood approach allows the model to be considered as a generalized linear mixed model (GLMM), and building on this connection, I construct the h-likelihood for the model in the context of the GLMM. In addition, I derive a bias correction for the model parameters and develop inference for the population size via the Horvitz-Thompson estimator.

My second project extends my approach to fit advanced models accounting for individual heterogeneity in which the capture probability may also depend on time and individuals' trap responses. The conditional likelihood approach enables these models to be treated as vector GLMMs. The h-likelihood approach from the first project is then extended to fit these models by allowing the response variables to be multi-dimensional. Bias correction is again considered, and the Horvitz-Thompson estimator is employed for estimating the population size as before.

Finally, I develop my h-likelihood approach to fit more flexible models describing individual heterogeneity. Standard models assume a linear relationship on some scale of the detection rate. The model I consider relaxes this assumption by applying the structure of generalized additive models via penalized spline, which can be regarded as a GLMM when the conditional likelihood is penalized for roughness. I apply the h-likelihood approach to fit this model and again estimate the population size using the Horvitz-Thompson estimator.

Keywords: Ecological statistics, Generalized additive models, Generalized linear mixed models, H-likelihood, Horvitz-Thompson estimator, Mark-recapture

Lay Summary

Mark-recapture methods play a key role in ecological studies monitoring wild animal populations. One consideration in analyzing mark-recapture data is individual variation in the detection rate. Classical methods for modelling heterogeneity require numerical integration and may be computationally intensive. This thesis presents a novel approach based on the h-likelihood to remedy such difficulties by avoiding numerical integration.

First, I present the h-likelihood approach for fitting the fundamental model describing individual heterogeneity in mark-recapture studies. The conditional likelihood approach allows the model to be regarded as a generalized linear mixed model (GLMM). I construct the h-likelihood for the model in the context of this GLMM. The population size is estimated via the Horvitz-Thompson estimator.

Second, I extend my approach to fit advanced models accounting for individual heterogeneity along with variation over time and individuals' trap responses. The conditional likelihood approach enables these models to be treated as vector GLMMs. The approach from the first project is adapted to fit these models with multi-dimensional response variables. The Horvitz-Thompson estimator is again employed to estimate the population size.

Finally, I develop the h-likelihood approach to fit more flexible models describing individual heterogeneity. As standard models assume a linear relationship, I apply the structure of generalized additive models through B-spline, which can be considered as a GLMM with the conditional likelihood penalized for roughness. Again, I apply the h-likelihood to fit this model and to estimate the population size using the Horvitz-Thompson estimator.

Acknowledgements

It is a genuine pleasure to express my deep sense of thanks and gratitude to my supervisor Dr. S. J. Bonner, Professor at the Department of Statistical and Actuarial Sciences, Western University. Dr. Bonner provided me invaluable guidance through this research and deeply inspired me with his dynamism, sincerity and motivation.

During my Ph.D., I got much support from my family and friends, who made my time at UWO joyful. I would like to give special thanks to my mother, Yeon-ok Roh, my sister, Seunghye Kim, and my best friends, Dr. Mihwa Seong, Dr. Lingzhi Chen and Ju Young Lee.

I would also like to give special thanks to some faculty members in the Department of Statistical and Actuarial Sciences, Western University. They proved of great assistance to me in making it through a rough time, especially when I lost my father.

Contents

Abstract **i**

Lay Summary **ii**

List of Figures **vi**

List of Tables **viii**

1 Introduction **1**

1.1 Common Methodology 3

1.1.1 MR models for heterogeneity 3

1.1.2 GLM, GLMMs and GAMs 5

1.1.3 H-likelihood 8

1.2 Summary of Projects 10

2 H-likelihood approach to basic MR models with heterogeneity **12**

2.1 Introduction 12

2.2 Methods 14

2.2.1 Description of MR model: $\mathcal{M}_h$ 14

2.2.2 Conditional h-likelihood 16

2.2.3 Bias correction for MHLEs 17

2.2.4 Fitting algorithm 19

2.2.5 Estimation of population size 23

2.3 Simulation Study 27

2.4 Application 30

2.5 Discussion 35

3 H-likelihood approach to extended MR models with heterogeneity **37**

3.1 Introduction 37

3.2 Method 39

3.2.1 Description of extended MR models: $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$ 39
3.2.2 Conditional h-likelihood 41
3.2.3 Bias correction for MHLEs 42
3.2.4 Fitting Algorithm 43
3.2.5 Estimation of population size 49
3.3 Simulation Study 52
3.4 Application 58
3.5 Discussion 60

4 P-spline approach to MR models with heterogeneity via h-likelihood 63
4.1 Introduction 63
4.2 Method 65
4.2.1 Description of MR model: $\mathcal{M}_s$ with smoothing functions 65
4.2.2 Connection between penalized likelihood and h-likelihood 67
4.2.3 Fitting algorithm 68
4.2.4 Estimation of population size 71
4.3 Simulation Study 73
4.4 Application 75
4.5 Discussion 77

5 Conclusion 79

Bibliography 82

Appendix 87

A R Code for Snowshoe Hare Data via the H-likelihood in Chapter 2 88

B R Code for Snowshoe Hare Data via the H-likelihood in Chapter 3 95

C R Code for Snowshoe Hare Data via the H-likelihood in Chapter 4 110

Curriculum Vitae 116

List of Figures

2.1 Relative Bias (RB) for $\hat{N}$ (upper left), $\hat{\alpha}$ (upper right), $\hat{\beta}_h$ (lower left) and $\hat{\sigma}_v$ (lower right), when $T = 5$, and $N = 100$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values. 29

2.2 Relative root mean square error (RRMSE) for $\hat{N}$ (upper left), $\hat{\alpha}$ (upper right), $\hat{\beta}_h$ (lower left) and $\hat{\sigma}_v$ (lower right), when $T = 5$, and $N = 100$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values. 30

2.3 Relative Bias (RB) for $\hat{N}$ (upper left), $\hat{\alpha}$ (upper right), $\hat{\beta}_h$ (lower left) and $\hat{\sigma}_v$ (lower right), when $T = 8$, and $N = 250$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values. 31

2.4 Relative root mean square error (RRMSE) for $\hat{N}$ (upper left), $\hat{\alpha}$ (upper right), $\hat{\beta}_h$ (lower left) and $\hat{\sigma}_v$ (lower right), when $T = 8$, and $N = 250$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values. 32

3.1 Relative Bias (RB) for $\hat{N}$ (top left), $\hat{\alpha}_t$ (top right), $\hat{\beta}_h$ (middle left), γ (middle right) and $\hat{\sigma}_v$ (lower right), when $T = 5$, and $N = 100$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values. 54

3.2 Relative root mean square error (RRMSE) for $\hat{N}$ (top left), $\hat{\alpha}_t$ (top right), $\hat{\beta}_h$ (middle left), γ (middle right) and $\hat{\sigma}_v$ (lower right), when $T = 5$, and $N = 100$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values. 55

3.3 Relative Bias (RB) for $\hat{N}$ (top left), $\hat{\alpha}_t$ (top right), $\hat{\beta}_h$ (middle left), γ (middle right) and $\hat{\sigma}_v$ (lower right), when $T = 8$, and $N = 250$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values. 56

3.4 Relative root mean square error (RRMSE) for $\hat{N}$ (top left), $\hat{\alpha}_t$ (top right), $\hat{\beta}_h$ (middle left), γ (middle right) and $\hat{\sigma}_v$ (lower right), when $T = 8$, and $N = 250$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values. 57

4.1 Relative bias (RB) (left) and relative root mean square error (RRMSE) (right) for $\hat{N}$ when $(T, N) = (5, 100)$ (upper) and $(T, N) = (8, 250)$ (lower). The scales of the y-axis of the plots for RB and RRMSE were changed to accommodate some extreme values when $\alpha = -2.2$. 74

List of Tables

1.1 Examples of distributions of v_j and their canonical scales. 9

2.1 Summary of notation used in Chapter 2. 15

2.2 Percent coverage for N based on both the log-normal CI and Wald CI as a function of σ_v and α which in turn defines the median capture probability, $\bar{p}_i$. . . 33

2.3 Results of estimating the population size of snowshoe hares. Three methods based on the h-likelihood, numerical integration via program MARK and the Bayesian method using MCMC implemented via JAGS are applied to fit $\mathcal{M}_h$ and estimate the population size. For the methods based on the h-likelihood and numerical integration, two CIs are reported: log-normal CI (left) and Wald CI (right). Only one CI is reported for the Bayesian estimate. 34

3.1 Summary of notation used in Chapter 3. 39

3.2 Percent coverage for N based on both the log-normal CI and Wald CI as a function of σ_v and α_t which in turn defines the median initial capture probability, $\bar{p}^{\dagger}_{it}$. 53

3.3 Results of estimating the population size of snowshoe hares. Three methods based on: h-likelihood, numerical integration via program MARK and Bayesian method (using MCMC) via JAGS are applied to fit $\mathcal{M}_h$, $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$ and estimate the population size. For the methods based on the h-likelihood and numerical integration, two CIs: log-normal CI (left) and Wald CI (right), are reported. For model comparison, values of model measurements according to the methods, cAIC for the h-likelihood, AICc for numerical integration, and WAIC for Bayeisan method, are included. 62

4.1 Summary of notations used in Chapter 4. 65

4.2 Percentage coverage for N based on log-normal CI and Wald CI as a function of α which in turn defines the backtransformed capture probability $\bar{p}_i$. 75

4.3 Results of estimating the population size of mountain pygmy possums. Six approaches for fitting $\mathcal{M}_h$ as: a GLM without a smoothing function on the body mass, a GAM with kernel smoothing, that via backfitting with the equivalent degree of freedom being 2 and 12, that via GCV and my approach using the h-likelihood, where the model is parameterized by basis functions (B-spline) with the degree being 3 and $L2$-norm for the penalization of coefficients of the basis functions. For all approaches, two CIs: log-normal CI (left) and Wald CI (right), are reported. 77

Chapter 1

Introduction

Understanding the dynamic forces of a population and the factors that affect these forces is a key step to developing conservation policies and management plans. In biological science, the study of population dynamics has aimed to preserve many endangered species distinguished by the Red List of the International Union for Conservation of Nature (IUCN). According to the IUCN, over 40,000 species are threatened with extinction, and their number is about 28% of all assessed species worldwide (see https://www.iucnredlist.org/). Considering specific taxa, Hoffmann et al. (2011) presented combined results for 5487 species of mammals under a conservation status and found that one-fifth are threatened with extinction with a higher risk for large-bodied mammals than for other species. Similarly, Dulvy et al. (2014) reported that one-quarter of sharks and ray species are in danger, while large-bodied and shallow-water species are at the highest risk of extinction. For amphibians, which comprise the second-largest proportion of the Red-listed species next to cycads (a relatively small group of plants containing only 300 species), Trull et al. (2018) found that the worldwide decline of amphibians is strongly related to climate changes, human disturbance, the presence of invasive species, nature-system modification, and pollution, with these factors compounding each other where they appear together. The endangered birds on the Red List are also significantly affected by climates change, along with the above factors influencing amphibians, as shown by Langham et al. (2015).

Another practice where understanding the dynamic forces of a population is important is the study of bird migration. Migration of birds species including storks and swallows has been observed since 4th century, BC, and over 4000 bird species, 40% of all birds species in the world, are known to migrate according to the Royal Society for the Protection of Birds (RSPB) (see https://www.rspb.org.uk/). As one main cause that birds migrate is to move from areas with a low or decreasing resources to other areas with high or increasing resources, studying their movement has frequently informed about the ecosystem and resource availability in an area.

This information is further applied to the control of the ecosystem artificially by managing bird migration; for example, Karp et al. (2013) prevented the consumption of US$192 ha-year on average for pest-removal in Costa Rica by providing habitats for borer-consuming birds. Since early 2000s, however, the climate change has significantly changed bird migration behaviours, including the timing of migration, as shown by Jenni and Kery (2003). To restore the behaviours and manage bird migrations better, potential factors that influence bird behaviour, including the climate change, have been actively studied. To avoid misleading analysis, it has been essential to choose appropriate experimental methods to obtain data and develop different statistical models to analyze data, depending on experimental conditions.

Mark-recapture (MR) experiments are one common method used to collect data to study the dynamics of many different populations. As an example, Pilliod et al. (2010) performed a MR experiment to collect data for the Eungella torrent frog (*Taudactylus eungellensis*) on the Red List, for which the exact number cannot be counted as they have habitats worldwide. In practice, animals are difficult to follow in the wild, and it is often the case that only a portion of individuals of interest can be observed throughout a study period. The MR method is convenient for such a case in that data are collected through a repeated process of capturing, marking, and recapturing a subset of individuals in a study area. Assuming that the individuals in the sample are representative of the entire population, the pattern of recaptures of the marked individuals provides information that can be used to make inferences about the entire population, including the individuals never captured (e.g., to estimate the total population size).

One common challenge in analyzing data from MR studies is accounting for individual heterogeneity. Individual variation is almost inevitable in the MR method as individuals never behave in exact same manner. For example, individuals with different characteristics, such as body mass, can have different detection rates. Many previous authors have considered the use of methods to account for individual heterogeneity when modelling data from MR studies. The study of Otis et al. (1978) conducted to estimate the population size for snowshoe hares (*Lepus americanus*) is a typical example that shows that modelling heterogeneity provides a more reliable estimate of the population size than the estimate without modelling it. Generally, the models with individual heterogeneity account for two forms of individual variation: that due to measurable characteristics (i.e., observed heterogeneity) or that due to unmeasured characteristics (i.e., unobserved heterogeneity). Some works have clearly shown that ignoring either type of individual heterogeneity causes biased estimates, which may lead to incorrect answers to ecological questions that researchers investigate (Hwang and Huggins, 2005; Pledger and Efford, 1998).

The three projects in this thesis consider multiple MR models accounting for both observed and unobserved heterogeneity, modelled by individual covariates and random effects. Each project develops a novel method for fitting different MR models based on a statistical approach using the h-likelihood, originally proposed by Lee and Nelder (1996). All previous methods for fitting the same MR models have been based on either frequentist or Bayesian approaches, which require quadrature or sampling methods to integrate out random effects and to compute the likelihood function. The h-likelihood, on the contrary, eliminates the need for numerical integration and allows for fitting algorithms for the MR models that are simple to implement in modern software. The remainder of this chapter provides background information, including an introduction to MR models and the h-likelihood, and a summary of the three projects.

1.1 Common Methodology

1.1.1 MR models for heterogeneity

In a typical MR experiment, subsets of individuals from a population are repeatedly captured (or in some way detected), marked (or otherwise identified), and released back into the population for a fixed number of times. The aim of the experiment is to understand characteristics of the population, such as an unknown population size, through information provided by the recapture of individuals. To describe MR data in this chapter, the following assumptions and mathematical notations are applied. Suppose that the experiment is conducted over T distinct sampling occasions, indexed by $t = 1, ..., T$. From the population, whose unknown size is denoted by N, $n \leq N$ individuals are captured in total. These individuals captured at least once are indexed by $i = 1, ..., n$, and similarly, those never captured are indexed by $i = n + 1, ..., N$. I additionally assume that the population is closed, which means that no birth, death, immigration, or emigration of individuals can occur during the experiment.

Statistical models describing MR data (i.e., MR models) are based on the observed variables

$$y_{it} = I(\text{individual } i \text{ is captured on occasion } t)\,,$$

where $I(\cdot)$ is the indicator function that returns 1 if the condition given in the argument is satisfied, and 0 otherwise. Each y_{it} is assumed to be an observation independently drawn from the Bernoulli distribution with the success probability, $p_{it} = P(y_{it} = 1)$. The success probability is called the capture probability in many areas of MR literature. Depending on the assumptions about the capture probability, p_{it}, may be simplified into p_i, p_t or p, where the subscript reduction indicates that p_{it} are equal over the index or indices disregarded. For example, if individual i has the same capture probability across the T sampling occasions, then $p_i = p_{i1} = ... = p_{iT}$. If this is the case, then it is sufficient to model the response variables as the total number of times an individual is captured, which is observed as $y_{i.} = \sum_{t=1}^{T} y_{it}$. Such response variables are independent of each other, and each follows the binomial distribution with T trials and success probability p_i.

Given that the capture probability is unknown, the likelihood function is written by

$$\mathcal{L}(N, \mathbf{p}; \mathbf{y}) = \prod_{i=1}^{N} \prod_{t=1}^{T} p_{it}^{y_{it}} (1 - p_{it})^{1-y_{it}} ,$$

using the Bernoulli response variables, where $\mathbf{p}$ and $\mathbf{y}$ are the vectors including all p_{it} and y_{it} across i and t, respectively. For clarity, the semicolon in $\mathcal{L}(\cdot)$ divides arguments into two types of quantities: unknown quantities on the left of semicolon, and known quantities, such as data, on the right. Huggins (1989) derived that this likelihood can be split into two functions as the following,

$$\begin{aligned}\mathcal{L}(N, \mathbf{p}; \mathbf{y}) &= \prod_{i=1}^{N} \prod_{t=1}^{T} p_{it}^{y_{it}} (1 - p_{it})^{1-y_{it}} \\ &= \left[\prod_{i=1}^{n} \frac{\prod_{t=1}^{T} p_{it}^{y_{it}} (1 - p_{it})^{1-y_{it}}}{1 - \prod_{t=1}^{T} (1 - p_{it})} \right] \times \\ &\quad \left\{ \prod_{j=1}^{n} \left[1 - \prod_{t=1}^{T} (1 - p_{jt}) \right] \prod_{k=n+1}^{N} \prod_{t=1}^{T} p_{kt}^{y_{kt}} (1 - p_{kt})^{1-y_{kt}} \right\} ,\end{aligned} \tag{1.1}$$

and maximized the first term, related to captured individuals only, to obtain the estimates of the capture probabilities, but not the population size, N. His method was shown to be valid as the first term corresponds to the conditional likelihood (Kalbfleisch and Sprott, 1973) taking N as a nuisance parameter. To estimate unknown population size, N, Huggins (1989) subsequently obtained the Horvitz-Thompson (HT) estimator (Horvitz and Thompson, 1952), which is

$$\begin{aligned}\hat{N}(\mathbf{p}) &= \sum_{i=1}^{N} I\Big(\sum_{i=1}^{T} y_{it} \geq 1 \Big) \frac{1}{P(\sum_{i=1}^{T} y_{it} \geq 1)} \\ &= \sum_{i=1}^{n} \frac{1}{1 - \prod_{t=1}^{T} (1 - p_{it})} ,\end{aligned} \tag{1.2}$$

a function of the true values of p_{it}. In practice, the values are unknown, so that the estimates of the capture probabilities, $\hat{p}_{it}$, are obtained from the conditional likelihood, and then substituted in place of p_{it} in the HT estimator.

The MR models build a mathematical relationship between data observed during the MR experiment and parameters characterizing a population. In general, the relationship is expressed by

$$p_{it} = g(\eta_{it}),$$

where

$$\eta_{it} = f(\mathbf{x}_{it}).$$

Here $\mathbf{x}_{it}$ is a vector of covariates observed for individual i on occasion t, $f(\cdot)$ is a known function, η_{it} is called the linear predictor, and $g(\cdot)$ is the link function. The link function is often set as the logit (i.e., $\text{logit}(a) = \log(a/(1-a))$) or log function. Otis et al. (1978) and Huggins (1989) defined η_{it} as a linear combination of the components in $\mathbf{x}_{it}$ and parameters associated with them; for example, $\eta_{it} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ if $\mathbf{x}_{it} = (x_1, x_2)'$, and β_0, β_1 and β_2 are the parameters to be estimated. The types of covariates can be any information quantified, such as individual and environmental, and depending on its properties, the index i or t may be dropped from $\mathbf{x}_{it}$. The structure of the MR models through the linear combination linked to p_{it} mimics that of generalized linear models (GLMs) as described below.

1.1.2 GLM, GLMMs and GAMs

GLMs (Nelder and Wedderburn, 1972) are applied for analyzing a wide variety of research outcomes. They generalize linear regression models, so that the response variables are related to factors and covariates via a specific link function and follow distributions other than a normal distribution. The class of GLMs covers many well-known statistical models, including logistic regression models for binary data and also Poisson log-linear regression models for count data.

The structure of a GLM consists of three components: a random component, a systematic component, and a link function. The random component specifies the distribution of the response variables, n random variables, denoted by $Y_1, ..., Y_n$, whose observations are given by $y_1, ..., y_n$. It is assumed that the response variables are independent of each other conditional on the explanatory variables (i.e., factors and covariates) and have the density function of form,

$$f(y_i; \theta_i, \phi) = \exp\left(\frac{\theta_i t(y_i) - b(\theta_i)}{\phi} + c(y_i, \phi)\right) \tag{1.3}$$

for $i = 1, ..., n$, where θ_i is the natural parameter, ϕ is the dispersion parameter, and $t(\cdot)$, $b(\cdot)$ and $c(\cdot, \cdot)$ are some known functions. The family of distributions with the density function written as above is called the exponential family. The systematic component includes the explanatory variables that combine to form the linear predictor

$$\eta_i = \mathbf{x}_i'\boldsymbol{\beta}, \tag{1.4}$$

where $\boldsymbol{\beta}$ is the vector of intercepts and fixed effects associated with the factors and covariates composing the vector, $\mathbf{x}_i$. The last component, the link function, then connects η_i to θ_i through the mean of $t(Y_i)$, denoted by $\mu_i = E(t(Y_i))$, such that

$$\eta_i = g(\mu_i)\,.$$

This function is said to be the canonical link if $g(\cdot)$ is chosen to provide the identity $\eta_i = \theta_i$. More generally, the linkage between η_i and θ_i is established by the property of the exponential family that

$$\kappa_j(t(Y_i)) = \frac{\partial^j b(\theta_i)}{\partial \theta_i^j}\,,$$

where $\kappa_j(t(Y_i))$ is the j-th cumulant of the random variable $t(Y_i)$, and thus the mean is given by $\mu_i = \kappa_1(t(Y_i)) = \partial b(\theta_i)/\partial\theta_i$, depending on θ_i. The property also provides a formula for the variance, $\mathrm{Var}(t(Y_i)) = \kappa_2(t(Y_i)) = \partial\mu_i/\partial\theta_i$.

In some cases, it is necessary to extend the linear predictor to include random effects which describe extra variation in the response variable that may, for example, be caused by a hierarchical structure in the population due to sampling design. Generalized linear mixed models (GLMMs) are the extended case of GLMs with such a linear predictor, where it is nearly always assumed that the random effects follow a multivariate normal distribution with mean zero vector and unknown variance–covariance matrix. Specifically, GLMMs have the general form of the linear predictor,

$$\eta_i = \mathbf{x}_i'\boldsymbol{\beta} + \mathbf{z}_i'\mathbf{v}\,, \tag{1.5}$$

extending equation (1.4) by adding the extra term, $\mathbf{z}_i'\mathbf{v}$, where $\mathbf{z}_i$ is a known covariate vector associated with $\mathbf{v} = (v_1, ..., v_m)$, v_j is a random effect, and $\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \Sigma)$. The variance–covariance matrix Σ is positive semidefinite, symmetric and has $m(m+1)/2$ unique elements. To simplify computation of parameter estimates by removing redundant elements when Σ has a specific correlation structure (e.g., $\Sigma = \lambda\mathbf{A}$, where $\mathbf{A}$ is a known matrix such a priori), it is often decomposed by the Cholesky factorization $\Sigma = \mathbf{LDL}'$ for some triangular matrix, $\mathbf{L}$, and a diagonal matrix, $\mathbf{D}$. Hence, the linear predictor in equation (1.5) can be replaced by

$$\eta_i = \mathbf{x}_i'\boldsymbol{\beta} + \mathbf{z}_i^{*\prime}\mathbf{v}^*\,,$$

where $\mathbf{z}_i^{*\prime} = \mathbf{z}_i'\mathbf{L}$, $\mathbf{v}^* = (v_1^*, ...v_m^*)'$, and $\mathbf{v}^* \sim \mathcal{N}(\mathbf{0}, \mathbf{D})$. In this way, statistical methods provide more stable parameter estimates than considering the model with parameterization, as in equation (1.5).

Another class of the models that extend GLMs in a different way is the class of generalized additive models (GAMs) which relax the assumption of linearity. In the framework of GAMs, the linear predictor is defined by

$$\eta_i = f_1(x_{i1}) + \cdots + f_p(x_{ip}),$$

given that the dimension of covariates is p, and $f_1(\cdot), ..., f_p(\cdot)$ are smooth functions such as splines or polynomial functions with a fixed degree. I shall consider the smooth functions expanded by the B-spline, which implies that

$$f_l(x_{il}) = \sum_{k=1}^{K_l} \beta_{lk} b_{lk}(x_{il}) \tag{1.6}$$

for $l = 1, ..., p$, where $b_{lk}(\cdot)$ are the basis functions as determined in the work of de Boor (1971). The computation of these functions is based on knots (i.e., pre-set points in the range of x_{il}) and can be readily obtained by recursive algorithms implemented in most modern software packages. The model in equation (1.6) is a parametric form as it is a linear combination of parameters and known covariates, so that it can be regarded as being in the class of GLMs.

One challenge with the above model is that its flexibility can lead to overfitting if it is fitted as a GLM. To prevent this issue, a penalty term controlling roughness of the function $f_l(\cdot)$ is incorporated into the likelihood, resulting in what is called the penalized likelihood (Green, 1987). The penalized likelihood on the log scale is

$$M(\boldsymbol{\beta}^*, \rho; \mathbf{y}) = \ell(\boldsymbol{\beta}^*; \mathbf{y}) - \sum_{l=1}^{p} \frac{1}{2}\rho_l J_l\,, \tag{1.7}$$

where $\boldsymbol{\beta}^*$ is the vector of all regression parameters, β_{lk}, $\ell(\boldsymbol{\beta}^*; \mathbf{y})$ is the log-likelihood function of the original model, J_l is roughness penalties depending on the basis functions, and ρ_l are tuning parameters that control the roughness penalty. The increase of J_l or ρ_l results in more smooth, and vice versa. In my project, J_l is specified by the L_2-norm,

$$\begin{aligned} J_l &= \int |f_l^{(d)}(x_l)|^2 dx \\ &= \boldsymbol{\beta}_l^{*'} \mathbf{P}_l \boldsymbol{\beta}_l^*\,, \end{aligned}$$

where the (a, b)-th element of $\mathbf{P}_l$ is $\int b_{la}^{(d)}(x_l) b_{lb}^{(d)}(x_l) dx_l$, and the superscript (d) denotes the d-order derivative of the basis functions. The second term in equation (1.7) is then replaced by

$$-\sum_{l=1}^{p} \frac{1}{2}\rho_l \boldsymbol{\beta}_l^{*'} \mathbf{P}_l \boldsymbol{\beta}_l^*\,, \tag{1.8}$$

which can be regarded as the function proportional to the log of the density of multivariate normal distribution for $\boldsymbol{\beta}_l^*$ with mean vector, $\mathbf{0}$, and the variance–covariance matrix, $(\rho_l \mathbf{P}_l)^{-1}$. Therefore, the penalized likelihood in equation (1.7) is equivalent to the joint density of $\mathbf{y}$, whose distribution belongs to the exponential family, and $\boldsymbol{\beta}^*$, which shows that GAMs belong to the class of GLMMs with p groups of random effects sharing the p separate dispersion parameters, ρ_1,...,ρ_p.

1.1.3 H-likelihood

In this dissertation, I focus on the estimation procedure based on the h-likelihood as the primary method for fitting MR models framed as either GLMMs or GAMs. The h-likelihood method was first introduced by Lee and Nelder (1996) as an alternative to the standard Bayesian and frequentist approaches for fitting GLMMs.

The h-likelihood is defined by the joint density of data and random effects, regarded as the function of all unknown quantities that include parameters and the random effects. For example, suppose that the models for data are GLMMs, as described in the previous section, in which the random effects v_j are assumed to be independent of each other and follow $\mathcal{N}(0, \lambda)$ for simplicity. The h-likelihood for GLMMs is given by

$$\mathcal{H}(\boldsymbol{\beta}, \mathbf{v}, \phi, \lambda; \mathbf{y}, \mathbf{v}) = \prod_{i=1}^{n} f(y_i|\mathbf{v}; \boldsymbol{\beta}, \phi) \prod_{j=1}^{m} f(v_j|\lambda)\,, \tag{1.9}$$

while the marginal likelihood for GLMMs, employed by classical and Bayesian approaches, is

$$\mathcal{L}(\boldsymbol{\beta}, \phi, \lambda; \mathbf{y}) = \int_{\mathcal{R}^m} \mathcal{H}\; d\mathbf{v}\,. \tag{1.10}$$

Maximum likelihood estimates (MLEs) of parameters are obtained by maximizing equation (1.10) with respect to all parameters, $\boldsymbol{\beta}$, ϕ and λ; meanwhile, in the h-likelihood estimation procedure, estimates of all unknown quantities are obtained by maximizing $\mathcal{H}$ in equation (1.9), or equivalently $h = \log(\mathcal{H})$, with respect to all parameters, $\boldsymbol{\beta}$, ϕ and λ but also the random effects, $\mathbf{v}$, simultaneously. In consequence, the resulting estimates from the h-likelihood, $\hat{\boldsymbol{\beta}}$, $\hat{\mathbf{v}}$, $\hat{\phi}$ and $\hat{\lambda}$, are the so-called maximum h-likelihood estimates (MHLEs). For clarity of notation, the semicolon in $\mathcal{H}(\cdot;\cdot)$ divides arguments into unknown quantities on the left of the semicolon and quantities whose distribution is considered in constructing the joint density on the right. The random effects, $\mathbf{v}$, appear on both sides of the semicolon as the h-likelihood is constructed in part from the density of $\mathbf{v}$, but $\mathbf{v}$ is a vector of unknown quantities to be estimated by maximizing the h-likelihood.

Table 1.1: Examples of distributions of v_j and their canonical scales.

Distribution of v_j	Canonical Scale
Normal	$\tau(v_j) = v_j$
Gamma	$\tau(v_j) = \log(v_j)$
Beta	$\tau(v_j) = \text{logit}(v_j) = \log\left(\frac{v_j}{1 - v_j}\right)$
Inverse-Gamma	$\tau(v_j) = -v_j^{-1}$

It is noted that not all extended likelihoods (i.e., any joint density of data and random effects) can be considered as h-likelihoods within the framework developed by Lee and Nelder (1996). The reason for this is that unless certain restrictions are applied, maximizing the extended likelihood may produce nonsense estimates (e.g., $-\infty$; see example in Lee et al. (2017, p.111)) while maximizing the marginal likelihood for the same model produces valid MLEs that obey the usual asymptotic properties. To avoid this problem, it was shown by Lee and Nelder (1996, 2001) and Lee et al. (2017) that the random effects must be transformed according to a function determined by their distribution. That is, for GLMMs, the linear predictor must be specified by

$$\eta_i = \mathbf{x}_i'\boldsymbol{\beta} + \mathbf{z}_i'\tau(\mathbf{v}),$$

where $\tau(\cdot)$ is a specific function depending on the distribution of $\mathbf{v}$. The function, $\tau(\mathbf{v})$, is known as the canonical scale and shown to be unique up to linear transformation (Lee et al., 2017). Some examples of canonical scales are provided in Table 1.1 for the case in which v_j can follow a common distribution other than the normal and are independent of each other. The canonical scale for the normal random effects v_j in GLMMs is the identity function, $\tau(\mathbf{v}) = \mathbf{v}$, so that the extended likelihood in equation (1.9) is a valid h-likelihood.

A key result of the estimation procedure based on a valid h-likelihood is that MHLEs also satisfy the usual asymptotic properties of the MLEs (Lee and Nelder, 1996). For example, given a large sample size, n, the MHLE, $\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\beta}}', \hat{\mathbf{v}}', \hat{\phi}, \hat{\lambda})'$, approximately follows a normal distribution such that

$$\hat{\boldsymbol{\theta}} - \boldsymbol{\theta} \dot{\sim} \mathcal{N}(\mathbf{0}, \text{Var}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})),$$

where

$$\text{Var}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \approx \left(-\frac{\partial^2 h}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\right)^{-1}. \tag{1.11}$$

Dividing $\boldsymbol{\theta}$ into $\boldsymbol{\delta} = (\boldsymbol{\beta}', \mathbf{v}')'$ and $\boldsymbol{\rho} = (\phi, \lambda)'$, the variance–covariance matrix is rewritten by

$$\frac{\partial^2 h}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'} = \begin{pmatrix} \dfrac{\partial^2 h}{\partial\boldsymbol{\delta}\partial\boldsymbol{\delta}'} & \dfrac{\partial^2 h}{\partial\boldsymbol{\delta}\partial\boldsymbol{\rho}'} \\ \dfrac{\partial^2 h}{\partial\boldsymbol{\rho}\partial\boldsymbol{\delta}'} & \dfrac{\partial^2 h}{\partial\boldsymbol{\rho}\partial\boldsymbol{\rho}'} \end{pmatrix}, \tag{1.12}$$

and Lee and Nelder (1996) showed that the off-diagonal blocks are matrices of all 0s if the link function, $g(\cdot)$, and the canonical scale, $\tau(\cdot)$, are identical. In fact, it was derived by Lee and Nelder (1996) that the off-diagonal blocks are approximately matrices of all 0s even though $g(\cdot)$ and $\tau(\cdot)$ are different. This implies that the estimations of $\boldsymbol{\delta}$ and $\boldsymbol{\rho}$ are approximately separable and so leads to the fitting procedure that recursively performs two separate estimation steps until convergence: one for estimating $\boldsymbol{\delta}$ from the h-likelihood while fixing $\boldsymbol{\rho} = \hat{\boldsymbol{\rho}}$ and the other for estimating $\boldsymbol{\rho}$ from the h-likelihood while fixing $\boldsymbol{\delta} = \hat{\boldsymbol{\delta}}$. Once $\hat{\boldsymbol{\delta}}$ and $\hat{\boldsymbol{\rho}}$ are obtained by the fitting procedure, equation (1.11) can be further approximated by

$$\mathrm{Var}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) \approx \left.\begin{pmatrix} \left(-\dfrac{\partial^2 h}{\partial\boldsymbol{\delta}\partial\boldsymbol{\delta}'}\right)^{-1} & \mathbf{0} \\ \mathbf{0} & \left(-\dfrac{\partial^2 h}{\partial\boldsymbol{\rho}\partial\boldsymbol{\rho}'}\right)^{-1} \end{pmatrix}\right|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}. \tag{1.13}$$

Wald-type inferences can be obtained from this approximated variance–covariance matrix; for example, the standard error of the i-th element in $\hat{\boldsymbol{\delta}}$ is approximately the i-th diagonal of the upper-left block matrix in equation (1.13), $(-\partial^2 h/\partial\boldsymbol{\delta}\partial\boldsymbol{\delta}')^{-1}|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}$.

1.2 Summary of Projects

The first project of my thesis (Chapter 2) involved developing the h-likelihood approach for fitting the simplest MR model accounting for individual heterogeneity. This model depends on individual covariates and random effects to model individual heterogeneity but no other covariates to explore population dynamics. To fit the model, I construct the h-likelihood based on MR data conditional on the individuals captured at least once, and show that the h-likelihood is equivalent to the h-likelihood of a GLMM in which the response variables follow a positive-binomial distribution of Patil (1962). Building on this connection, I employ the fitting procedure of Lee and Nelder (1996, 2001), developed for fitting the class of GLMMs, for estimating all parameters in the model, except for unknown population size. The unknown population size is sequentially estimated by the HT estimator, given both parameters and random effects estimated from the h-likelihood.

The second project (Chapter 3) is an extension of the first project to the case where the capture probability changes during the experiment either due to time or behavioural effects. The idea of this extension originated from the MR models of Otis et al. (1978), where the additional parameters can relax the strong assumption that the capture probability is equal over all sampling occasions. In particular, the time effect discretizes the capture probability by the number of sampling occasions, and the behavioural effect divides the capture probability into the probability of initially capturing an individual and that of recapturing the individual. The connection between these extended MR models and a general algebraic-advantageous form of statistical models is available by extending the class of GLMMs to vector GLMMs (VGLMMs). The framework of VGLMMs allow the response variables to be multi-dimensional, such as the vector of indicators that if an individual is captured or not for every sampling occasion. I construct the h-likelihood of these extended MR models based on MR data conditional on individuals captured at least once and show that the h-likelihood is a special case of the h-likelihood of VGLMMs. I propose the fitting algorithm for the models regarded as VGLMMs, similar to that of the simplest model in the previous chapter. The estimation of unknown population size is again based on the HT estimator, given the parameters and random effects estimated from the h-likelihood.

The third project (Chapter 4) considers MR models that I use a GAM formulation to allow for the capture probability to depend on a non-linear function of individual covariates for modelling observed heterogeneity. This project is motivated by the potential that the relationship between observed capture history and the individual covariates is more complicated than simply a relationship explained by a linear combination of individual covariates. To alter the linear relationship, an arbitrary function is defined and applied to each covariate in the model, which mimics the structure of GAMs. Specifically, I convert each function into a sum of known functions, called basis functions, using B-spline, and assign a regression parameter to each basis function to regard the MR model as the model depending only on fixed effects within the capture probability. Because the basis dimension can be high, overfitting this type of MR model can be a severe issue, and therefore I construct the conditional likelihood of the MR models with a penalized term, which indicates that the MR models belong to the class of GLMMs. Hence, I directly apply the fitting algorithm based on the h-likelihood approach for fitting GLMMs, as shown by Lee and Nelder (1996, 2001). The estimation of the unknown population size is subsequently obtained by the HT estimator by substituting parameter estimates from the h-likelihood.

Chapter 2

H-likelihood Approach to Basic MR Models with Heterogeneity

2.1 Introduction

Modelling individual variation (i.e., differences among individuals) is a vital issue in analyzing MR data, as is the case in other areas of statistical modelling. In biological science, individual variation often refers to individual differences in physiological, morphological or behavioural traits, including fitness components effecting reproduction and survival (Clutton-Brock, 1988; Newton, 1989). Individual variation is commonly described under the umbrella term heterogeneity, and for many years, has been studied along with different statistical models to understand phenomena arising in biological evolution.

In many cases, heterogeneity has been modelled by data observable from individuals (i.e., individual covariates) obtained during the process of a MR experiment. Individual covariates are classified as discrete (i.e., categorical), such as strata, where each individual belongs to a sub-population having the same defined feature, and continuous (i.e., numeric), such as physical measurements, describing a characteristic of individuals using numerical values. To model MR data by individual covariates, the sequence of observations, data on when individuals are captured in a study (e.g., capture history), has been linked to parameters characterizing the effect of the individual covariates on these observations. The form of the linkage is often mimics the structure of generalized linear models (GLMs), which particularly connects the probability of capturing an individual (i.e., capture probability) to a linear combination of individual covariates.

One key drawback of modelling heterogeneity only based on individual covariates is that it is never possible to model all sources of between-individual variation. Examples are factors that are not observed due to some experimental issues (e.g., lack of measurement resources) as well as any unobservable factors that affect heterogeneity; however, the existence of unobservable factors cannot be clearly determined. This component of variation is commonly referred to as unobserved heterogeneity (Gimenez et al., 2018). To allow for unobserved heterogeneity, early works proposed MR models that assumed the capture probability to be a random variable for each individual following a distribution with support between 0 and 1. Some main works that contributed to such MR models were provided by Burnham and Overton (1979), who employed a beta distribution, Pledger et al. (2003), whose models assign a finite mixture distribution to the capture probability as well as to the survival probability, and Coull and Agresti (1999), who used a log-normal distribution alternative to the beta distribution of Burnham and Overton (1979). The MR models from the latter work is the model most similar to the structure of GLMMs and thus has drawn most interest from researchers when applying various statistical methods widely used for fitting GLMMs.

In general, the methods applied for fitting such MR models have been based on either frequentist or Bayesian approaches. The common idea is that parameters are estimated from the marginal likelihood, the function that integrates out the capture probability for each individual from the joint density of MR data and the capture probabilities (i.e., complete data likelihood). As the marginal likelihood has no analytic form, the methods have mainly relied on quadrature or sampling to approximate the integrals, in addition to EM algorithm and Laplace approximation, as shown by Van Deusen (2002) and Herliansyah et al. (2022), respectively. For the frequentist approach using quadrature, the methods are used to obtain MLEs, in which the marginal likelihood is typically approximated by the Gaussian quadrature (Coull and Agresti, 1999; Gimenez and Choquet, 2010; White and Cooch, 2017). Within this approach, the methods are again classified by two ways of defining the MR likelihood, one computed from the capture histories of all individuals including those never captured (Coull and Agresti, 1999; Gimenez and Choquet, 2010), and the other computed conditional on the individuals that are captured at least once (White and Cooch, 2017). It has been shown that both forms of the likelihood result in almost identical parameter estimates, governing the capture probability, but provide different estimates of the population size, for which the methods with a full likelihood estimate the population size at the same time as the other parameters, while for those with the conditional likelihood estimate the population size in a separate step, such as using the Horvitz-Thompson (HT) estimator. For the Bayesian approach, most methods are based on approximating the integral through Markov chain Monte Carlo (MCMC) sampling, and parameters are estimated through the properties of the posterior density approximated by MCMC (Bonner and Schofield, 2014; Durban and Elston, 2005; King and Brooks, 2008; King et al., 2016a; Royle et al., 2007). Related works differ based on the techniques used to implement

MCMC; for example, data augmentation completes the data by adding a large number of hypothetical unobserved individuals which enables the MR models to be fitted by pre-implemented MCMC in modern software (Royle et al., 2007). This technique particularly presets the super population of pseudo-individuals with the potential to be in the real population with an unknown probability, and thus the dimension of parameters to be estimated is fixed.

The objective of my work in this chapter is to develop a novel approach for fitting MR models including unobserved heterogeneity based on the h-likelihood of Lee and Nelder (1996). To my knowledge, this project represents the first attempt to apply the methods of h-likelihood to MR data. The MR models I consider have been widely used for exploring closed populations in which individuals do not migrate throughout a study area, and no birth or death of individuals occur during the experiment. For simplicity, I focus initially on models that assume that the capture probability is constant over time but do allow for individual effects as the result of covariates. I define the h-likelihood based on the MR data restricted to the individuals captured at least once, and the population size is subsequently estimated by the HT estimator. One main advantage of my approach is the availability of a simple fitting algorithm, which is similar to the iterative re-weighted least square (IRLS) that has been generally used for fitting ordinary GLMs. Moreover, another key advantage is that the h-likelihood is constructed from the complete-data likelihood itself, which avoids any integration by quadrature or sampling, as required for all previous methods. In addition to the fitting algorithm, the bias correction for parameter estimates from the h-likelihood is provided by following the technique introduced by Yun and Lee (2004), so the population size depending on these parameters is estimated more accurately.

The format of this chapter is as follows. Section 2.2 describes the framework of the MR models and the development of my approach based on the h-likelihood. Section 2.3 provides a simulation study with multiple scenarios, and in Section 2.4, I apply my approach to a well-known MR data set collected for snowshoe hares (*Lepus americanus*). Section 2.5 discusses the results obtained in Sections 2.3 and 2.4.

2.2 Methods

2.2.1 Description of MR model: $\mathcal{M}_h$

I consider a particular type of MR model which depends on two main components: individual covariates and random effects for modelling observed and unobserved heterogeneity, respectively. The models are specified in terms of the capture probability for each individual, assumed to follow a logit-normal distribution and be linked to a linear combination of the covariates and

Table 2.1: Summary of notation used in Chapter 2.

Notation	Definition
N	Unknown population size
n	Number of individuals captured
T	Number of sampling occasions
$y_{i.}$	Number of sampling occasions that individual i is captured
p_i	Capture probability for individual i
α	Intercept parameter
$\boldsymbol{\beta}$	Vector of all fixed effects including α
$\mathbf{x}_i$	Covariate vector associated with $\boldsymbol{\beta}$ for individual i
$\mathbf{X}$	Design matrix with the i-th row $\mathbf{x}_i$
v_i	Random effect for individual i
σ_v	Unknown standard deviation of v_i
$\boldsymbol{\theta}$	Vector of all unknown quantities; $(\boldsymbol{\beta}', \mathbf{v}', \sigma_v)'$
$\boldsymbol{\delta}$	Vector of all fixed and random effects; $(\boldsymbol{\beta}', \mathbf{v}')'$
$\mathbf{I}_a$	$a \times a$ identity matrix
$\mathbf{0}_a$	Vector of 0s with dimension of a
$\mathbf{0}_{a\times b}$	$a \times b$ matrix of 0s
$\mathbf{1}_a$	Vector of 1s with dimension of a
$\mathbf{a}$	Vectorized form of elements a_i for $i = 1, ..., n$; e.g., $\mathbf{v} = (v_1, ..., v_n)'$
dim($\mathbf{a}$)	Dimension of vector $\mathbf{a}$
diag(a, b, c)	Diagonal matrix consisting of the elements a, b and c in the diagonal
$a \odot b$	Element-wise multiplication of vectors a and b; e.g., $(2,4)' \odot (1,2)' = (2,8)'$
$a \oslash b$	Element-wise division of vectors a and b; e.g., $(2,4)' \odot (1,2)' = (2,2)'$

random effects, following a normal distribution. These assumptions were introduced by Coull and Agresti (1999) who also considered the same model I consider in their work. In addition, I apply the condition that the individual covariates are constant over time, and other types of covariates, such as environmental factors, are not studied for the simplicity of the models. The models also extend one of eight MR models proposed by Otis et al. (1978), which assumes that the capture probability for each individual is sampled from identical beta distributions, instead of the logit-normal distribution, and does not allow individual covariates for modelling observed heterogeneity. I shall follow the notation used by Otis et al. (1978) and denote the model I consider by $\mathcal{M}_h$ throughout this chapter for convenience.

Key mathematical notations are summarized in Table 2.1. Based on these notations, the framework of $\mathcal{M}_h$ is defined as follows. The values, $y_{i.} = \sum_{t=1}^{T} y_{it}$ for $i = 1, ..., n$, are the observations of the response variables for $\mathcal{M}_h$. If the capture probability is constant over time, then the i-th response variable follows the binomial distribution with the number of trials set as T and the success probability, p_i. The density function of the response variable is given by

$$f(y_{i.}|v_i; p_i) = \binom{T}{y_{i.}} p_i^{y_{i.}} (1 - p_i)^{T - y_{i.}}\,, \tag{2.1}$$

where p_i depends on the vector of observed covariates, $\mathbf{x}_i$, and the random effect, v_i. Specifically, I model p_i as

$$\text{logit}(p_i) = \mathbf{x}_i'\boldsymbol{\beta} + v_i\,, \tag{2.2}$$

where $v_i \sim \mathcal{N}(0, \sigma_v)$, and $v_1, ..., v_n$ are independent of each other. When no individual covariates are observed, p_i is alternatively modelled by

$$\text{logit}(p_i) = \alpha + v_i\,. \tag{2.3}$$

Modelling p_i by equation (2.2) and (2.3) indicates that $\text{logit}(p_i) \sim \mathcal{N}(\mathbf{x}_i'\boldsymbol{\beta}, \sigma_v)$ and $\text{logit}(p_i) \sim \mathcal{N}(\alpha, \sigma_v)$. Therefore, the framework of $\mathcal{M}_h$ correctly assumes that the capture probabilities follow a logit-normal distribution. Except as noted otherwise, I use $\mathcal{M}_h$ to refer to the more general model for p_i in equation (2.2).

2.2.2 Conditional h-likelihood

Conditioning on the individuals captured at least once, I first define the complete-data likelihood of $\mathcal{M}_h$, which is

$$\begin{aligned}\mathcal{L}(\boldsymbol{\beta}, \sigma_v; \mathbf{y}_., \mathbf{v}) &= \prod_{i=1}^{n} f(y_{i.}|y_{i.} > 0; p_i) \times f(v_i; \sigma_v) \\ &= \prod_{i=1}^{n} \left[\frac{\binom{T}{y_{i.}} p_i^{y_{i.}} (1 - p_i)^{T - y_{i.}}}{\pi_i} \right] \times \left[\frac{1}{\sigma_v \sqrt{2\pi}} \exp\left(- \frac{v_i^2}{2\sigma_v^2} \right) \right],\end{aligned} \tag{2.4}$$

where $\pi_i = 1 - (1 - p_i)^T$ is the probability that individual i is captured at least once. The previous methods for obtaining parameter estimates are based on the marginal likelihood

$$\mathcal{L}(\boldsymbol{\beta}, \sigma_v; \mathbf{y}_.) = \int_{\mathcal{R}^n} \mathcal{L}(\boldsymbol{\beta}, \sigma_v; \mathbf{y}_., \mathbf{v}) d\mathbf{v}\,, \tag{2.5}$$

in which quadrature or sampling methods are necessary for the integration. In the h-likelihood approach, $\boldsymbol{\beta}$, σ_v and $\mathbf{v}$ are estimated by maximizing equation (2.4), which satisfies the definition of the h-likelihood with the canonical scale, described in Chapter 1. According to Yee et al. (2015), the first term of the product in equation (2.4) is identical to the conditional likelihood in equation (1.1) for $\mathcal{M}_h$ and can be regarded as the likelihood of a GLM with the following components: the observed response variables, $y_{i.}$ for $i = 1, ..., n$, that are greater than 0, the linear predictor, $\eta_i = \text{logit}(p_i)$, and the density of $y_{i.}$ in form of equation (1.3) with letting

$$\begin{aligned} y_i &= y_{i.} \\ \theta_i &= \eta_i \\ t(y_i) &= y_{i.} \\ b(\theta_i) &= -T\log(1 - p_i) + \log(\pi_i) \\ \phi &= 1 \end{aligned}$$

and

$$c(y_i, \phi) = \log\binom{T}{y_{i.}}$$

(the notations in the left-hand sides of the equations are described in Chapter 1). In consequence, the complete-data likelihood of $\mathcal{M}_h$ corresponds to the h-likelihood of a GLMM that extends this GLM to the random effects $\mathbf{v}$ through η_i. I call the h-likelihood built on the conditional likelihood as the conditional h-likelihood and denote it by $\mathcal{H}_c(\cdot;\cdot)$. The conditional h-likelihood of $\mathcal{M}_h$ is then given by

$$\mathcal{H}_c(\boldsymbol{\theta}; \mathbf{y}_{.}, \mathbf{v}) = \mathcal{L}(\boldsymbol{\beta}, \sigma_v; \mathbf{y}_{.}, \mathbf{v}), \tag{2.6}$$

where $\boldsymbol{\theta}$ is the vector including all unknown quantities. Additionally, the log of the conditional h-likelihood is denoted by h_c.

2.2.3 Bias correction for MHLEs

According to Lee (2001) and Lee et al. (2017), MHLEs (i.e., estimates obtained by directly maximizing the h-likelihood) can be severely biased. This bias occurs particularly when the sample space of the response variable is very restricted, including the case for $\mathcal{M}_h$ having the sample space $\{1, \ldots, T\}$. Specifically, when the sample space has a few discrete components, then the range of MHLEs, depending on data, will have a restricted range of possible values or a discrete space including only a few distinct points in the worst case. The restricted or discrete range of the MHLEs for random effects affects MHLEs for all unknown quantities if the distributions of the random effects have supports on a continuous space, such as $\mathbb{R}$. I explain the bias in the MHLE for $\boldsymbol{\theta}$ by the following rationales:

1. First, the MHLEs for v_i are biased due to the restricted range of possible estimates, which is not enough to cover the continuous support $\mathbb{R}$.

2. As a result, the MHLEs for $\boldsymbol{\beta}$ and σ_v will be biased as they are computed conditional on the MHLEs for v_i from $\mathcal{H}_c$.

My rationale in Step 1 is supported by the normal equation, where the MHLE for $\mathbf{v}$ is computed by solving

$$\frac{\partial h_c}{\partial \mathbf{v}} = \mathbf{y}_{\cdot} - (T\mathbf{1}_n \odot \mathbf{p}) \oslash \boldsymbol{\pi} - \frac{1}{\sigma_v^2}\mathbf{v} = \mathbf{0}_n\,,$$

in which $\mathbf{y}$ and $\mathbf{x}_i$, determining $\mathbf{p}$ and $\boldsymbol{\pi}$, are only the variables. Hence, if $\mathbf{x}_i$ has a narrow range or is discrete, or T is small, the range of the solution for every v_i is restricted to fall in a small number of narrow intervals in $\mathbb{R}$ or discrete, which cannot cover the support of the distribution of v_i, $\mathbb{R}$, sufficiently.

To come up with a solution to correct the bias, I apply the bias correction proposed by Yun and Lee (2004), based on maximizing the adjusted profile h-likelihood (APHL), introduced by Lee and Nelder (1996). The method corrects the bias through the following estimation steps separated:

Step 1 (fixed effects estimation) Given $\hat{\mathbf{v}}$ and $\hat{\sigma}_v$, $\boldsymbol{\beta}$ is estimated by maximizing the APHL.

Step 2 (random effects estimation) Given $\hat{\boldsymbol{\beta}}$ from Step 1 and $\hat{\sigma}_v$, $\mathbf{v}$ is estimated by maximizing the conditional h-likelihood.

Step 3 (dispersion parameters estimation) Given $\hat{\boldsymbol{\beta}}$ from Step 1 and $\hat{\mathbf{v}}$ from Step 2, σ_v is estimated by maximizing the APHL.

Step 4 Iterate Steps 1 - 3 until convergence.

In detail, following Lee and Nelder (1996), I define the APHLs in Steps 1 and 3 by

$$h_c^A(\boldsymbol{\beta}; \mathbf{y}_{\cdot}, \hat{\mathbf{v}}, \hat{\sigma}_v) = h_c - \frac{1}{2}\log\left[\det\left(-\frac{1}{2\pi}\frac{\partial^2 h_c}{\partial \mathbf{v}\partial \mathbf{v}'}\right)\right]\Bigg|_{\mathbf{v}=\hat{\mathbf{v}},\sigma_v=\hat{\sigma}_v} \tag{2.7}$$

and

$$h_c^A(\sigma_v; \mathbf{y}_{\cdot}, \hat{\mathbf{v}}, \hat{\boldsymbol{\beta}}) = h_c - \frac{1}{2}\log\left[\det\left(-\frac{1}{2\pi}\frac{\partial^2 h_c}{\partial \mathbf{v}\partial \mathbf{v}'}\right)\right]\Bigg|_{\boldsymbol{\delta}=\hat{\boldsymbol{\delta}}}, \tag{2.8}$$

where equation (2.7) is regarded as a function of $\boldsymbol{\beta}$, and equation (2.8) is regarded as that of σ_v. It was shown by Lee and Nelder (1996) that the APHL is equal to the marginal likelihood in

Algorithm 1 Fitting algorithm for $\mathcal{M}_h$.

```
Set initial value $\hat{\boldsymbol{\theta}}^{(0)} = (\hat{\boldsymbol{\delta}}'^{(0)}, \hat{\sigma}_v^{(0)})'$
Let $r = 0$
while convergence criterion $\hat{\boldsymbol{\theta}}^{(r)} \approx \hat{\boldsymbol{\theta}}^{(r+1)}$ not met do
    Let $\hat{\boldsymbol{\delta}}^{(r,0)} = \hat{\boldsymbol{\delta}}^{(r)}$;
    Let $t = 0$;
    while convergence criterion $\hat{\boldsymbol{\delta}}^{(r,t)} \approx \hat{\boldsymbol{\delta}}^{(r,t+1)}$ not met do
        (Step 1) Given $\hat{\sigma}_v^{(r)}$, solve equation (2.9) for $\hat{\boldsymbol{\delta}}^{(r,t)}$;
        $t \leftarrow t + 1$
    end while
    Let $\hat{\boldsymbol{\delta}}^{(r+1)} = \hat{\boldsymbol{\delta}}^{(r,t)}$
    (Step 2) Given $\hat{\boldsymbol{\delta}}^{(r+1)}$, obtain $\hat{\sigma}_v^{(r+1)}$ by fitting the gamma GLM with
      • observed response variables: $\hat{\mathbf{v}}^{(r+1)} = (\hat{v}_1^{(r+1)}, ..., \hat{v}_n^{(r+1)})'$
      • prior weight: $q_i$ in equation (2.10) for $\hat{v}_i^{(r+1)}$, given $\hat{\boldsymbol{\delta}}^{(r+1)}$ and $\hat{\sigma}_v^{(r)}$
      • linear predictor: $\tau = \log(\sigma_v^2)$
end while
```

which the integral in equation (2.5) is approximated by the first-order Laplace's method, and the estimation of dispersion parameter in GLMMs is separable from that of fixed effects (e.g., $\partial h_c^A / \partial \boldsymbol{\beta} \partial \sigma_v \approx 0$). Hence, the parameters estimated from equations (2.7) and (2.8) are expected to be close to those obtained from the marginal likelihood in equation (2.5). These APHLs do not estimate v_i along with $\boldsymbol{\beta}$ and σ_v, and so the bias in the raw MHLEs for $\boldsymbol{\beta}$ and σ_v, caused by the first rationale, can be reduced considerably by replacing them with $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}_v$ obtained from the APHLs.

2.2.4 Fitting algorithm

I derive a fitting algorithm for obtaining bias-corrected estimates of parameters and random effects, $\hat{\boldsymbol{\theta}}$, based on the three main steps using the APHLs and the conditional h-likelihood. The fitting algorithm is extended from the algorithm provided by Noh and Lee (2007), who considered the same bias correction and the framework of GLMMs with binary data, but not the framework of any MR models such as $\mathcal{M}_h$. The algorithm performs the three main steps by the iterative re-weighted least squares (IRLS), while combining Steps 1 and 2 to reduce computation time.

In Algorithm 1, I illustrate the details of the algorithm for fitting $\mathcal{M}_h$. Step 1 of the algorithm solves two normal equations,

$$\frac{\partial h_c^A}{\partial \boldsymbol{\beta}} = \mathbf{X}'(\mathbf{y}_. - \boldsymbol{\mu} - \mathbf{s}) = \mathbf{0}_{\dim(\boldsymbol{\beta})}$$

with respect to $\boldsymbol{\beta}$, and

$$\frac{\partial h_c}{\partial \mathbf{v}} = (\mathbf{y}_. - \boldsymbol{\mu}) - \frac{1}{\sigma_v^2}\mathbf{v} = \mathbf{0}_n$$

with respect to $\mathbf{v}$, of which the equation for the IRLS,

$$\mathbf{T}'\mathbf{W}^{(t-1)}\mathbf{T}\hat{\boldsymbol{\delta}}^{(t)} = \mathbf{T}'\mathbf{W}^{(t-1)}\mathbf{z}^{*(t-1)}\,, \tag{2.9}$$

is derived after some re-arrangements. In this equation, $\boldsymbol{\delta}^{(t)} = \hat{\boldsymbol{\delta}}$ at the t-th iteration,

$$\mathbf{T} = \begin{pmatrix} \mathbf{X} & \mathbf{I}_n \\ \mathbf{0}_{n\times\dim(\boldsymbol{\beta})} & \mathbf{I}_n \end{pmatrix}$$

is the the design matrix such that $(\boldsymbol{\eta}', \mathbf{v}')' = \mathbf{T}\boldsymbol{\delta}$ with $\boldsymbol{\eta} = (\eta_1, ..., \eta_n)' = (\text{logit}(p_1), ..., \text{logit}(p_n))$,

$$\mathbf{W}^{(t)} = \left.\begin{pmatrix} \text{diag}\left(\frac{\partial \mu_1}{\partial \eta_1}, ..., \frac{\partial \mu_n}{\partial \eta_n}\right) & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \frac{1}{\sigma_v^2}\mathbf{I}_n \end{pmatrix}\right|_{\boldsymbol{\delta}=\hat{\boldsymbol{\delta}}^{(t)}, \sigma_v=\hat{\sigma}_v}$$

is the t-th weight matrix, and

$$\mathbf{z}^{*(t)} = \left.\begin{pmatrix} \boldsymbol{\eta} + \text{diag}\left(\frac{\partial \mu_1}{\partial \eta_1}, ..., \frac{\partial \mu_n}{\partial \eta_n}\right)(\mathbf{y}_. - \boldsymbol{\mu} - \mathbf{s}) \\ \mathbf{s} \end{pmatrix}\right|_{\boldsymbol{\delta}=\hat{\boldsymbol{\delta}}^{(t)}, \sigma_v=\hat{\sigma}_v}$$

is the t-th adjusted response variable of the IRLS equation. For the terms within $\mathbf{T}$, $\mathbf{W}^{(t)}$ and $\mathbf{z}^{*(t)}$, I write

$$\boldsymbol{\mu} = E(\mathbf{y}) = (\mu_1, ..., \mu_n)'$$

and

$$\mathbf{s} = (s_1', ..., s_n')'\,,$$

and have derived that

$$\mu_i = \frac{Tp_i}{\pi_i}$$

and

$$s_i = \frac{1}{2}\Big(- \frac{\partial^2 h_c}{\partial v_i^2}\Big)^{-1}\Big(\frac{\partial^2 \mu_i}{\partial \eta_i^2} + \frac{\partial^2 \mu_i}{\partial \eta_i \partial v_i}\Big|_{v_i=\hat{v}_i} \frac{\partial \hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial \eta_i}\Big),$$

where $\hat{v}_i(\boldsymbol{\beta}, \sigma_v)$ is the MHLE for v_i given fixed values of $\boldsymbol{\beta}$ and σ_v,

$$\frac{\partial \mu_i}{\partial \eta_i} = \frac{T p_i(1-p_i)}{\pi_i} - \frac{T^2 p_i^2 (1-\pi_i)}{\pi_i^2},$$

$$\begin{aligned}\frac{\partial^2 \mu_i}{\partial \eta_i^2} = \frac{\partial^2 \mu_i}{\partial \eta_i \partial v_i} = {} & \frac{T p_i(1-p_i)}{\pi_i} - \frac{T^2 p_i^2(1-\pi_i)}{\pi_i^2} - \frac{2T p_i^2(1-p_i)}{\pi_i} + \frac{T^2 p_i^3(1-\pi_i)}{\pi_i^2} \\ & - \frac{2T^2 p_i^2(1-p_i)}{\pi_i^2} + \frac{2T^3 p_i^3(1-\pi_i)}{\pi_i^3} + \frac{2T^2 p_i^2(1-p_i)}{\pi_i} - \frac{T^3 p_i^3(1-\pi_i)}{\pi_i^2},\end{aligned}$$

and

$$\frac{\partial \hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial \eta_i} = \Big(- \frac{\partial^2 h_c}{\partial v_i^2}\Big)^{-1} \frac{\partial^2 h_c}{\partial v_i \partial \eta_i}$$

in which

$$-\frac{\partial^2 h_c}{\partial v_i^2} = \frac{\partial \mu_i}{\partial \eta_i} + \frac{1}{\sigma_v^2}$$

and

$$\frac{\partial^2 h_c}{\partial v_i \partial \eta_i} = -\frac{\partial \mu_i}{\partial \eta_i}.$$

The IRLS algorithm repeatedly solves equation (2.9) about $\hat{\boldsymbol{\delta}}^{(t)}$ until convergence is achieved for the solution of equation (2.9). In Step 2 of the algorithm, the normal equation

$$\frac{\partial h_c^A}{\partial \tau} = \sum_{i=1}^{n} \frac{\partial \sigma_v^2}{\partial \tau} \frac{v_i - (1-q_i)\sigma_v^2}{2\sigma_v^4} = 0$$

is solved as a function of $\tau = \log(\sigma_v^2)$, where

$$q_i = \Big(- \frac{\partial^2 h_c}{\partial v_i^2}\Big)^{-1}\Big(\frac{\partial^2 \mu_i}{\partial \eta_i \partial v_i}\Big|_{v_i=\hat{v}_i} \frac{\partial \hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial \tau} + \frac{\partial^2 h_c}{\partial v_i \partial \tau}\Big) \tag{2.10}$$

in which

$$\frac{\partial \hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial \tau} = \Big(- \frac{\partial^2 h_c}{\partial v_i^2}\Big)^{-1} \frac{\partial^2 h_c}{\partial v_i \partial \tau}$$

and

$$\frac{\partial^2 h_c}{\partial v_i \partial \tau} = \frac{1}{\sigma_v^2}.$$

The normal equation is equivalent to that of a gamma GLM (i.e., an ordinary GLM with the response variable following a gamma distribution) such that the response variables are observed as v_i, each of which has the prior weight, q_i, and linear predictor, $\tau = \log(\sigma_v^2)$. This model can be fit easily by using existing routines in most modern software, and the estimate $\hat{\tau}$ can then be backtransformed to $\hat{\sigma}_v = \exp(\hat{\tau})$. The general IRLS formula for ordinary GLMs can be found in the work of McCullagh and Nelder (2019).

To compute the variance–covariance matrix for $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$, I apply the ML-like properties of the h-likelihood, as described in Chapter 1. According to equation (1.13), the covariance-variance matrix for $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ is given by

$$\text{Var}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) = \begin{pmatrix} \text{Var}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) & \mathbf{0}_{\text{dim}(\boldsymbol{\delta})} \\ \mathbf{0}'_{\text{dim}(\boldsymbol{\delta})} & \text{Var}(\hat{\sigma}_v) \end{pmatrix},$$

where I have derived that

$$\text{Var}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) \approx (\mathbf{T}'\mathbf{W}^*\mathbf{T})^{-1}|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}.$$

I have computed $\text{Var}(\hat{\sigma}_v)$ by applying the delta method, such that $\text{Var}(\hat{\sigma}_v) \approx \sigma_v^2/4 \times \text{Var}(\hat{\tau})$, and obtained $\text{Var}(\hat{\tau})$ directly from the IRLS algorithm for fitting the gamma GLM as described above. When the population size is estimated (Section 2.2.5), I have need of the variance of $\hat{\mathbf{v}}(\boldsymbol{\beta}, \sigma_v)$ and approximated it by

$$\text{Var}(\hat{\mathbf{v}}(\boldsymbol{\beta}, \sigma_v) - \mathbf{v}) \approx \left(-\frac{\partial^2 h_c}{\partial \mathbf{v} \partial \mathbf{v}'}\right)^{-1} \approx \left[\text{diag}\left(\frac{\partial \mu_1}{\partial \eta_1}, ..., \frac{\partial \mu_n}{\partial \eta_n}\right) + \frac{1}{\sigma_v^2}\mathbf{I}_{n \times n}\right]^{-1}\Bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}.$$

These covariance matrices can be used to draw Wald-typed inference through the asymptotic properties of the h-likelihood,

$$\hat{\boldsymbol{\delta}} - \boldsymbol{\delta} \mathrel{\dot{\sim}} \mathcal{N}(\mathbf{0}_{\text{dim}(\boldsymbol{\delta})}, \text{Var}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})) \tag{2.11}$$

and

$$\mathbf{v}|\mathbf{y}_{.} \mathrel{\dot{\sim}} \mathcal{N}(\hat{\mathbf{v}}(\boldsymbol{\beta}, \sigma_v), \text{Var}(\hat{\mathbf{v}}(\boldsymbol{\beta}, \sigma_v) - \mathbf{v})), \tag{2.12}$$

as shown by Lee and Nelder (1996), if the sample size, n, is large enough. I shall use these asymptotic properties to make inference for the population size, N.

2.2.5 Estimation of population size

The method I propose to estimate N is based on the HT estimator in equation (1.2). Given $\boldsymbol{\delta}$, I rewrite the HT estimator as

$$\hat{N}(\boldsymbol{\delta}) = \sum_{i=1}^{N} \frac{c_i}{\pi_i} = \sum_{i=1}^{n} \frac{1}{\pi_i}\,,$$

where $c_i = I(y_{i.} > 0)$. If $\mathbf{v}$ is not considered in the model (i.e., unobserved heterogeneity is not modelled), then as $\boldsymbol{\beta}$ is unknown in practice, Huggins (1989) suggested the idea that the HT estimator is approximated by substituting $\hat{\boldsymbol{\beta}}$. My method extends this idea to $\boldsymbol{\delta}$ and initially considers the estimator

$$\hat{N}(\hat{\boldsymbol{\delta}}) = \sum_{i=1}^{n} \frac{1}{\hat{\pi}_i}\,,$$

where $\hat{\boldsymbol{\delta}}$ is obtained by maximizing equation (2.6) as in Algorithm 1.

Practically, I have found that $\hat{N}(\hat{\boldsymbol{\delta}})$ underestimates the true value of N, typically when the values of p_i are close to 0. To show the bias in $\hat{N}(\hat{\boldsymbol{\delta}})$ and derive an unbiased estimator by correcting this bias, I expand $\hat{N}(\hat{\boldsymbol{\delta}})$ about $\boldsymbol{\delta}$ by the first-order Taylor series, which results in

$$\hat{N}(\hat{\boldsymbol{\delta}}) \approx \hat{N}(\boldsymbol{\delta}) + \frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}'}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})\,, \tag{2.13}$$

and expand $\hat{N}(\boldsymbol{\delta})$ and $\frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}'}$ about $\tilde{\boldsymbol{\delta}} = (\boldsymbol{\beta}, \hat{v}(\boldsymbol{\delta}, \sigma_v))$ by the same method as well, which brings about

$$\hat{N}(\boldsymbol{\delta}) \approx \hat{N}(\tilde{\boldsymbol{\delta}}) + \frac{\partial \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'}(\boldsymbol{\delta} - \tilde{\boldsymbol{\delta}}) \tag{2.14}$$

and

$$\frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}'} \approx \frac{\partial \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'} + (\boldsymbol{\delta} - \tilde{\boldsymbol{\delta}})' \frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}' \partial \tilde{\boldsymbol{\delta}}}\,. \tag{2.15}$$

Applying the expectation to the above equations, conditioning on c_i for $i = 1, ..., N$ and $\boldsymbol{\delta}$ only for equation (2.13), I obtain

$$E(\hat{N}(\hat{\boldsymbol{\delta}})|\mathbf{c}, \boldsymbol{\delta}) \approx \hat{N}(\boldsymbol{\delta}) + \frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}'}(\tilde{\boldsymbol{\delta}} - \boldsymbol{\delta}) \tag{2.16}$$

$$E(\hat{N}(\boldsymbol{\delta})|\mathbf{c}) \approx \hat{N}(\tilde{\boldsymbol{\delta}}) \tag{2.17}$$

and

$$E\left(\frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}'}\middle|\mathbf{c}\right) \approx \frac{\partial \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'}\,, \tag{2.18}$$

in which the properties that $E(\hat{\boldsymbol{\delta}}|\boldsymbol{\delta}) \approx \tilde{\boldsymbol{\delta}}$ and $E(\boldsymbol{\delta}) \approx \tilde{\boldsymbol{\delta}}$ from equations (2.11) and (2.12) are applied. Additionally, I obtain the following expectation from equation (2.15),

$$\begin{aligned} E\Big(\frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}'}\boldsymbol{\delta}\Big|\mathbf{c}\Big) &\approx \frac{\partial \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}}\tilde{\boldsymbol{\delta}} + E\Big(\boldsymbol{\delta}'\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'\partial \tilde{\boldsymbol{\delta}}}\boldsymbol{\delta}\Big|\mathbf{c}\Big) - \tilde{\boldsymbol{\delta}}'\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'\partial \tilde{\boldsymbol{\delta}}}\tilde{\boldsymbol{\delta}} \\ &= \frac{\partial \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}}\tilde{\boldsymbol{\delta}} + tr\Big(\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'\partial \tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\Big), \end{aligned} \tag{2.19}$$

required in the expectation in equation (2.21). Note that the final manipulation depends on the identity

$$E(\boldsymbol{\epsilon}'\Lambda\boldsymbol{\epsilon}) = tr(\Lambda \mathrm{Var}(\boldsymbol{\epsilon})) + E(\boldsymbol{\epsilon})'\Lambda E(\boldsymbol{\epsilon}) \tag{2.20}$$

for a random vector $\boldsymbol{\epsilon}$ and a known matrix Λ, as shown by Mathai and Provost (1992, pg.50). The application of the expectation about $\boldsymbol{\delta}$ to equations (2.16) and substitution of equations (2.17), (2.18) and (2.19) into this expectation leads to

$$\begin{aligned} E_{\boldsymbol{\delta}}(E(\hat{N}(\hat{\boldsymbol{\delta}})|\mathbf{c},\boldsymbol{\delta})) &= E(\hat{N}(\hat{\boldsymbol{\delta}})|\mathbf{c}) \\ &\approx E(\hat{N}(\boldsymbol{\delta})|\mathbf{c}) + E\Big(\frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}'}\tilde{\boldsymbol{\delta}}\Big|\mathbf{c}\Big) - E\Big(\frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}'}\boldsymbol{\delta}\Big|\mathbf{c}\Big) \\ &= \hat{N}(\tilde{\boldsymbol{\delta}}) - tr\Big(\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'\partial \tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\Big). \end{aligned} \tag{2.21}$$

As Huggins (1989) showed that $E(\hat{N}(\tilde{\boldsymbol{\delta}})) \approx N$ in equation (2.21), it is true that

$$\begin{aligned} E(E(\hat{N}(\hat{\boldsymbol{\delta}})|\mathbf{c})) &= E(\hat{N}(\hat{\boldsymbol{\delta}})) \\ &\approx N - tr\Big(\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'\partial \tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\Big), \end{aligned}$$

and this result implies that $\hat{N}(\hat{\boldsymbol{\delta}})$ has negative bias $tr\Big(\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'\partial \tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\Big)$. Hence, I propose the new, approximately unbiased estimator

$$\hat{N}^{\star}(\hat{\boldsymbol{\delta}}) = \sum_{i=1}^{n}\frac{1}{\hat{\pi}_i} + tr\Big(\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'\partial \tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\Big)\Big|_{\boldsymbol{\delta}=\hat{\boldsymbol{\delta}}},$$

where

$$\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\delta}})}{\partial \tilde{\boldsymbol{\delta}}'\partial \tilde{\boldsymbol{\delta}}} = -\sum_{i=1}^{n} T p_i(1-\pi_i)\Big[\frac{(1-p_i)}{\pi_i^2} + T p_i\Big(\frac{1}{\pi_i^2} - \frac{2}{\pi^3}\Big)\Big]\mathbf{x}_i\mathbf{x}_i'$$

and

$$\mathrm{Var}(\boldsymbol{\delta}) = \begin{pmatrix} \mathbf{0}_{\dim(\boldsymbol{\beta})\times\dim(\boldsymbol{\beta})} & \mathbf{0}_{\dim(\boldsymbol{\beta})\times n} \\ \mathbf{0}_{n\times\dim(\boldsymbol{\beta})} & \mathrm{Var}(\hat{\mathbf{v}}(\boldsymbol{\beta},\sigma_v)) \end{pmatrix}.$$

I found in my simulation study that this unbiased estimator, $N^{\star}(\hat{\delta})$, more accurately estimates N than the naive estimator, $\hat{N}(\hat{\boldsymbol{\delta}})$, in general.

To compute the variance of $\hat{N}^{\star}(\hat{\boldsymbol{\delta}})$, and so construct a confidence interval (CI) for N, I employ the law of iterated expectations, which indicates that

$$\begin{aligned}
\mathrm{Var}(\hat{N}^{\star}(\hat{\boldsymbol{\delta}})) &= E(\mathrm{Var}(\hat{N}^{\star}(\hat{\boldsymbol{\delta}})|\mathbf{c})) + \mathrm{Var}(E(\hat{N}^{\star}(\hat{\boldsymbol{\delta}})|\mathbf{c})) \\
&= E_{\mathbf{c}}[E_{\boldsymbol{\delta}}(\mathrm{Var}(\hat{N}^{\star}(\hat{\boldsymbol{\delta}})|\mathbf{c},\boldsymbol{\delta}))] + E_{\mathbf{c}}[\mathrm{Var}_{\boldsymbol{\delta}}(E(\hat{N}^{\star}(\hat{\boldsymbol{\delta}})|\mathbf{c},\boldsymbol{\delta}))] \\
&\quad + \mathrm{Var}(E(\hat{N}^{\star}(\hat{\boldsymbol{\delta}})|\mathbf{c}))\,.
\end{aligned}$$

In this equation, I substitute

$$\begin{aligned}
1)\ E_{\mathbf{c}}[E_{\boldsymbol{\delta}}(\mathrm{Var}(\hat{N}^{\star}(\hat{\boldsymbol{\delta}})|\mathbf{c},\boldsymbol{\delta}))] &\approx E_{\mathbf{c}}\left[E_{\boldsymbol{\delta}}\left(\frac{\partial\hat{N}(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}'}\mathrm{Var}(\hat{\boldsymbol{\delta}}-\boldsymbol{\delta})\frac{\partial\hat{N}(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}}\right)\right] \\
&\approx \frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'}\mathrm{Var}(\hat{\boldsymbol{\delta}}-\boldsymbol{\delta})\frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}} + tr\Big(\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\hat{\boldsymbol{\delta}}-\boldsymbol{\delta})\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\Big),
\end{aligned} \tag{2.22}$$

obtained by computing the variance of the expression in equation (2.13), applying the expectation in equation (2.15) and the formula in equation (2.20);

$$\begin{aligned}
2)\ E_{\mathbf{c}}[\mathrm{Var}_{\boldsymbol{\delta}}(E(\hat{N}^{\star}(\hat{\boldsymbol{\delta}})|\mathbf{c},\boldsymbol{\delta}))] &\approx E_{\mathbf{c}}\left[\mathrm{Var}_{\boldsymbol{\delta}}(\hat{N}(\boldsymbol{\delta})) + \tilde{\boldsymbol{\delta}}'\mathrm{Var}_{\boldsymbol{\delta}}\left(\frac{\partial\hat{N}(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}'}\right)\tilde{\boldsymbol{\delta}} + \mathrm{Var}_{\boldsymbol{\delta}}\left(\frac{\partial\hat{N}(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}'}\boldsymbol{\delta}\right)\right] \\
&= E_{\mathbf{c}}[\mathrm{Var}(\hat{N}(\boldsymbol{\delta})|\mathbf{c})] + E_{\mathbf{c}}\left[\tilde{\boldsymbol{\delta}}'\mathrm{Var}\left(\frac{\partial\hat{N}(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}'}\bigg|\mathbf{c}\right)\tilde{\boldsymbol{\delta}}\right] + E_{\mathbf{c}}\left[\mathrm{Var}\left(\frac{\partial\hat{N}(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}'}\boldsymbol{\delta}\bigg|\mathbf{c}\right)\right]\,,
\end{aligned} \tag{2.23}$$

derived by computing the variance of the expression in equation (2.13), where

$$E_{\mathbf{c}}[\mathrm{Var}(\hat{N}(\boldsymbol{\delta})|\mathbf{c})] \approx \frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'}\mathrm{Var}(\boldsymbol{\delta})\frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}}$$

$$E_{\mathbf{c}}\left[\tilde{\boldsymbol{\delta}}'\mathrm{Var}\left(\frac{\partial\hat{N}(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}'}\bigg|\mathbf{c}\right)\tilde{\boldsymbol{\delta}}\right] \approx \tilde{\boldsymbol{\delta}}'\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\tilde{\boldsymbol{\delta}}$$

and

$$
\begin{aligned}
E_{\mathbf{c}}\left[\mathrm{Var}\left(\frac{\partial\hat{N}(\boldsymbol{\delta})}{\partial\boldsymbol{\delta}'}\boldsymbol{\delta}\middle|\mathbf{c}\right)\right] \approx{}& \frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'}\mathrm{Var}(\boldsymbol{\delta})\frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}} \\
&+ \mathrm{Var}\left(\boldsymbol{\delta}'\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\boldsymbol{\delta}\right) + \tilde{\boldsymbol{\delta}}'\frac{\partial^2\hat{N}(\boldsymbol{\delta})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\frac{\partial^2\hat{N}(\boldsymbol{\delta})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\tilde{\boldsymbol{\delta}} \\
={}& \frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'}\mathrm{Var}(\boldsymbol{\delta})\frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}} \\
&+ 2tr\left(\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\right) + 5\tilde{\boldsymbol{\delta}}'\frac{\partial^2\hat{N}(\boldsymbol{\delta})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\frac{\partial^2\hat{N}(\boldsymbol{\delta})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\tilde{\boldsymbol{\delta}},
\end{aligned}
$$

obtained by computing the variance of the expressions in equations (2.14) and (2.15), and applying the formula,

$$
\mathrm{Var}(\boldsymbol{\epsilon}'\Lambda\boldsymbol{\epsilon}) = 2tr(\Lambda\mathrm{Var}(\boldsymbol{\epsilon})\Lambda\mathrm{Var}(\boldsymbol{\epsilon})) + 4E(\boldsymbol{\epsilon})'\Lambda\mathrm{Var}(\boldsymbol{\epsilon})\Lambda E(\boldsymbol{\epsilon}),
$$

as given by Mathai and Provost (1992, p.76); and

$$
\begin{aligned}
3)\ \mathrm{Var}(E(\hat{N}^{\star}(\hat{\boldsymbol{\delta}})|\mathbf{c})) &\approx \mathrm{Var}(\hat{N}(\boldsymbol{\delta})) \\
&\approx \sum_{i=1}^{n}\hat{\pi}_i^{-2} - \hat{\pi}_i^{-1},
\end{aligned}
\tag{2.24}
$$

as shown by Huggins (1989). In consequence, the final form of $\mathrm{Var}(\hat{N}^{\star}(\hat{\boldsymbol{\delta}}))$ is

$$
\begin{aligned}
\mathrm{Var}(\hat{N}^{\star}(\hat{\boldsymbol{\delta}})) \approx{}& \frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'}\mathrm{Var}(\hat{\boldsymbol{\delta}}-\boldsymbol{\delta})\frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}} + 2\frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'}\mathrm{Var}(\boldsymbol{\delta})\frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}} \\
&+ 6\tilde{\boldsymbol{\delta}}'\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\tilde{\boldsymbol{\delta}} + 2tr\left(\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\right) \\
&+ tr\left(\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\hat{\boldsymbol{\delta}}-\boldsymbol{\delta})\frac{\partial^2\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}'\partial\tilde{\boldsymbol{\delta}}}\mathrm{Var}(\boldsymbol{\delta})\right) + \sum_{i=1}^{n}\frac{1}{\pi_i^2} - \frac{1}{\pi_i},
\end{aligned}
$$

where

$$
\frac{\partial\hat{N}(\tilde{\boldsymbol{\delta}})}{\partial\tilde{\boldsymbol{\delta}}} = -\sum_{i=1}^{n} Tp_i\left(\frac{1}{\pi_i^2} - \frac{1}{\pi_i}\right)\mathbf{x}_i
$$

and I approximate the variance by substituting $\hat{\boldsymbol{\delta}}$ into $\boldsymbol{\delta}$ in the final form. The variance is used to compute CIs for N, as described below.

I consider two methods of constructing a CI for N, based either on the assumption that N follows a normal distribution or that $N - n$ follows a log-normal distribution. The assumption of the normal distribution is available by the asymptotic property of the HT estimator (i.e.,

$\hat{N}^\star(\hat{\boldsymbol{\delta}}) \dot{\sim} \mathcal{N}(N, \mathrm{Var}(\hat{N}^\star(\hat{\boldsymbol{\delta}})))$), shown by Huggins (1989), and provides Wald CI,

$$\hat{N}^\star(\hat{\boldsymbol{\delta}}) \pm 1.96 \sqrt{\mathrm{Var}(\hat{N}^\star(\hat{\boldsymbol{\delta}}))}\,,$$

when the confidence level is 95%. As an alternative to the Wald CI, Burnham et al. (1987) and Chao (1987) derived a second CI with the assumption of the log-normal distribution for $N - n$,

$$(n + \hat{f}_0/C, n + \hat{f}_0 \times C)\,,$$

where

$$C = \exp\left\{1.96\left[\log\left(1 + \frac{\mathrm{Var}(\hat{N}^\star(\hat{\boldsymbol{\delta}}))}{\hat{f}_0^2}\right)\right]^{1/2}\right\}$$

and $\hat{f}_0 = \hat{N}^\star(\hat{\boldsymbol{\delta}}) - n$. The log-normal CI has been recommended to solve the issue that the sample distribution of N is often right-skewed rather than symmetric (i.e., fails to hold the asymptotic normality of $\hat{N}^\star(\hat{\boldsymbol{\delta}})$). In my simulation study, I found that the Wald CI generally provides good reliability compared to the log-normal CI.

2.3 Simulation Study

I conducted a simulation study to assess the performance of my approach based on the h-likelihood for fitting $\mathcal{M}_h$ and estimating the population size. In particular, I generated $n_{sim} = 1000$ data sets from $\mathcal{M}_h$, in which the linear predictor is defined as equation (2.2), where $\boldsymbol{\beta} = (\alpha, \beta_h)$, and β_h is the effect of a single individual covariate x_i. I considered 32 different scenarios in total by setting α at one of the four values, -2.2, -1.39, -0.85 and -0.41, and σ_v to be one of 0.1, 0.4, 0.7 and 1.0, T, while the combination, (T, N), is set as either (5, 100) and (8, 250), expected to generate two different ranges of sample sizes (low and moderate), respectively. In all scenarios, x_i is generated from $U(-1, 1)$ for each i, and the fixed effect β_h is always set as 0.7. The four divisions of α indicated four different levels of the median capture probability, $\bar{p}_i \approx 0.1$, 0.2, 0.3 and 0.4, corresponding to $\alpha = -2.2$, -1.39, -0.85 and -0.41, respectively.

My expectation for the simulation was that parameters would be more accurately estimated as α increases, which raises p_i values, while the CI based on the log-normal distribution would outperform the Wald CI particularly under the setting of a lower value of α. My reasoning was that the sample size would be affected by the values of α since the values in increasing order provide the medians of π_i, $\bar{\pi}_i \approx 0.57$, 0.83, 0.94 and 0.98, and so a larger number of individuals are captured at least once when α gets larger. If the sample size is large enough, for which scenarios with $\alpha = -0.41$ are the optimal cases, the asymptotic property should hold so that Wald CI should be reliable in general, while the asymptotic property is likely violated when α is small and the Wald CI may not cover the true value of the population size very often. The four values of σ_v will also affect the sample size generated, and I was also interested in observing the influence of increasing of σ_v on the estimation of the remaining parameters.

The performance was assessed by three quantities: relative bias (RB), relative root mean square error (RRMSE), and coverage probability (CP). The RB is computed by

$$RB = \sum_{s=1}^{n_{sim}} \frac{\hat{\theta}_s - \theta}{|\theta|}\,, \tag{2.25}$$

where $\hat{\theta}_s$ is the s-th estimate of a parameter θ. The RRMSE is computed by

$$RRMSE = \sqrt{\frac{1}{n_{sim}} \sum_{s=1}^{n_{sim}} \left(\frac{\hat{\theta}_s - \theta}{\theta} \right)^2}\,. \tag{2.26}$$

The CP was computed for Wald CIs of all parameters estimated from the h-likelihood property as well as the log-normal CI for N. The CP is defined as the proportion of coverage of θ, defined by

$$CP = 100 \times \frac{1}{n_{sim}} \sum_{s=1}^{n_{sim}} I(L_s < \theta_s < U_s)\,, \tag{2.27}$$

where L_s and U_s are the lower and upper bounds of the s-th 95% CI of interest obtained. I computed all CIs with the confidence level set at 95%, and so their CP should be close to 95% if the CIs were adequately reliable.

The results of the RB and RRMSE for each parameter estimate when $(T, N) = (5, 100)$ are illustrated in Figures 2.1 and 2.2, and those when $(T, N) = (8, 250)$ are illustrated in Figures 2.3 and 2.4. Under both settings of (T, N), RB and RRMSE of all parameters have shown similar patterns, but their values are larger in magnitude when $(T, N) = (5, 100)$. As expected, all parameter estimates, except for $\hat{\alpha}$, are estimated more accurately as α increases because their RB and RRMSE get closer to 0 as α increases in general. However, the RB of $\hat{\alpha}$ seems not to depend on α, and even more surprising, the RRMSE of it seems to be larger as α increases. When α is fixed at a value, and σ_v increases, the RB and RRMSE of all parameter estimates, except for $\hat{\sigma}_v$, get farther from 0. Meanwhile, the RB and RRMSE of $\hat{\sigma}_v$ appear to be extremely large if $\alpha = -2.2$ and $\sigma_v = 0.1$. Although these results about $\hat{\alpha}$ and $\hat{\sigma}_v$ are unexpected, the most important estimator, $\hat{N}$, has negligible bias when $\alpha > -2.2$ and the RRMSE is less than 10% when $\alpha > -2.2$, regardless of (T, N) setting.

The results of the CP for log-normal and Wald CIs for N are provided in Table 2.2. Con-

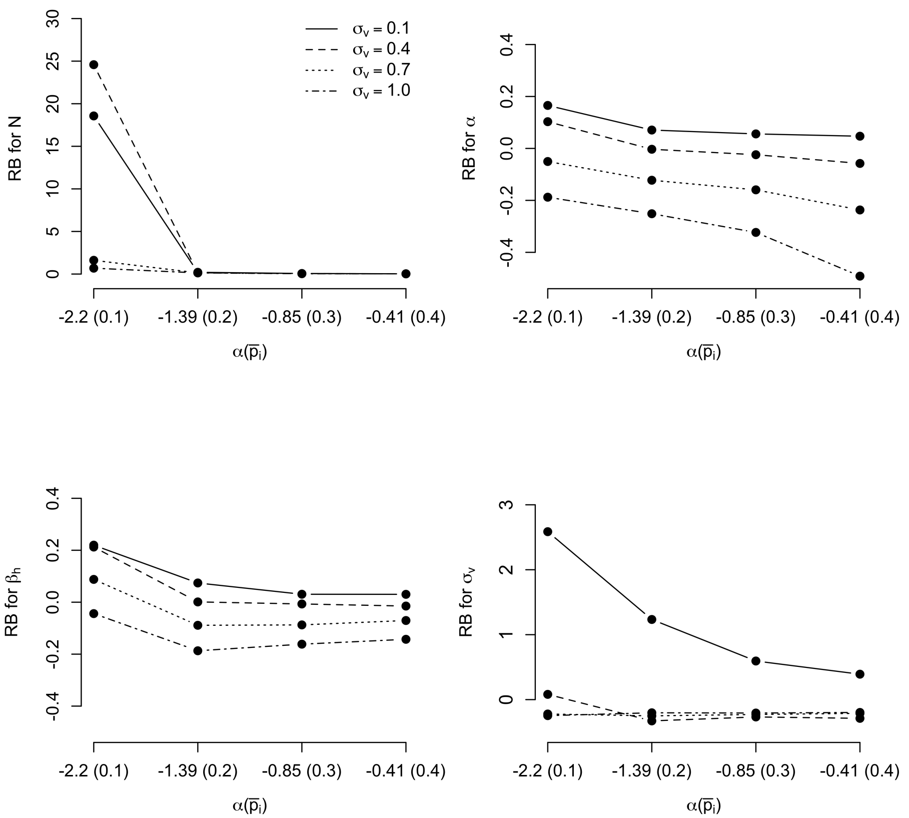

Figure 2.1: Relative Bias (RB) for $\hat{N}$ (upper left), $\hat{\alpha}$ (upper right), $\hat{\beta}_h$ (lower left) and $\hat{\sigma}_v$ (lower right), when $T = 5$, and $N = 100$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values.

trary to my hypothesis, I observed that the Wald CI outperforms the log-normal CI under all scenarios. Specifically, the log-normal CI provided CPs comparable to the Wald CI only when $\sigma_v = 1.0$ but fails to cover N too often when σ_v is set at the other three values. Meanwhile, the CPs of the Wald CI seem to be unaffected by any changes in α or σ_v when $(T, N) = (8, 250)$ and even always close to 95%; however, when $(T, N) = (5, 100)$, as α get lower, the CP gets less than 95% for all settings of σ_v, except the value 0.1. In general, it can be concluded that Wald CI is highly reliable when $(T, N) = (8, 250)$, while underperforming when $(T, N) = (5, 100)$, but is more reliable when compared to the log-normal CI. It is surprising that log-normal CI resulted in low CPs when α is low, which suggests that the distribution of the HT estimator is almost symmetric even though p_i is low.

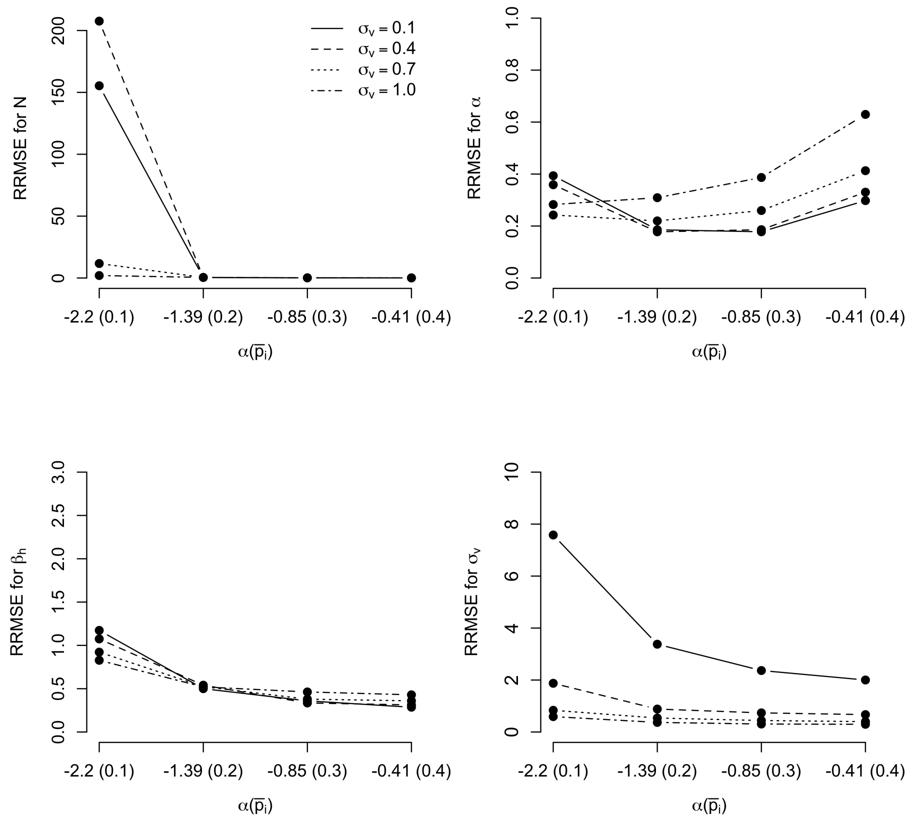

Figure 2.2: Relative root mean square error (RRMSE) for $\hat{N}$ (upper left), $\hat{\alpha}$ (upper right), $\hat{\beta}_h$ (lower left) and $\hat{\sigma}_v$ (lower right), when $T = 5$, and $N = 100$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values.

2.4 Application

The data I have applied my approach to comes from a well-known CR data set for snowshoe hares (*Lepus americanus*), originally collected by Burnham and Cushwa and initially analyzed by Otis et al. (1978). Several previous works, including Coull and Agresti (1999), Royle et al. (2007) and King et al. (2016b), have used the same data to illustrate different methods for fitting $\mathcal{M}_h$ with the same linear predictor defined in equation (2.3). The data includes records for $n = 68$ individuals captured on 9 successive days in the winter of 1972, although only the last $T = 6$ days were considered since no captures were reported in the first three days.

For the purpose of comparing different approaches for fitting $\mathcal{M}_h$ and estimating the population size, I have applied three different methods:

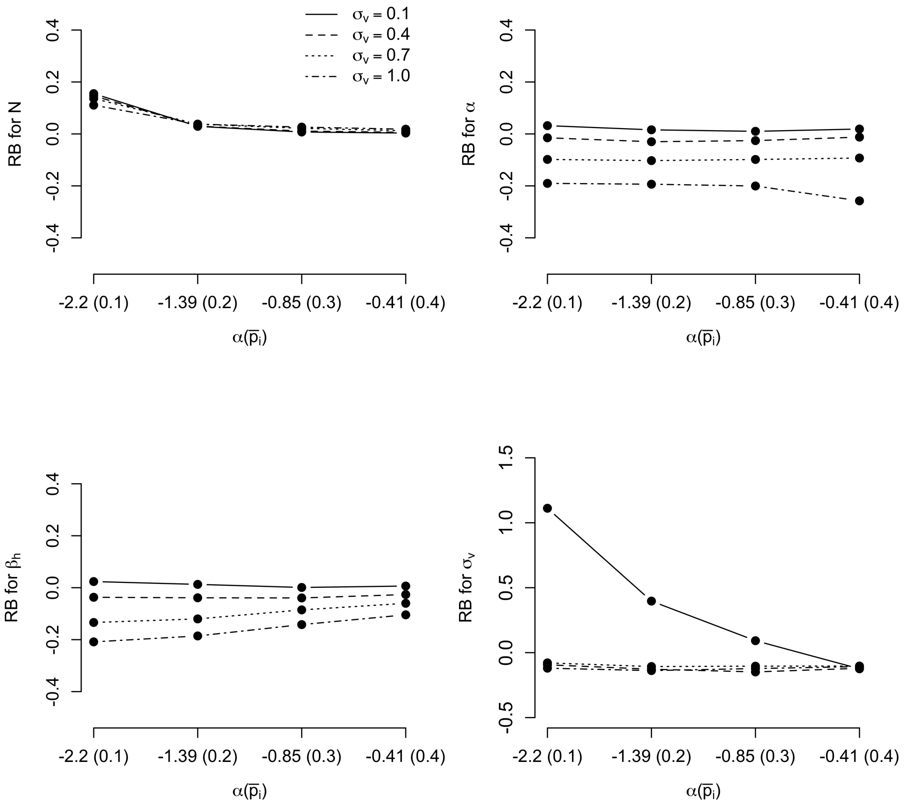

Figure 2.3: Relative Bias (RB) for $\hat{N}$ (upper left), $\hat{\alpha}$ (upper right), $\hat{\beta}_h$ (lower left) and $\hat{\sigma}_v$ (lower right), when $T = 8$, and $N = 250$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values.

1. h-likelihood: I implemented Algorithm 1 in software R (R Core Team, 2020) and obtained my bias-corrected HT estimator with log-normal and Wald CIs proposed in section 2.2.5. The initial values used in the algorithm are: a value of α, generated by using R package VGAM, which fits $\mathcal{M}_h$ without v_i, $v_i = 0$ for all i, and $\sigma_v = 0.01$. I found that changes in any component of the initial values did not affect the results of parameter estimates. R code for my method is included in Appendix A.

2. frequentist (numerical integration): the method I applied was proposed by White and Cooch (2017). It uses Gauss-Hermite quadrature (GHQ) for maximizing the marginal likelihood, constructed by data only for individuals captured at least once, and computes the HT estimator with $\hat{p}_i$, obtained by MLEs, while integrating out v_i from the estimator. The log-normal and Wald CIs are also used in this method. To implement the method, I used program MARK (White and Burnham, 1999), which fitted $\mathcal{M}_h$ automatically when

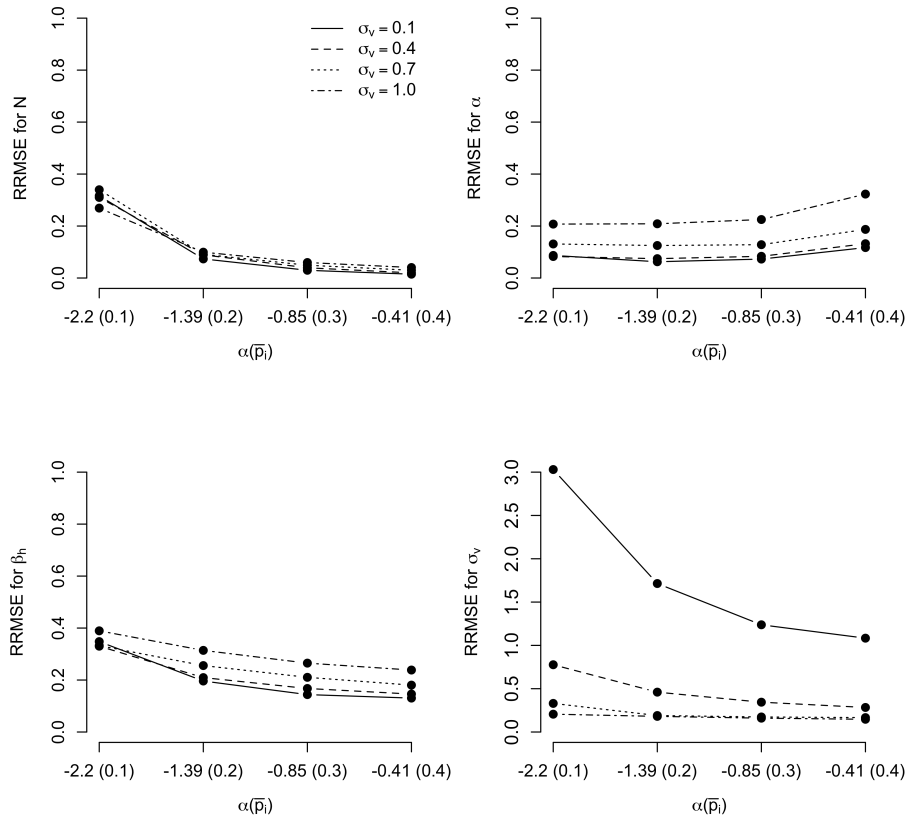

Figure 2.4: Relative root mean square error (RRMSE) for $\hat{N}$ (upper left), $\hat{\alpha}$ (upper right), $\hat{\beta}_h$ (lower left) and $\hat{\sigma}_v$ (lower right), when $T = 8$, and $N = 250$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values.

I input the data into the program. The number of integration nodes for GHQ was set as 101 (default), and the initial values were -0.1 (default) for all parameters, in which σ_v was on log scale.

3. Bayesian: I applied the method illustrated by Royle et al. (2007), who proposed the technique called data augmentation, for fitting $\mathcal{M}_h$ via MCMC sampling. The method presets a super-population, $M >> N$, of pseudo-individuals with the potential to be in the real population with probability ψ, and uses MCMC to sample posterior densities of parameters, including ψ, from the marginal likelihood constructed from the data of all M pseudo-individuals. Setting $M = 1000$, I implemented MCMC in JAGS (Plummer, 2003) run through the R package rjags, and fitted the following model: for $i = 1, ..., M$

Table 2.2: Percent coverage for N based on both the log-normal CI and Wald CI as a function of σ_v and α which in turn defines the median capture probability, $\bar{p}_i$.

			$\alpha\,(\bar{p}_i)$			
(T, N)	CI type	σ_v	$-2.2\,(0.1)$	$-1.39\,(0.2)$	$-0.85\,(0.3)$	$-0.41\,(0.4)$
(5, 100)	log-normal	0.1	82.0	84.8	86.7	86.3
		0.4	78.7	87.3	84.5	85.8
		0.7	79.3	85.1	85.5	82.6
		1.0	75.0	81.7	89.1	88.2
	Wald	0.1	96.4	97.2	97.6	96.4
		0.4	95.4	94.3	95.8	89.3
		0.7	88.5	88.5	87.3	88.1
		1.0	79.9	86.3	89.3	86.2
(8, 250)	log-normal	0.1	86.1	90.4	91.5	90.3
		0.4	85.2	88.7	88.5	88.8
		0.7	85.3	92.9	91.6	88.8
		1.0	93.1	94.9	92.7	92.5
	Wald	0.1	97.7	96.4	93.9	96.1
		0.4	92.9	92.3	93.5	93.1
		0.7	92.3	96.9	95.7	93.8
		1.0	96.0	96.9	97.8	96.9

and $t = 1, ..., T$,

$$\begin{aligned} y_{it} &\sim \text{Bernoulli}(z_i p_i) \\ \text{logit}(p_i) &= \mu + \alpha^* + v_i \end{aligned}$$

and

$$z_i \sim \text{Bernoulli}(\psi)\,,$$

where z_i indicates whether or not pseudo-individual i exists within the population, and the population size is treated as a derived parameter, $N = \sum_{i=1}^{M} z_i$. The linear predictor in the second line is an alternative form of equation (2.3) such that $\mu + \alpha^* = \alpha$. Prior distributions were chosen to be identical to those given by King et al. (2009, pg. 347-

Table 2.3: Results of estimating the population size of snowshoe hares. Three methods based on the h-likelihood, numerical integration via program MARK and the Bayesian method using MCMC implemented via JAGS are applied to fit $\mathcal{M}_h$ and estimate the population size. For the methods based on the h-likelihood and numerical integration, two CIs are reported: log-normal CI (left) and Wald CI (right). Only one CI is reported for the Bayesian estimate.

Method	$\hat{N}$	95% CIs for N		$\hat{\sigma}_v$	Interval
		log-normal	Wald		estimate of $\hat{\sigma}_v$
h-likelihood	94.0	[80, 124]	[73, 115]	0.78	[0.63, 0.92]
numerical integration	91.7	[76, 137]	[64, 119]	0.92	[0.48, 1.76]
Bayesian (MCMC)	94.4	[77, 126]		0.95	[0.63, 1.42]

350):

$$\begin{aligned} \mu &\sim \mathcal{N}(0, 10) \\ \alpha &\sim \mathcal{N}(0, \sigma_\alpha^2) \\ v_i &\sim \mathcal{N}(0, \sigma_v^2) \\ \sigma_\alpha^2, \sigma_v^2 &\sim \Gamma^{-1}(4, 3) \end{aligned}$$

and

$$\psi \sim \text{Beta}(0.001, 1)\,.$$

The initial values used for MCMC were: $\mu = 1.0$, $\alpha^* = 0.5$, $\sigma_\alpha, \sigma_v = 1$, and $\psi = 0.1$, and a single chain was sampled with $2e^6$ iterations with thinning interval of 20. I obtained the point estimate for N and σ_v by the means of their posterior densities, and the credible intervals by their highest posterior density intervals (HPDIs).

Table 2.3 illustrates the results of estimating the population size by applying the three methods. I found that the point estimates, but also the interval estimates, of the population size were similar for all the methods. It is observed that the h-likelihood method identified less heterogeneity in the capture probability (i.e. smaller $\hat{\sigma}_v$ and narrower interval estimate for σ_v) than the other two methods; however, as the sample size, $n = 68$, is small, the estimation of σ_v might not be reliable regardless of the method used. Nonetheless, it is clearly shown that the h-likelihood method provided the results for the population size, comparable to the other methods classically based on Bayesian and frequentist approaches.

2.5 Discussion

The results in Sections 2.3 and 2.4 show that the h-likelihood approach provides a valid method to fit basic MR models for modelling individual heterogeneity and estimating the size of a closed population, which performs as well or better than the existing methods of frequentist and Bayesian approaches. For my method, I construct the conditional h-likelihood from the joint density of data for individuals captured at least once and random effects for modelling unobserved heterogeneity. I then compute the estimates of the parameters and the random effects jointly by maximizing the conditional h-likelihood while correcting the bias in the resulting estimates by using the APHL. I subsequently estimate the population size by using the HT estimator with a bias correction and by substituting the estimates of all unknown quantities obtained from the h-likelihood. The whole process of estimating parameters is free of integration in the h-likelihood approach, whereas numerical integration through quadrature or sampling has been essential for estimating parameters if frequentist or Bayesian frameworks are applied.

The models I have examined in this chapter only consider individual variables as covariates, but it is simple to include other forms of covariates that do not change over time as well. For example, environmental factors, such as forest class and average temperature during the time a MR experiment is performed, can considerably affect the capture probability of individuals. By extending the basic MR model $\mathcal{M}_h$ to these covariates through the linear predictor defined in equation (2.2), the fitting algorithm remains the same, except for the design matrix $\mathbf{T}$ in the IRLS formula in equation (2.9), which is updated with the addition of these covariates. One may be concerned that the assumption of constant covariates over time is too strict to model the capture probability adequately, and thus I relax this assumption in the next chapter.

One unexpected result of applying my approach was that the estimation of the standard deviation, σ_v, of random effects was unsuccessful when the true value of σ_v was extremely small. Such a result was prominently featured in my simulation study under the scenario that assigns σ_v and α to their lowest values, 0.1 and -2.2. According to the results observed from Figures 2.3 and 2.4, this scenario provided the largest RB over 1.0 among all scenarios as well as the largest RRMSE observed at an extreme value over 3.0. This unsatisfactory performance might occur due to the narrow range of the distribution of the capture probability generated by the value of σ_v, while the median capture probability is very low, which causes a small value of the expected sample size. The RB and RRMSE could be improved by increasing either α or σ_v; hence, when σ_v is unknown in practice, a large sample size is suggested to avoid a large bias in $\hat{\sigma}_v$.

One concern that arises when modelling unobserved heterogeneity from MR data is that the distribution of the capture probability may not be strictly identifiable if multiple distributions are considered, and $\mathcal{M}_h$ is no exception. If the model assigns a single family of distributions to the capture probability (e.g., normal random effects are defined as the logit of the capture probabilities for individuals), then the model will be identifiable; however, Link (2003) and Holzmann et al. (2006) showed that if the model of the capture probability contains distributions from multiple families (e.g., the random effects may either be normal on the logit scale or beta on the identity scale), then the model is no longer identifiable. Distributions from the different families may provide an almost identical fit to the observed capture histories, regardless of the number of individuals captured, but may present very different estimates of the population size. As in most models that account for heterogeneity in the capture probability, I have ignored this issue by assuming that the random effects come from a single distribution (specifically, normal on the logit scale). I plan to explore the issue of the non-identifiability of $\mathcal{M}_h$, which is associated with the h-likelihood estimation, in further research.

Chapter 3

H-likelihood Approach to MR Models with Heterogeneity, Time Variation and Behavioural Response

3.1 Introduction

Allowing the capture probability to be time-varying (i.e., different capture probabilities for different time occasions) has been an important issue in modelling MR data as the assumption of the constant capture probability regardless of time may be too strict to adequately model data. In a traditional way, the time-varying capture probability has been studied by estimating a set of capture probabilities, of which each is assigned to a time when individuals are captured (i.e., sampling occasions) and possibly modelled by a function of covariates, such as individual covariates, to account for the effects of the covariates on the capture probability. As an example, Otis et al. (1978) and Chao et al. (1992) analyzed MR data about cottontail rabbits, corrected through 18 successive days, and they found that the MR models with the time-varying capture probability better explained the observed data than those with the constant capture probability.

Another common source of variation in the capture probability is the behavioural effect that alters the capture probability for each individual in response to being previously caught (Pollock, 1982). Two main behavioural tendencies occur based on the experience of trapping: a trap-shy response occurs if individuals are less likely to be captured after being trapped once, and a trap-happy response occurs if individuals are more likely to be captured after being trapped once. These tendencies cause the capture probability in MR models to be split into two probabilities, where one stands for the probability of capturing an individual initially (i.e., initial capture probability) and the other stands for the probability of re-capturing an individidual given that it has been captured once or more before the current occasion (i.e., recapture probability). An example of such MR models is in the study of Wegge et al. (2004), who were interested in estimating tiger abundance and showed that the behavioural effect was significant as the tigers became afraid of the flash from cameras used to detect them.

The same approaches for fitting the basic MR model from the previous chapter with individual heterogeneity have also been applied to the models, also accounting for the time and behavioural effects on the capture probability. This extension generated a total of eight separate models, as proposed by Otis et al. (1978), which describe all possible combinations of the different sources of variation in the capture probability: individual heterogeneity, time, and behavioural effects. As discussed in Chapter 2, individual heterogeneity may be modelled by individual covariates or random effects, representing observed and unobserved individual variation, respectively, while the time and behavioural effects are usually modelled by fixed effects (intercepts) that differ based on sampling occasions and the trapping experience of each individual at each occasion. In frequentist approaches, the fixed effects can be estimated by MLEs obtained from the classical likelihood function, relying only on fixed effects if no random effects are included (Huggins, 1989; Otis et al., 1978), or from the marginal likelihood integrating out the random effects otherwise (Coull and Agresti, 1999; Gimenez and Choquet, 2010; Otis et al., 1978; White and Cooch, 2017). For Bayesian approaches, the integration in the marginal likelihood is usually approximated by MCMC sampling if unobserved heterogeneity is described by the random effects and the time and behavioural effects are estimated from the properties of their posterior densities (King and Brooks, 2008; Royle et al., 2007). The estimation of the population size of some frequentist methods are separated from that of other parameters, and achieved by such as the HT estimator, for example, depending on the time and behavioural effects estimated.

This chapter extends my approach using the h-likelihood for fitting MR models accounting for individual heterogeneity, as in the previous project, to the models that also describe the time and behavioural effects. There are three different types of the models wherein the capture probability is modelled by a function of parameters for either time or behavioural effects or both along with individual covariates and random effects modelling individual heterogeneity. I construct the h-likelihood based on the conditional likelihood for these models and provide the fitting algorithm with applying the bias correction technique, as shown in the previous chapter. Estimating the population size is based on the HT estimator as before, for which I substitute parameter estimates obtained from the h-likelihood and correct the potential bias. To demonstrate my approach, I provide a simulation study with the most general model among the three models as well as a further analysis of MR data related to snowshoe hares considered in the previous chapter.

Table 3.1: Summary of notation used in Chapter 3.

Notation	Definition
y_{it}	Indicator if individual i is captured on occasion t
A_{it}	Indicator if individual i is captured before occasion t
p_{it}	Capture probability for individual i on occasion t
$p^{\dagger}_{it}$	Capture probability for individual i on occasion t when $A_{it} = 0$
$p^{\ddagger}_{it}$	Capture probability for individual i on occasion t when $A_{it} = 1$
α_t	Intercept parameter associated with occasion t
γ	Behavioural effect
$\mathbf{x}_{it}$	Covariate vector associated with $\boldsymbol{\beta}$ for individual i on occasion t
$\mathbf{X}_i$	Design matrix with the i-th row $\mathbf{x}_{it}$
$\boldsymbol{\kappa}$	Vector of all fixed and random effects without γ
$\mathbf{a}_i$	Vectorized form of elements a_{it} for $t = 1, ..., T$; e.g., $\mathbf{y}_i = (y_{i1}, ..., y_{iT})'$
$\mathbf{a}_{i,t=b:c}$	Vectorized form of elements a_{it} for $t = b, ..., c$
$\mathbf{a}$	Vectorized form of elements $\mathbf{a}_i$ for $i = 1, ..., n$; e.g., $\mathbf{y} = (\mathbf{y}'_1, ..., \mathbf{y}'_n)$
$\mathbf{a}_{t=b:c}$	Vectorized form of elements $\mathbf{a}_{i,t=b:c}$
bdiag$(\mathbf{A}, \mathbf{B}, \mathbf{C})$	Block-diagonal matrix consisting of the diagonal blocks $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$
$\mathbf{A} \otimes \mathbf{B}$	Kronecker product of matrices $\mathbf{A}$ and $\mathbf{B}$

3.2 Method

3.2.1 Description of extended MR models: $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$

Considering the basic MR model described in Chapter 2, I extend this model to the capture probability depending on sampling occasions and the trap-responses of individuals. It was noted that the basic MR model in Chapter 2 was extended from one of Otis et al. (1978) models, namely $\mathcal{M}_h$, which allows only for individual heterogeneity. Here I consider three other models extended from the models of Otis et al. (1978), namely $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$, in which the additional subscripts t and b stand for the effect of time-variation and the trap-response on the capture probability, respectively. The choice of the subscripts in the model notation implies the factors determining the capture probability; specifically, $\mathcal{M}_{th}$ is the model that allows the capture probability to vary by individuals and sampling occasions, but does not include the behavioural effect, $\mathcal{M}_{bh}$ is the model that allows the capture probability to vary by individuals and behavioural effect, but be constant over time, and $\mathcal{M}_{tbh}$ is the model that differs by all three factors, individuals, sampling occasions and behavioural effects. For convenience, I follow the same model notations of Otis et al. (1978) throughout this chapter for denoting the extended MR models I consider.

To describe the extended MR models and my approach for fitting them, I re-use the notation in Table 2.1, provided previously, and also use additional notation, where some are updated from Table 2.1, illustrated in Table 3.1. Given these notations, the structures of the extended MR models are explained as follows. Let y_{it} denote the indicator that individual i is captured on occasion t, and $p_{it} = P(y_{it} = 1)$. The challenge with these more complicated models is that p_{it} may now vary by sampling occasions as well as by individuals. This means that $\sum_{t=1}^{T} y_{it}$ is no longer the realization of the response variable that follows a binomial distribution, as in Chapter 2. Instead, we must consider the response variable for each individual to be a vector, observed as $\mathbf{y}_i = (y_{i1}, \ldots, y_{iT})$, whose density is

$$f(\mathbf{y}_i|v_i; \mathbf{p}_i) = \prod_{t=1}^{T} p_{it}^{y_{it}}(1 - p_{it})^{1-y_{it}}\,, \tag{3.1}$$

where p_{it} are functions of fixed effects, $\boldsymbol{\beta}$, and random effects, v_i. As in $\mathcal{M}_h$ in Chapter 2, $v_i \sim \mathcal{N}(0, \sigma_v^2)$ for all i, and they are independent of each other, to continue the assumption of the logit-normal distribution for the capture probability. The linear models previously constructed on p_i in equations (2.2) and (2.3) are extended to p_{it}, which provides the general form,

$$\text{logit}(p_{it}) = \mathbf{x}_{it}'\boldsymbol{\beta} + v_i\,, \tag{3.2}$$

where $\mathbf{x}_{it}$ is a general vector of covariates that may include the time-varying intercepts, the behavioural effect, and other observed individual covariates. If no individual covariates are observed, then it is obtained that

$$\text{logit}(p_{it}) = \begin{cases} \alpha_t + v_i & \text{for } \mathcal{M}_{th} \\ \alpha + \gamma A_{it} + v_i & \text{for } \mathcal{M}_{bh} \\ \alpha_t + \gamma A_{it} + v_i & \text{for } \mathcal{M}_{tbh} \end{cases}.$$

Except as noted otherwise, I use $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$, and $\mathcal{M}_{tbh}$ to refer the general model for p_{it} in equation (3.2).

3.2.2 Conditional h-likelihood

To estimate parameters in the extended MR models, the conditional h-likelihood for each model is built in the same way as the conditional h-likelihood for $\mathcal{M}_h$ in Chapter 2. Using the density function in equation (3.1), I construct the conditional h-likelihoods for the models by

$$\begin{aligned}\mathcal{H}_c(\boldsymbol{\theta};\mathbf{y},\mathbf{v}) &= \prod_{i=1}^{n} f(\mathbf{y}_i|y_i > 0, v_i; \mathbf{p}_i) \times f(v_i;\sigma_v)\\ &= \prod_{i=1}^{n}\left[\frac{\prod_{t=1}^{T} p_{it}^{y_{it}}(1-p_{it})^{1-y_{it}}}{\pi_i}\right]\times\left[\frac{1}{\sigma_v\sqrt{2\pi}}\exp\left(-\frac{v_i^2}{2\sigma_v^2}\right)\right]\end{aligned} \tag{3.3}$$

in general form, conditioning data on the individuals captured at least once. Equation (3.3) differs between the MR models through π_i, the probability that individual i is captured at least one time, defined as

$$\pi_i = \begin{cases} 1-\prod_{t=1}^{T}(1-p_{it}) & \text{for } \mathcal{M}_{th} \\ 1-\prod_{t=1}^{T}(1-p_{it}^{\dagger}) & \text{for } \mathcal{M}_{bh} \text{ and } \mathcal{M}_{tbh} \end{cases}. \tag{3.4}$$

The sign † on the capture probability indicates the probability that an individual is captured on an occasion given that it has not been captured previously. In contrast, the sign ‡ will indicate the probability of capturing an individual given that the individual has been captured at least once before; see Table 3.1 for the details.

Similarly with the linkage between the conditional h-likelihood of $\mathcal{M}_h$ and the h-likelihood of GLMMs in Chapter 2, the conditional h-likelihood in equation (3.3) provides the viewpoint that the extended MR models are vector GLMMs (VGLMMs), which allow the response variables to be multi-dimensional. The general framework of VGLMMs assumes that the response variables, $\mathbf{y}_i$, follows a distribution that falls into the vector exponential family with the density function

$$f(\mathbf{y}_i|\mathbf{v};\boldsymbol{\beta},\phi) = \exp\left(\frac{\boldsymbol{\theta}_i'\mathbf{t}(\mathbf{y}_i) - b(\boldsymbol{\theta}_i)}{\phi} + c(\mathbf{y}_i,\phi)\right), \tag{3.5}$$

where $\boldsymbol{\theta}_i$ is the vector of the natural parameters, linked to the vector of linear predictors $\boldsymbol{\eta}_i$ such that $\boldsymbol{\eta}_i = \mathbf{g}(E(\mathbf{t}(\mathbf{y}_i)))$ with a link function $\mathbf{g}(\cdot)$, ϕ is the dispersion parameter, and $\mathbf{t}(\cdot)$, $b(\cdot)$ and $c(\cdot,\cdot)$ are some known functions. Following Yee et al. (2015), who converted the conditional likelihood, the first term in equation (3.3), to the likelihood of GLMs, I derived that the first term in equation (3.3) corresponds to the density function in equation (3.5) by supposing that

$\mathbf{y}_i$ for $i = 1, ..., n$ are observed random variables such that $y_{i.} > 0$ with

$$\boldsymbol{\theta}_i = \boldsymbol{\eta}_i = \begin{cases} \text{logit}(\mathbf{p}_i) & \text{for } \mathcal{M}_{th} \\ \text{logit}((\mathbf{p}_i^{\dagger\prime}, \mathbf{p}_{i,t=2:T}^{\ddagger\prime})') & \text{for } \mathcal{M}_{bh} \text{ and } \mathcal{M}_{tbh} \end{cases}$$

$$\mathbf{t}(\mathbf{y}_i) = \begin{cases} \mathbf{y}_i & \text{for } \mathcal{M}_{th} \\ (1 - \mathbf{A}_i', \mathbf{A}_{i,t=2:T}')' \odot (\mathbf{y}_i', \mathbf{y}_{i,t=2:T}')' & \text{for } \mathcal{M}_{bh} \text{ and } \mathcal{M}_{tbh} \end{cases}$$

$$b(\boldsymbol{\theta}_i) = \log(\pi_i) + \sum_{t=1}^{T} \log(1 - p_{it})$$

$$\phi = 1$$

and

$$c(\mathbf{y}_i, \phi) = 0\,.$$

It is noted that many properties of the ordinary exponential family described in Chapter 1 extend to the vector exponential family with the density function in equation (3.5). For example, one key property useful in maximizing the conditional h-likelihood is that $E(\mathbf{t}(\mathbf{y}_i)) = \partial b(\boldsymbol{\theta}_i)/\partial \boldsymbol{\theta}_i$ which extends the usual property that $E(y_i) = \partial b(\theta_i)/\partial \theta_i$, where y_i and θ_i are scalar natural parameter and observed response variable of the ordinary exponential family with the density function in equation (1.3). The conditional h-likelihood can be then maximized to obtain the MHLE for $\boldsymbol{\theta}$ based on these extended properties. The fitting algorithm described in Chapter 2 can be also extended for the maximization of the h-likelihood, along with the bias correction for the MHLE, originally developed for fitting GLMMs with a scalar response variable y_i. I describe the bias correction for the MHLE for $\boldsymbol{\theta}$ in $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$ first as below so that the fitting algorithm including the bias correction, for these models, regarded as VGLMMs, are derived.

3.2.3 Bias correction for MHLEs

For the same reason in Chapter 2 that the MHLEs computed directly from the conditional h-likelihood for $\mathcal{M}_h$ can be severely biased, I perform the bias correction for the MHLE for $\boldsymbol{\theta}$ from the conditional h-likelihoods of the extended MR models. The bias correction is achieved through the multi-dimensional analogue of the four steps of the bias correction, as described in Section 2.2.3, where I define the APHLs

$$h_c^A(\boldsymbol{\beta}; \mathbf{y}, \hat{\mathbf{v}}, \hat{\sigma}_v) = h_c - \frac{1}{2}\log\Big[\det\Big(-\frac{1}{2\pi}\frac{\partial^2 h_c}{\partial \mathbf{v} \partial \mathbf{v}'}\Big)\Big]\Big|_{\mathbf{v}=\hat{\mathbf{v}}, \sigma_v=\hat{\sigma}_v} \tag{3.6}$$

Algorithm 2 Fitting algorithm for $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$

```
Set initial value $\hat{\boldsymbol{\theta}}^{(0)} = (\hat{\boldsymbol{\delta}}'^{(0)}, \hat{\sigma}_v^{(0)})'$ (components of $\boldsymbol{\delta}$ depend on the model)
Let $r = 0$
while convergence criterion $\hat{\boldsymbol{\theta}}^{(r)} \approx \hat{\boldsymbol{\theta}}^{(r+1)}$ not met do
    Let $\hat{\boldsymbol{\delta}}^{(r,0)} = \hat{\boldsymbol{\delta}}^{(r)}$;
    Let $t = 0$;
    while convergence criterion $\hat{\boldsymbol{\delta}}^{(r,t)} \approx \hat{\boldsymbol{\delta}}^{(r,t+1)}$ not met do
        (Step 1) Given $\hat{\sigma}_v^{(r)}$, solve equation (3.8) for $\hat{\boldsymbol{\delta}}^{(r,t)}$;
        $t \leftarrow t + 1$
    end while
    Let $\hat{\boldsymbol{\delta}}^{(r+1)} = \hat{\boldsymbol{\delta}}^{(r,t)}$
    (Step 2) Given $\hat{\boldsymbol{\delta}}^{(r+1)}$, obtain $\hat{\sigma}_v^{(r+1)}$ by fitting the gamma GLM with
        • observed response variables: $\hat{\mathbf{v}}^{(r+1)} = (\hat{v}_1^{(r+1)}, ..., \hat{v}_n^{(r+1)})'$
        • prior weight: $q_i$ in equation (3.10) for $\hat{v}_i^{(r+1)}$, given $\hat{\boldsymbol{\delta}}^{(r+1)}$ and $\hat{\sigma}_v^{(r)}$
        • linear predictor: $\tau = \log(\sigma_v^2)$
end while
```

and

$$h_c^A(\sigma_v; \mathbf{y}, \hat{\mathbf{v}}, \hat{\boldsymbol{\beta}}) = h_c - \frac{1}{2}\log\left[\det\left(- \frac{1}{2\pi}\frac{\partial^2 h_c}{\partial \mathbf{v} \partial \mathbf{v}'}\right)\right]\Bigg|_{\boldsymbol{\delta}=\hat{\boldsymbol{\delta}}} \tag{3.7}$$

in Steps 1 and 3, respectively. Although these APHLs are functionally equivalent to the AHPLs in equations (2.7) and (2.8), they have different arguments (e.g., the components of $\boldsymbol{\beta}$) depending on the models. Performing the four steps until convergence, I obtain $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}_v$ from the APHLs of Steps 1 and 3 while obtaining $\hat{\mathbf{v}}$ from the conditional h-likelihood of Step 2, given $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}_v$ updated from the other steps.

3.2.4 Fitting Algorithm

I derive the general algorithm for fitting the extended MR models in the same way as the algorithm for fitting $\mathcal{M}_h$ in Chapter 2, illustrated in Algorithm 1. Algorithm 2 describes the details of the algorithm for fitting the extended MR models. Step 1 of the algorithm again solves two normal equations,

$$\frac{\partial h_c^A}{\partial \boldsymbol{\beta}} = \mathbf{X}'(\mathbf{y} - \boldsymbol{\mu} - \mathbf{s}) = \mathbf{0}_{\dim(\boldsymbol{\beta})}$$

with respect to $\boldsymbol{\beta}$, and

$$\frac{\partial h_c}{\partial \mathbf{v}} = (\mathbf{y} - \boldsymbol{\mu}) - \frac{1}{\sigma_v^2}\mathbf{v} = \mathbf{0}_n$$

with respect to $\mathbf{v}$, and I have derived that the equation for the IRLS is

$$\mathbf{T}'\mathbf{W}^{(t-1)}\mathbf{T}\hat{\boldsymbol{\delta}}^{(t)} = \mathbf{T}'\mathbf{W}^{(t-1)}\mathbf{z}^{*(t-1)}, \tag{3.8}$$

which remains the same form of equation (2.9) but contains updated components in $\mathbf{T}$, $\mathbf{W}^{(t)}$ and $\mathbf{z}^{*(t)}$. Specifically,

$$\mathbf{T} = \begin{pmatrix} \mathbf{X} & \mathbf{Z} \\ \mathbf{0}_{n\times\dim(\boldsymbol{\beta})} & \mathbf{I}_n \end{pmatrix}$$

is the design matrix such that $(\boldsymbol{\eta}', \mathbf{v}')' = \mathbf{T}\boldsymbol{\delta}$, and $\mathbf{Z} = \mathbf{1}_{n\times n} \otimes \mathbf{1}_T$ for $\mathcal{M}_{th}$ and $\mathbf{1}_{n\times n} \otimes \mathbf{1}_{2T-1}$ for $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$,

$$\mathbf{W}^{(t)} = \left.\begin{pmatrix} \text{bdiag}\left(\frac{\partial\boldsymbol{\mu}_1}{\partial\boldsymbol{\eta}_1}, ..., \frac{\partial\boldsymbol{\mu}_n}{\partial\boldsymbol{\eta}_n}\right) & \mathbf{0}_{\dim(\boldsymbol{\eta})\times n} \\ \mathbf{0}_{n\times\dim(\boldsymbol{\eta}))} & \frac{1}{\sigma_v^2}\mathbf{I}_n \end{pmatrix}\right|_{\boldsymbol{\delta}=\hat{\boldsymbol{\delta}}^{(t)}, \sigma_v=\hat{\sigma}_v}$$

is the t-th weight matrix, and

$$\mathbf{z}^{*(t)} = \left.\begin{pmatrix} \boldsymbol{\eta} + \text{bdiag}\left(\frac{\partial\boldsymbol{\mu}_1}{\partial\boldsymbol{\eta}_1}, ..., \frac{\partial\boldsymbol{\mu}_n}{\partial\boldsymbol{\eta}_n}\right)(\mathbf{y} - \boldsymbol{\mu} - \mathbf{s}) \\ \mathbf{Z}'\mathbf{s} \end{pmatrix}\right|_{\boldsymbol{\delta}=\hat{\boldsymbol{\delta}}^{(t)}, \sigma_v=\hat{\sigma}_v}$$

is the t-th adjusted response variable of the IRLS equation. In the updated components, I write

$$\boldsymbol{\mu} = E(\mathbf{y}) = (\boldsymbol{\mu}_1', ..., \boldsymbol{\mu}_n')'$$

and

$$\mathbf{s} = (\mathbf{s}_1', ..., \mathbf{s}_n')'$$

in general for all models, and have derived that

1) for $\mathcal{M}_{th}$,

$$\begin{aligned} \boldsymbol{\mu}_i &= (\mu_{i1}, ..., \mu_{iT})' \\ \mu_{it} &= \frac{1-\pi_i}{\pi_i} p_{it} \\ \mathbf{s}_i &= (s_{i1}, ..., s_{iT})' \end{aligned}$$

and

$$s_{it} = \frac{1}{2}\text{diagonal}\left(\left(-\frac{\partial^2 h_c}{\partial v_i^2}\right)^{-1} \mathbf{J}_T\left(\frac{\partial^2\boldsymbol{\mu}_i}{\partial\boldsymbol{\eta}_i\partial\eta_{it}} + \left.\frac{\partial^2\boldsymbol{\mu}_i}{\partial\boldsymbol{\eta}_i\partial v_i}\right|_{v_i=\hat{v}_i} \frac{\partial\hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial\eta_{it}}\right)\right),$$

where $\mathbf{J}_T$ is the square matrix of 1s with dimension T,

$$-\frac{\partial^2 h_c}{\partial v_i^2} = \mathbf{1}_T' \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\eta}_i} \mathbf{1}_T + \frac{1}{\sigma_v^2},$$

in which $\partial \boldsymbol{\mu}_i / \partial \boldsymbol{\eta}_i$ has the (a, b)-th element

$$\frac{\partial \mu_{ia}}{\partial \eta_{ib}} = \begin{cases} \dfrac{p_{ia}}{\pi_i}\left(1 - \dfrac{p_{ia}}{\pi_i}\right) & \text{if } a = b \\ \dfrac{p_{ia}}{\pi_i}\left(p_{ib} - \dfrac{p_{ib}}{\pi_i}\right) & \text{if } a \neq b \end{cases},$$

$\partial^2 \boldsymbol{\mu}_i / \partial \boldsymbol{\eta}_i \partial \eta_{it}$ has the (a, b)-th element

$$\frac{\partial^2 \mu_{ia}}{\partial \eta_{ib} \partial \eta_{it}} = \begin{cases} \left(\dfrac{1}{\pi_i} - \dfrac{2p_{it}}{\pi_i^2}\right) p_{it}(1 - p_{it}) - \left(\dfrac{p_{it}}{\pi_i^2} - \dfrac{2p_{it}^2}{\pi_i^3}\right) p_{it}(1 - \pi_i) & \text{if } a = b = t \\ -\left(\dfrac{p_{ia}}{\pi_i^2} - \dfrac{2p_{ia}^2}{\pi_i^3}\right) p_{it}(1 - \pi_i) & \text{if } a = b \neq t \\ p_{ia} p_{ib}(1 - p_{ia})\left(\dfrac{1}{\pi_i} - \dfrac{1}{\pi_i^2}\right) + p_{it} p_{ib}(1 - \pi_i)\left(\dfrac{p_{it}}{\pi^2} - \dfrac{2p_{it}}{\pi^3}\right) & \text{if } a = t \neq b \\ p_{ia} p_{ib}(1 - p_{ib})\left(\dfrac{1}{\pi_i} - \dfrac{1}{\pi_i^2}\right) + p_{it} p_{ia}(1 - \pi_i)\left(\dfrac{p_{it}}{\pi^2} - \dfrac{2p_{it}}{\pi^3}\right) & \text{if } a \neq b = t \\ p_{ia} p_{ib}(1 - \pi_i)\left(\dfrac{p_{it}}{\pi^2} - \dfrac{2p_{it}}{\pi^3}\right) & \text{if } a \neq b \neq t, a \neq t \end{cases},$$

$\partial^2 \boldsymbol{\mu}_i / \partial \boldsymbol{\eta}_i \partial v_i$ has the (a, b)-th element

$$\frac{\partial^2 \mu_{ia}}{\partial \eta_{ib} \partial v_i} = \begin{cases} \left(\dfrac{1}{\pi_i} - \dfrac{2p_{ia}}{\pi_i^2}\right) p_{ia}(1 - p_{ia}) - \left(\dfrac{p_{ia}}{\pi_i^2} - \dfrac{p_{ia}^2}{\pi_i^3}\right)(1 - \pi_i) \displaystyle\sum_{r=1}^{T} p_{ir} & \text{if } a = b \\ p_{ia} p_{ib}(2 - p_{ia} - p_{ib})\left(\dfrac{1}{\pi_i} - \dfrac{1}{\pi_i^2}\right) - p_{ia} p_{ib}\left(\dfrac{1}{\pi_i^2} - \dfrac{1}{\pi_i^3}\right)(1 - \pi_i) \displaystyle\sum_{r=1}^{T} p_{ir} & \text{if } a \neq b \end{cases},$$

and

$$\frac{\partial \hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial \eta_{it}} = \left(-\frac{\partial^2 h_c}{\partial \eta_{it}^2}\right)^{-1} \frac{\partial^2 h_c}{\partial v_i \partial \eta_{it}},$$

in which

$$-\frac{\partial^2 h_c}{\partial \eta_{it}^2} = \sum_{r=1}^{T} \frac{\partial^2 \mu_{ir}}{\partial \eta_{it}^2} + \frac{1}{T \sigma_v^2}$$

and

$$\frac{\partial^2 h_c}{\partial v_i \partial \eta_{it}} = -\sum_{r=1}^{T} \frac{\partial^2 \mu_{ir}}{\partial \eta_{it}^2} \, ;$$

2) for $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$,

$$\begin{aligned}
\boldsymbol{\mu}_i &= (\mu_{i1}^{\dagger}, ..., \mu_{iT}^{\dagger}, \mu_{i2}^{\ddagger}, ..., \mu_{iT}^{\ddagger})' \\
\mu_{it}^{\dagger} &= \frac{1-\pi_i}{\pi_i} p_{it}^{\dagger} + (1 - A_{it}) p_{it}^{\dagger} \\
\mu_{it}^{\ddagger} &= A_{it} p_{it}^{\ddagger} \\
\mathbf{s}_i &= (s_{i1}^{\dagger}, ..., s_{iT}^{\dagger}, s_{i2}^{\ddagger}, ..., s_{iT}^{\ddagger})' \\
s_{it}^{\dagger(\ddagger)} &= \frac{1}{2}\text{diagonal}\left(\left(-\frac{\partial^2 h_c}{\partial v_i^2}\right)^{-1} \mathbf{J}_{2T-1}\left(\frac{\partial^2 \boldsymbol{\mu}_i^{\dagger(\ddagger)}}{\partial \boldsymbol{\eta}_i^{\dagger(\ddagger)} \partial \eta_{it}^{\dagger(\ddagger)}} + \left.\frac{\partial^2 \boldsymbol{\mu}_i^{\dagger(\ddagger)}}{\partial \boldsymbol{\eta}_i^{\dagger(\ddagger)} \partial v_i}\right|_{v_i=\hat{v}_i} \frac{\partial \hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial \eta_{it}^{\dagger(\ddagger)}}\right)\right) \\
\boldsymbol{\eta}_i &= (\eta_{i1}^{\dagger}, ..., \eta_{iT}^{\dagger}, \eta_{i2}^{\ddagger}, ..., \eta_{iT}^{\ddagger})'
\end{aligned}$$

and

$$\eta_{it}^{\dagger(\ddagger)} = \log(p_{it}^{\dagger(\ddagger)}) \, ,$$

where $\mathbf{J}_{2T-1}$ is the square matrix of 1s with dimension $2T-1$,

$$-\frac{\partial^2 h_c}{\partial v_i^2} = \mathbf{1}_{2T-1}' \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\eta}_i} \mathbf{1}_{2T-1} + \frac{1}{\sigma_v^2}$$

with

$$\frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\eta}_i} = \text{bdiag}\left(\frac{\partial \boldsymbol{\mu}_i^{\dagger}}{\partial \boldsymbol{\eta}_i^{\dagger}}, \frac{\partial \boldsymbol{\mu}_i^{\ddagger}}{\partial \boldsymbol{\eta}_i^{\ddagger}}\right) ,$$

in which $\partial \boldsymbol{\mu}_i^{\dagger(\ddagger)} / \partial \boldsymbol{\eta}_i^{\dagger(\ddagger)}$ has the (a, b)-th element

$$\frac{\partial \mu_{ia}^{\dagger}}{\partial \eta_{ib}^{\dagger}} = \begin{cases} \dfrac{p_{ia}^{\dagger}}{\pi_i}\left(1 - \dfrac{p_{ia}^{\dagger}}{\pi_i}\right) + (1 - A_{ia}) p_{ia}^{\dagger}(1 - p_{ia}^{\dagger}) & \text{if } a = b \\ \dfrac{p_{ia}^{\dagger}}{\pi_i}\left(p_{ib}^{\dagger} - \dfrac{p_{ib}^{\dagger}}{\pi_i}\right) & \text{if } a \neq b \end{cases} ,$$

and

$$\frac{\partial \mu^{\ddagger}_{ia}}{\partial \eta^{\ddagger}_{ib}} = \begin{cases} A_{it} p^{\ddagger}_{ia}(1 - p^{\ddagger}_{ia}) & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases},$$

$\partial^2 \boldsymbol{\mu}^{\dagger(\ddagger)}_i / \partial \boldsymbol{\eta}^{\dagger(\ddagger)}_i \partial \eta^{\dagger(\ddagger)}_{it}$ has the (a, b)-th element

$$\frac{\partial^2 \mu^{\dagger}_{ia}}{\partial \eta^{\dagger}_{ib} \partial \eta^{\dagger}_{it}} = \begin{cases} \left(\frac{1}{\pi_i} - \frac{2p^{\dagger}_{it}}{\pi_i^2}\right) p^{\dagger}_{it}(1 - p^{\dagger}_{it}) - \left(\frac{p^{\dagger}_{it}}{\pi_i^2} - \frac{2p^{\dagger 2}_{it}}{\pi_i^3}\right) p^{\dagger}_{it}(1 - \pi_i) & \text{if } a = b = t \\ \qquad +(1 - A_{it})(1 - 2p^{\dagger}_{it}) p^{\dagger}_{it}(1 - p^{\dagger}_{it}) & \\ -\left(\frac{p^{\dagger}_{ia}}{\pi_i^2} - \frac{2p^{2^{\dagger}}_{ia}}{\pi_i^3}\right) p^{\dagger}_{it}(1 - \pi_i) & \text{if } a = b \neq t \\ p^{\dagger}_{ia} p^{\dagger}_{ib}(1 - p^{\dagger}_{ia}) \left(\frac{1}{\pi_i} - \frac{1}{\pi_i^2}\right) + p^{\dagger}_{it} p^{\dagger}_{ib}(1 - \pi_i) \left(\frac{p^{\dagger}_{it}}{\pi^2} - \frac{2p^{\dagger}_{it}}{\pi^3}\right) & \text{if } a = t \neq b \\ p^{\dagger}_{ia} p^{\dagger}_{ib}(1 - p^{\dagger}_{ib}) \left(\frac{1}{\pi_i} - \frac{1}{\pi_i^2}\right) + p^{\dagger}_{it} p^{\dagger}_{ia}(1 - \pi_i) \left(\frac{p^{\dagger}_{it}}{\pi^2} - \frac{2p^{\dagger}_{it}}{\pi^3}\right) & \text{if } a \neq b = t \\ p^{\dagger}_{ia} p^{\dagger}_{ib}(1 - \pi_i) \left(\frac{p^{\dagger}_{it}}{\pi^2} - \frac{2p^{\dagger}_{it}}{\pi^3}\right) & \text{if } a \neq b \neq t, a \neq t \end{cases}$$

and

$$\frac{\partial^2 \mu^{\ddagger}_{ia}}{\partial \eta^{\ddagger}_{ib} \partial \eta^{\ddagger}_{it}} = \begin{cases} A_{it}(1 - 2p^{\ddagger}_{it}) p^{\ddagger}_{it}(1 - p^{\ddagger}_{it}) & \text{if } a = b = t \\ 0 & \text{otherwise} \end{cases},$$

$\partial^2 \boldsymbol{\mu}^{\dagger(\ddagger)}_i / \partial \boldsymbol{\eta}^{\dagger(\ddagger)}_i \partial v_i$ has the (a, b)-th element

$$\frac{\partial^2 \mu^{\dagger}_{ia}}{\partial \eta^{\dagger}_{ib} \partial v_i} = \begin{cases} \left(\frac{1}{\pi_i} - \frac{2p^{\dagger}_{ia}}{\pi_i^2}\right) p^{\dagger}_{ia}(1 - p^{\dagger}_{ia}) - \left(\frac{p^{\dagger}_{ia}}{\pi_i^2} - \frac{p^{\dagger 2}_{ia}}{\pi_i^3}\right)(1 - \pi_i) \sum_{r=1}^{T} p^{\dagger}_{ir} & \text{if } a = b \\ \qquad +(1 - A_{it})(1 - 2p^{\dagger}_{it}) p^{\dagger}_{it}(1 - p^{\dagger}_{it}) & \\ p^{\dagger}_{ia} p^{\dagger}_{ib}(2 - p^{\dagger}_{ia} - p^{\dagger}_{ib}) \left(\frac{1}{\pi_i} - \frac{1}{\pi_i^2}\right) - p^{\dagger}_{ia} p^{\dagger}_{ib} \left(\frac{1}{\pi_i^2} - \frac{1}{\pi_i^3}\right)(1 - \pi_i) \sum_{r=1}^{T} p^{\dagger}_{ir} & \text{if } a \neq b \end{cases}$$

and

$$\frac{\partial^2 \mu^{\ddagger}_{ia}}{\partial \eta^{\ddagger}_{ib} \partial v_i} = \frac{\partial^2 \mu^{\ddagger}_{ia}}{\partial \eta^{\ddagger}_{ib} \partial \eta^{\ddagger}_{it}},$$

and

$$\frac{\partial \hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial \eta_{it}^{\dagger(\ddagger)}} = \left(-\frac{\partial^2 h_c}{\partial \eta_{it}^{2\dagger(\ddagger)}}\right)^{-1} \frac{\partial^2 h_c}{\partial v_i \partial \eta_{it}^{\dagger(\ddagger)}},$$

in which

$$-\frac{\partial^2 h_c}{\partial \eta_{it}^{2\dagger(\ddagger)}} = \sum_{r=1}^{T} \frac{\partial^2 \mu_{ir}^{\dagger(\ddagger)}}{\partial \eta_{it}^{2\dagger(\ddagger)}} + \frac{1}{(2T-1)\sigma_v^2}$$

and

$$\frac{\partial^2 h_c}{\partial v_i \partial \eta_{it}^{\dagger(\ddagger)}} = -\sum_{r=1}^{T} \frac{\partial^2 \mu_{ir}^{\dagger(\ddagger)}}{\partial \eta_{it}^{2\dagger(\ddagger)}}.$$

The IRLS algorithm repeatedly solves equation (3.8) about $\hat{\boldsymbol{\delta}}$ until convergence is achieved for the solution of equation (3.8). n Step 2 of the algorithm, the normal equation

$$\frac{\partial h_c^A}{\partial \tau} = \sum_{i=1}^{n} \frac{\partial \sigma_v^2}{\partial \tau} \frac{v_i - (1-q_i)\sigma_v^2}{2\sigma_v^4} = 0 \tag{3.9}$$

is again solved as a function of $\tau = \log(\sigma_v^2)$, where

$$q_i = \begin{cases} \left(-\frac{\partial^2 h_c}{\partial v_i^2}\right)^{-1}\left(\sum_{t=1}^{T} \frac{\partial^2 \mu_i}{\partial \eta_{it} \partial v_i}\Big|_{v_i=\hat{v}_i} \frac{\partial \hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial \tau} + \frac{\partial^2 h_c}{\partial v_i \partial \tau}\right) & \text{for } \mathcal{M}_h \\ \left(-\frac{\partial^2 h_c}{\partial v_i^2}\right)^{-1}\left[\left(\sum_{t=1}^{T} \frac{\partial^2 \mu_i}{\partial \eta_{it}^{\dagger} \partial v_i} + \sum_{t=2}^{T} \frac{\partial^2 \mu_i}{\partial \eta_{it}^{\ddagger} \partial v_i}\right)\Big|_{v_i=\hat{v}_i} \frac{\partial \hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial \tau} + \frac{\partial^2 h_c}{\partial v_i \partial \tau}\right] & \text{for } \mathcal{M}_{bh} \text{ and } \mathcal{M}_{tbh} \end{cases} \tag{3.10}$$

in which

$$\frac{\partial \hat{v}_i(\boldsymbol{\beta}, \sigma_v)}{\partial \tau} = \left(-\frac{\partial^2 h_c}{\partial v_i^2}\right)^{-1} \frac{\partial^2 h_c}{\partial v_i \partial \tau}.$$

and

$$\frac{\partial^2 h_c}{\partial v_i \partial \tau} = \frac{1}{\sigma_v^2}.$$

Equation (3.9) corresponds to the normal equation of the gamma GLM such that the response variables are observed as v_i, each of which has the prior weight, q_i, and linear predictor, $\tau = \log(\sigma_v^2)$. Hence, similarly with Step 2 of Algorithm 1, I fit the gamma GLM through the IRLS algorithm, which can be implemented through existing routines in most modern software.

The covariance matrix of $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ is obtained in the same way as it obtained in Chapter 2. The covariance matrix of $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ is given by

$$\mathrm{Var}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) = \begin{pmatrix} \mathrm{Var}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) & \mathbf{0}_{\mathrm{dim}(\boldsymbol{\delta})} \\ \mathbf{0}'_{\mathrm{dim}(\boldsymbol{\delta})} & \mathrm{Var}(\hat{\sigma}_v) \end{pmatrix}, \tag{3.11}$$

according to the h-likelihood properties given in Chapter 1, in which I have derived that

$$\mathrm{Var}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) \approx (\mathbf{T}'\mathbf{W}^*\mathbf{T})^{-1}|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\,. \tag{3.12}$$

I approximate $\mathrm{Var}(\hat{\sigma}_v)$ by the delta method, such that $\mathrm{Var}(\hat{\sigma}_v) \approx \sigma_v^2/4 \times \mathrm{Var}(\hat{\tau})$, where $\mathrm{Var}(\hat{\tau})$ is obtained directly from the IRLS algorithm for fitting the gamma GLM. The variance–covariance matrix of $\hat{\mathbf{v}}(\boldsymbol{\beta}, \sigma_v)$, required in the population size estimation (Section 3.2.5), is approximated by

$$\begin{aligned} \mathrm{Var}(\hat{\mathbf{v}}(\boldsymbol{\beta}, \sigma_v) - \mathbf{v}) &\approx \left(- \frac{\partial^2 h_c}{\partial \mathbf{v} \partial \mathbf{v}'} \right)^{-1} \\ &\approx \begin{cases} \left[\mathrm{bdiag}\left(\mathbf{1}'_T \dfrac{\partial \mu_1}{\partial \eta_1} \mathbf{1}_T, ..., \mathbf{1}'_T \dfrac{\partial \mu_n}{\partial \eta_n} \mathbf{1}_T \right) + \dfrac{1}{\sigma_v^2} \mathbf{I}_{n \times n} \right]^{-1} \Bigg|_{\mathbf{v}=\hat{\mathbf{v}}} & \text{for } \mathcal{M}_{th} \\ \left[\mathrm{bdiag}\left(\mathbf{1}'_{2T-1} \dfrac{\partial \mu_1}{\partial \eta_1} \mathbf{1}_{2T-1}, ..., \mathbf{1}'_{2T-1} \dfrac{\partial \mu_n}{\partial \eta_n} \mathbf{1}_{2T-1} \right) \right. & \text{for } \mathcal{M}_{bh} \text{ and } \mathcal{M}_{tbh} \\ \qquad \left. + \dfrac{1}{\sigma_v^2} \mathbf{I}_{n \times n} \right]^{-1} \Bigg|_{\mathbf{v}=\hat{\mathbf{v}}} & \end{cases} . \end{aligned} \tag{3.13}$$

These variance–covariance matrices can be used to draw Wald-typed inference through the asymptotic properties of the h-likelihood,

$$\hat{\boldsymbol{\delta}} - \boldsymbol{\delta} \dot{\sim} \mathcal{N}(\mathbf{0}_{\mathrm{dim}(\boldsymbol{\theta})}, \mathrm{Var}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})) \tag{3.14}$$

and

$$\mathbf{v}|\mathbf{y} \dot{\sim} \mathcal{N}(\hat{\mathbf{v}}(\boldsymbol{\beta}, \sigma_v), \mathrm{Var}(\hat{\mathbf{v}}(\boldsymbol{\beta}, \sigma_v) - \mathbf{v}))\,, \tag{3.15}$$

as shown by Lee and Nelder (1996), if the sample size, n, is large enough. I shall use these asymptotic properties to make inference for the population size N.

3.2.5 Estimation of population size

The estimation of the population size for the models $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$ is based on the HT estimator, as for the basic model, $\mathcal{M}_h$, in the previous chapter. Denoting $\boldsymbol{\delta}$, but excluding the behaviour effect, γ, as $\boldsymbol{\kappa}$ (remain $\boldsymbol{\delta}$ for $\mathcal{M}_{th}$), I first rewrite the HT estimator in equation 1.2 as the general form,

$$\hat{N}(\boldsymbol{\kappa}) = \sum_{i=1}^{N} \frac{1}{\pi_i} = \sum_{i=1}^{n} \frac{c_i}{\pi_i},$$

where $c_i = I(y_{i.} > 0)$, and π_i is defined in equation (3.4) for each model. I use the notation, $\boldsymbol{\kappa}$, for indicating that the HT estimator does not depend on the behavioural effect, γ, since it is a function of the probability that the individuals captured at least once, π_i, which is free of γ. In reality, as $\boldsymbol{\kappa}$ is unknown, my estimator for N builds on the idea of Huggins (1989), by which any parameter estimates are substituted into the HT estimator, and so is initially considered as

$$\hat{N}(\hat{\boldsymbol{\kappa}}) = \sum_{i=1}^{n} \frac{c_i}{\hat{\pi}_i},$$

where $\hat{\boldsymbol{\kappa}}$ is obtained from by maximizing the likelihood as in Algorithm 2.

In my simulation study with $\mathcal{M}_{tbh}$ and $\hat{N}(\hat{\boldsymbol{\kappa}})$ as the estimator of N, I observed that $\hat{N}(\hat{\boldsymbol{\kappa}})$ underestimates the true value of N, especially when the p_{it} are small on average. To show the bias in $\hat{N}(\hat{\boldsymbol{\kappa}})$ and so derive an unbiased estimator by correcting the bias, I use the first-order Taylor expansion to approximate the following three functions:

$$\hat{N}(\hat{\boldsymbol{\kappa}}) \approx \hat{N}(\boldsymbol{\kappa}) + \frac{\partial \hat{N}(\boldsymbol{\kappa})}{\partial \boldsymbol{\kappa}'}(\hat{\boldsymbol{\kappa}} - \boldsymbol{\kappa}) \tag{3.16}$$

$$\hat{N}(\boldsymbol{\kappa}) \approx \hat{N}(\tilde{\boldsymbol{\kappa}}) + \frac{\partial \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}'}(\boldsymbol{\kappa} - \tilde{\boldsymbol{\kappa}}) \tag{3.17}$$

and

$$\frac{\partial \hat{N}(\boldsymbol{\kappa})}{\partial \boldsymbol{\kappa}'} \approx \frac{\partial \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}'} + (\boldsymbol{\kappa} - \tilde{\boldsymbol{\kappa}})' \frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}}, \tag{3.18}$$

where $\hat{N}(\hat{\boldsymbol{\kappa}})$ is expanded about $\boldsymbol{\kappa}$, and the other functions are expanded about $\tilde{\boldsymbol{\kappa}}$, denoting $\boldsymbol{\kappa}$ with the estimate $\hat{v}_i(\boldsymbol{\beta}, \sigma_v)$ in place of the true value v_i. Following the same steps of derivations through equations (2.16-2.21) and replacing the Taylor expansions in equations (2.13), (2.14) and (2.15) with the above expressions, I derived the approximation for the mean of the sampling distribution for $\hat{N}(\hat{\boldsymbol{\kappa}})$,

$$E(\hat{N}(\hat{\boldsymbol{\kappa}})) \approx N - tr\Big(\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}} \mathrm{Var}(\boldsymbol{\kappa})\Big),$$

and so $\hat{N}(\hat{\boldsymbol{\kappa}})$ is shown to have negative bias $tr\Big(\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}} \mathrm{Var}(\boldsymbol{\kappa})\Big)$. Hence, I propose the new, approximately unbiased estimator,

$$\hat{N}^{\star}(\hat{\boldsymbol{\kappa}}) = \sum_{i=1}^{n} \frac{1}{\hat{\pi}_i} + tr\Big(\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}} \mathrm{Var}(\boldsymbol{\kappa})\Big)\Big|_{\boldsymbol{\kappa} = \hat{\boldsymbol{\kappa}}},$$

where

$$\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}} = \begin{cases} -\sum_{i=1}^{n} \sum_{t=1}^{T} p_{it}(1-\pi_i)\Big[\frac{(1-p_{it})}{\pi_i^2} + p_{i.}\Big(\frac{1}{\pi_i^2} - \frac{2}{\pi^3}\Big)\Big]\mathbf{w}_i\mathbf{w}_i' & \text{for } \mathcal{M}_{th} \\ -\sum_{i=1}^{n} \sum_{t=1}^{T} p_{it}(1-\pi_i)\Big[\frac{(1-p_{it})}{\pi_i^2} + p_{i.}\Big(\frac{1}{\pi_i^2} - \frac{2}{\pi^3}\Big)\Big]\mathbf{w}_i\mathbf{w}_i' & \text{for } \mathcal{M}_{bh} \text{ and } \mathcal{M}_{tbh} \end{cases}$$

with $p_{i.}^{(\dagger)} = \sum_{t=1}^{T} p_{it}^{(\dagger)}$ and $\mathbf{w}_i$ being the covariate vector associated with $\boldsymbol{\kappa}$, and

$$\text{Var}(\boldsymbol{\kappa}) = \begin{pmatrix} \mathbf{0}_{[\dim(\boldsymbol{\beta})-1]\times[\dim(\boldsymbol{\beta})-1]} & \mathbf{0}_{[\dim(\boldsymbol{\beta})-1]\times n} \\ \mathbf{0}_{n\times[\dim(\boldsymbol{\beta})-1]} & \text{Var}(\hat{\mathbf{v}}(\boldsymbol{\beta}, \sigma_v)) \end{pmatrix}.$$

I found in my simulation study that this estimator has much reduced bias when compared to the naive estimator, $\hat{N}(\hat{\boldsymbol{\kappa}})$.

The computation of $\text{Var}(\hat{N}^\star(\hat{\boldsymbol{\kappa}}))$ can derived from the same logic I proposed to compute variance of the estimator for N for model $\mathcal{M}_h$ in Chapter 2. It is based on the law of iterated expectations as before, which implies that

$$\begin{aligned} \text{Var}(\hat{N}^\star(\hat{\boldsymbol{\kappa}})) &= E(\text{Var}(\hat{N}^\star(\hat{\boldsymbol{\kappa}})|\mathbf{c})) + \text{Var}(E(\hat{N}^\star(\hat{\boldsymbol{\kappa}})|\mathbf{c})) \\ &= E_{\mathbf{c}}[E_{\boldsymbol{\kappa}}(\text{Var}(\hat{N}^\star(\hat{\boldsymbol{\kappa}})|\mathbf{c}, \boldsymbol{\kappa})))] + E_{\mathbf{c}}[\text{Var}_{\boldsymbol{\kappa}}(E(\hat{N}^\star(\hat{\boldsymbol{\kappa}})|\mathbf{c}, \boldsymbol{\kappa})))] \\ &\quad + \text{Var}(E(\hat{N}^\star(\hat{\boldsymbol{\kappa}})|\mathbf{c})). \end{aligned}$$

The three terms in the last line in the equation are derived identically as equations (2.22), (2.23) and (2.24) with substitution of the new values of $\boldsymbol{\delta}$ by $\boldsymbol{\kappa}$. In consequence, the final form of $\text{Var}(\hat{N}^\star(\hat{\boldsymbol{\kappa}}))$ is given by

$$\begin{aligned} \text{Var}(\hat{N}^\star(\hat{\boldsymbol{\kappa}})) \approx\ & \frac{\partial \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}'} \text{Var}(\hat{\boldsymbol{\kappa}} - \boldsymbol{\kappa}) \frac{\partial \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}} + 2 \frac{\partial \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}'} \text{Var}(\boldsymbol{\kappa}) \frac{\partial \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}} \\ & + 6\tilde{\boldsymbol{\kappa}}' \frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}} \text{Var}(\boldsymbol{\kappa}) \frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}} \tilde{\boldsymbol{\kappa}} + 2tr\Big(\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}} \text{Var}(\boldsymbol{\kappa}) \frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}} \text{Var}(\boldsymbol{\kappa})\Big) \\ & + tr\Big(\frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}} \text{Var}(\hat{\boldsymbol{\kappa}} - \boldsymbol{\kappa}) \frac{\partial^2 \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}' \partial \tilde{\boldsymbol{\kappa}}} \text{Var}(\boldsymbol{\kappa})\Big) + \sum_{i=1}^{n} \frac{1}{\pi_i^2} - \frac{1}{\pi_i}, \end{aligned}$$

where $\text{Var}(\hat{\boldsymbol{\kappa}} - \boldsymbol{\kappa})$ is equal to $\text{Var}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})$ after removing the row and column corresponding to the behavioural effect, γ, which does not affect the HT estimator,

$$\frac{\partial \hat{N}(\tilde{\boldsymbol{\kappa}})}{\partial \tilde{\boldsymbol{\kappa}}} = -\sum_{i=1}^{n} p_{i.}\Big(\frac{1}{\pi_i^2} - \frac{1}{\pi_i}\Big)\mathbf{w}_i,$$

and I approximate the variance by substituting $\hat{\boldsymbol{\kappa}}$ for $\boldsymbol{\kappa}$ in the final form. The variance is used to compute CIs for N, when the model of interest is any of $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$, as described below.

The two assumptions, the normal distribution for N and the log-normal distribution for $N - n$, as in Chapter 2, are the basis of computing CIs for N through the models $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$. The assumption of the normal distribution is available by the asymptotic property of the HT estimator, as shown by Huggins (1989), and provides Wald CI,

$$\hat{N}^{\star}(\hat{\boldsymbol{\delta}}) \pm 1.96\sqrt{\mathrm{Var}(\hat{N}^{\star}(\hat{\boldsymbol{\delta}}))}\,,$$

when the confidence level is 95%. As an alternative to the Wald CI, I again consider a second CI based on the assumption of the log-normal distribution for $N - n$,

$$(n + \hat{f}_0/C, n + \hat{f}_0 \times C)\,,$$

where

$$C = \exp\left\{1.96\left[\log\left(1 + \frac{\mathrm{Var}(\hat{N}^{\star}(\hat{\boldsymbol{\kappa}}))}{\hat{f}_0^2}\right)\right]^{1/2}\right\},$$

and $\hat{f}_0 = \hat{N}^{\star}(\hat{\boldsymbol{\kappa}}) - n$. The log-normal CI has been recommended for the cases that N is right-skewed rather than symmetric, and in my simulation study, the log-normal CI outperforms the Wald CI under some specific conditions.

3.3 Simulation Study

I conducted a simulation study to check the performance of my approach based on the h-likelihood for fitting $\mathcal{M}_{tbh}$, the most general model among the models I consider, and estimating the population size. In this simulation study, I generated $n_{sim} = 1000$ data sets from $\mathcal{M}_{tbh}$, in which the linear predictor is defined as in equation (3.2), where $\boldsymbol{\beta} = (\alpha', \beta_h, \gamma)'$, and $\mathbf{x}_{it} = (\mathbf{1}_{t-1}, 0, \mathbf{1}_{T-t}, x_i, A_{it})'$ with x_i being a individual covariate whose effect is described by β_h. I considered 16 scenarios by setting (α_t, γ) at one of the four combinations, $(-2.2, 0.81)$, $(-1.39, 0.54)$, $(-0.85, 0.44)$ and $(-0.41, 0.41)$, and σ_v at one of the four values, 0.1, 0.4, 0.7 and 1.0. In all scenarios, x_i is generated from $U(-1, 1)$ for each i, and the fixed effect β_h is always set as 0.7. The number of sampling occasions was $T = 8$, and the true population size was $N = 250$. The four parameter combinations $(\alpha_t, \gamma) = (-2.2, 0.81)$, $(-1.39, 0.54)$, $(-0.85, 0.44)$ and $(-0.41, 0.41)$ are equivalent to setting the median of the initial capture and recapture probabilities, $(\bar{p}_{it}^{\dagger}, \bar{p}_{it}^{\ddagger})$ as $(0.1, 0.2)$, $(0.2, 0.3)$, $(0.3, 0.4)$ and $(0.4, 0.5)$ respectively.

Table 3.2: Percent coverage for N based on both the log-normal CI and Wald CI as a function of σ_v and α_t which in turn defines the median initial capture probability, $\bar{p}_{it}^{\dagger}$.

			$\alpha_t(\bar{p}_{it}^{\dagger})$			
(T, N)	CI type	σ_v	−2.2 (0.1)	−1.39 (0.2)	−0.85 (0.3)	−0.41 (0.4)
(5, 100)	log-normal	0.1	80.6	92.8	94.2	92.4
		0.4	81.6	91.5	93.3	93.9
		0.7	81.5	89.8	92.0	90.0
		1.0	81.0	90.3	90.6	88.5
	Wald	0.1	66.5	84.0	86.8	89.7
		0.4	68.0	81.7	84.8	89.2
		0.7	67.0	79.5	83.1	83.4
		1.0	64.7	78.7	81.9	81.4
(8, 250)	log-normal	0.1	95.5	95.5	92.2	89.4
		0.4	93.2	96.4	90.9	83.9
		0.7	93.9	96.1	86.9	81.8
		1.0	94.0	95.6	91.1	85.2
	Wald	0.1	90.1	93.1	93.7	93.4
		0.4	87.3	93.3	94.4	92.0
		0.7	87.4	92.8	93.6	92.9
		1.0	85.6	93.3	95.0	94.9

The three quantities provided for assessing the performance of the simulation in Chapter 2, RB, RRMSE and CP in equations (2.25), (2.26) and (2.27), are again computed in the simulation study. As in Chapter 2, I compute both the Wald CI and the log-normal CI with confidence level set at 95% for comparison.

The results of the RB and RRMSE for each parameter estimate are illustrated in Figures 3.1 and 3.2 when $(T, N) = (5, 100)$, and Figures 3.3 and 3.4 when $(T, N) = (8, 250)$. Regardless of the setting of (T, N), any parameter estimate produces RB closer to 0 as α_t increases. RRMSEs for most parameter estimates also get closer to 0 as α_t increases, though this pattern does not apply to both $\hat{\gamma}$ and $\hat{\alpha}_t$ if $(T, N) = (8, 250)$, those with $\hat{N}$ if $(T, N) = (5, 100)$. It is surprising that the RB for $\hat{\gamma}$ does not depend on the change in σ_v, while it gets close to 0 as α_t increases. The pattern of RRMSE for $\hat{\alpha}_t$ seems not to be monotonic as a function of α_t and is smallest when $\alpha_t = -1.39$ and largest when $\alpha_t = -0.41$, the largest value tested in this simulation study.

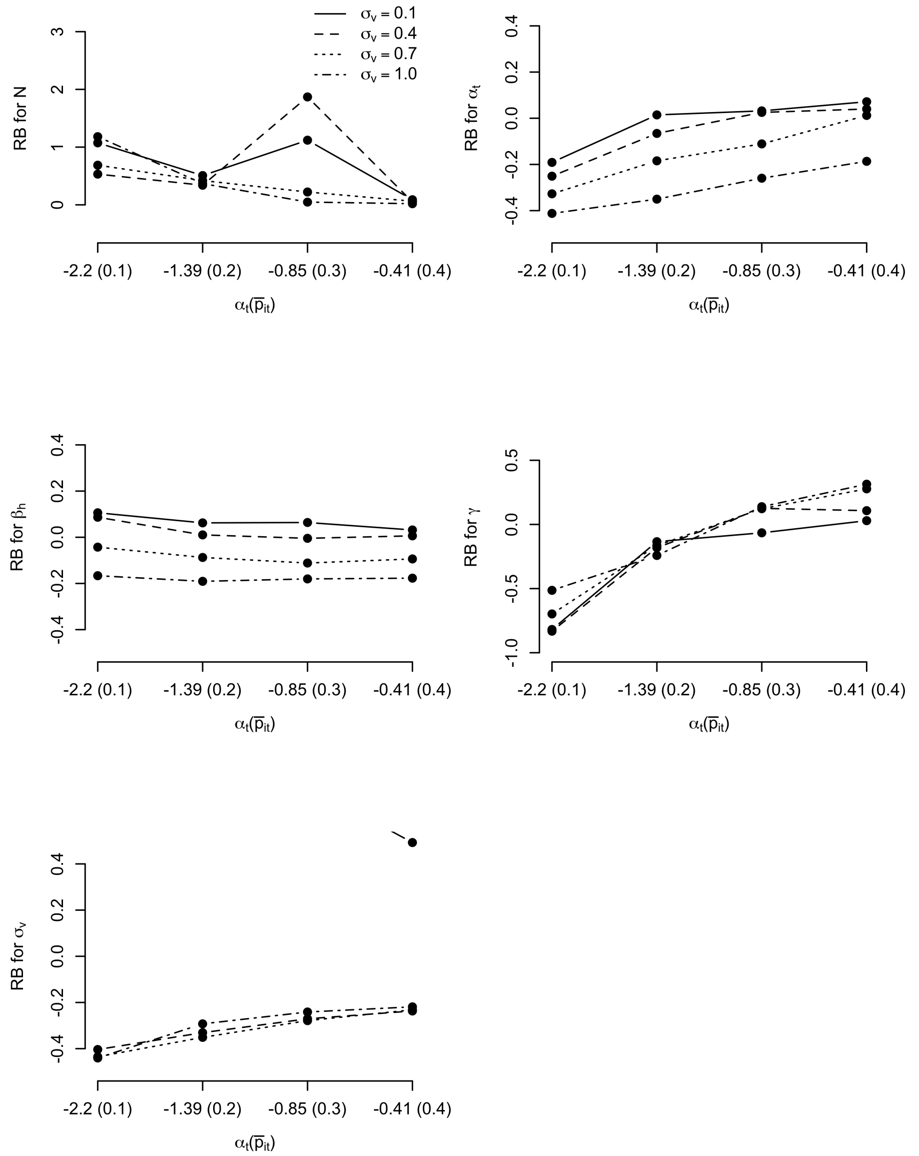

Figure 3.1: Relative Bias (RB) for $\hat{N}$ (top left), $\hat{\alpha}_t$ (top right), $\hat{\beta}_h$ (middle left), γ (middle right) and $\hat{\sigma}_v$ (lower right), when $T = 5$, and $N = 100$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values.

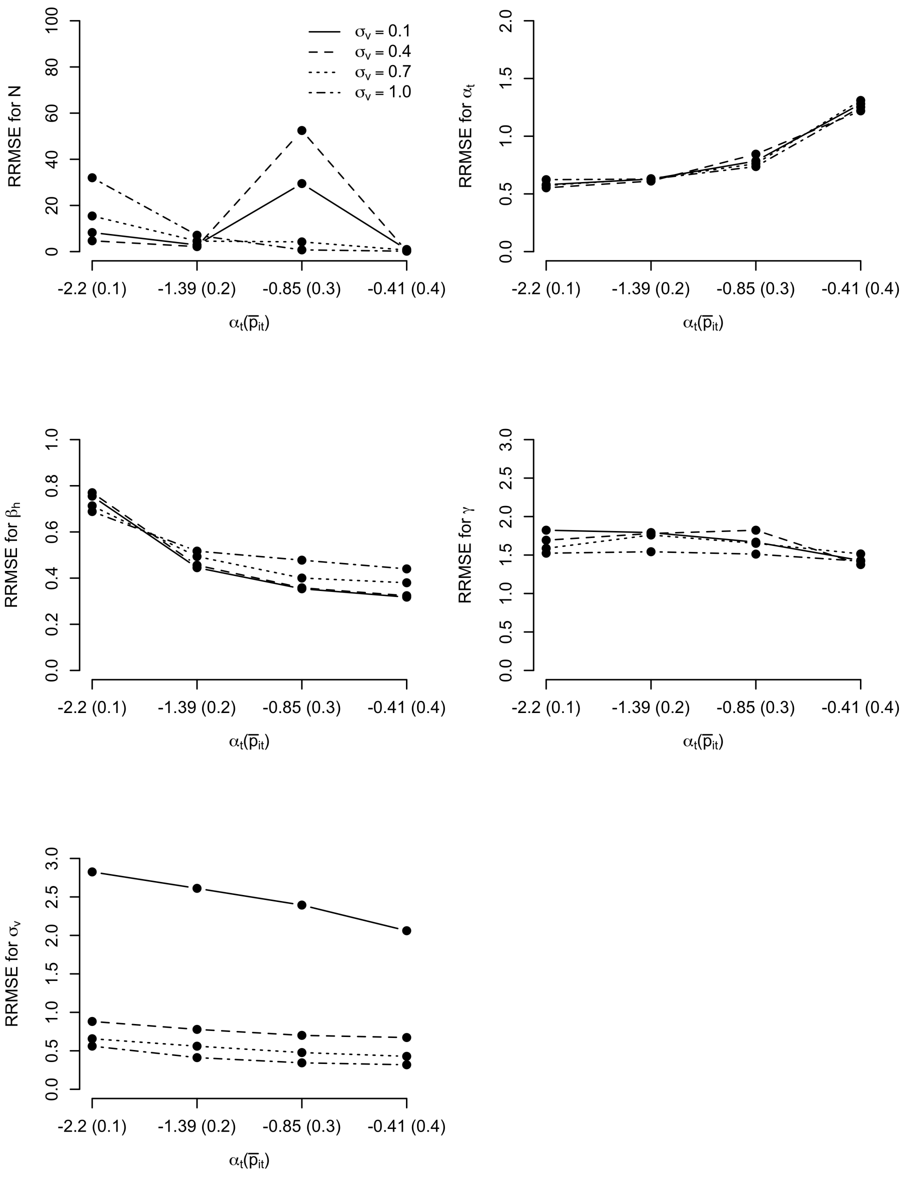

Figure 3.2: Relative root mean square error (RRMSE) for $\hat{N}$ (top left), $\hat{\alpha}_t$ (top right), $\hat{\beta}_h$ (middle left), γ (middle right) and $\hat{\sigma}_v$ (lower right), when $T = 5$, and $N = 100$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values.

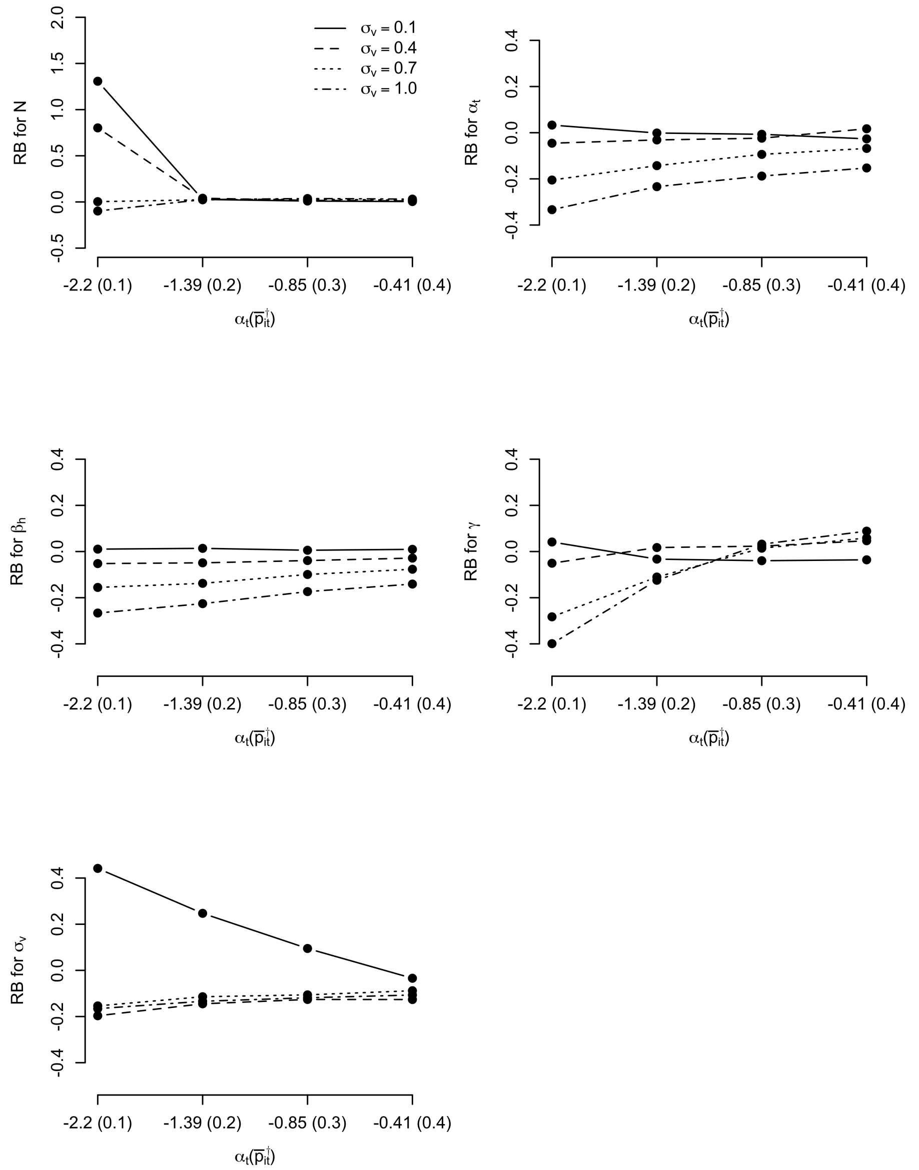

Figure 3.3: Relative Bias (RB) for $\hat{N}$ (top left), $\hat{\alpha}_t$ (top right), $\hat{\beta}_h$ (middle left), γ (middle right) and $\hat{\sigma}_v$ (lower right), when $T = 8$, and $N = 250$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values.

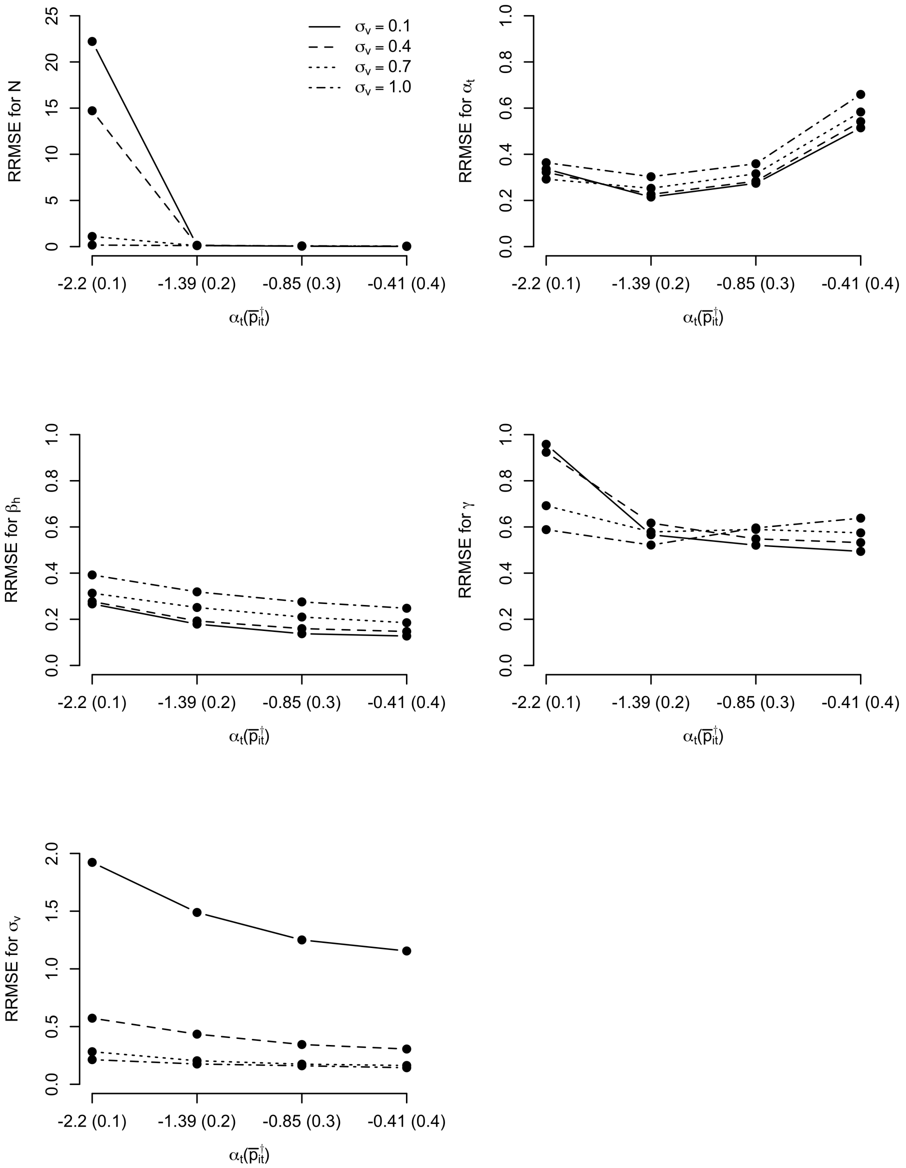

Figure 3.4: Relative root mean square error (RRMSE) for $\hat{N}$ (top left), $\hat{\alpha}_t$ (top right), $\hat{\beta}_h$ (middle left), γ (middle right) and $\hat{\sigma}_v$ (lower right), when $T = 8$, and $N = 250$. The legend describing the linetypes for denoting the four values of σ_v in the upper left plot is also applied to the other three plots. The scales of the y-axis of some plots were changed to accommodate some extreme values.

The RB for two parameters, $\hat{\beta}_h$ and $\hat{\alpha}_t$, approached 0 as σ_v decreased, while α_t remains fixed, and the RB for the remaining parameters either remained constant or increased in magnitude. Nonetheless, the RB and RRMSE for the most important parameter estimate $\hat{N}$ seem to be small when $\alpha_t > -2.2$ with the exception that $\alpha_t = -0.85$ and $(T, N) = (5, 100)$. They seem to be high when $\alpha_t = -2.2$, and further be the worst when $\sigma_v = 0.1$.

The results of the CP for the log-normal and Wald CIs for N are provided in Table 3.2. I observed that the log-normal CI outperforms the Wald CI under all scenarios when $(T, N) = (5, 100)$, and vice versa under all scenarios with $(T, N) = (8, 250)$ except the cases that $\alpha_t = -2.2$. The log-normal CI provided CPs close to the ideal value 95% in two cases: when $\alpha_t > -1.39$ and $\sigma_v < 0.7$ along with $(T, N) = (5, 100)$, and when $\alpha_t <= 1.39$ along with $(T, N) = (8, 250)$. Meanwhile, the Wald CI fails to cover N too often under all scenarios with $(T, N) = (5, 100)$ and all the scenarios when $\alpha_t = -2.2$ and $(T, N) = (8, 250)$. This suggests that the distribution of the HT estimator is almost symmetric when $\alpha \geq -2.2$ with a moderate sample size n, and more likely right-skewed, otherwise.

3.4 Application

I have applied my approach to the same MR data for snowshoe hares (*Lepus americanus*) considered in Chapter 2 for fitting additional models $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$. As described in Chapter 2, two additional methods including the frequentist approach fitting based on numerical integration and the Bayesian approach through MCMC sampling were also applied to fit these MR models and estimate the population size. The software used for implementing all the methods remained the same as before, and the settings were updated as the follows:

1. h-likelihood: Algorithm 2 was applied for fitting the three MR models. The initial values for all fixed effects are again obtained from the VGAM package in R, and I initially set $v_i = 0$ for $i = 1, ..., n$, and $\sigma_v = 0.01$. I found that changes in any component of the initial values did not affect the results of parameter estimates. R code for my method is included in Appendix B.

2. frequentist (numerical integration): Program MARK is employed to fit all three MR models with the same settings for GHQ. The initial values were set at -0.1 for all parameters including σ_v on log scale.

3. Bayesian: MCMC sampling with data augmentation was performed to obtain the posterior densities of all parameters of the three MR models. When MCMC is implemented

in JAGS, the models were re-parameterized so that for each $i = 1, ..., M$, and $t = 1, ..., T$,

$$z_i \sim \text{Bernoulli}(\psi)\,,$$

$$\text{logit}(p_{it}) = \begin{cases} \mu + \alpha_t^* + v_i & \text{for } \mathcal{M}_{th} \\ \mu + \alpha^* + \gamma + v_i & \text{for } \mathcal{M}_{bh} \\ \mu + \alpha_t^* + \gamma + v_i & \text{for } \mathcal{M}_{tbh} \end{cases}$$

and

$$y_{it}|z_i \sim \text{Bernoulli}(z_i p_{it})$$

where M is the super-population size and z_i indicates whether or not pseudo-individual i exists within the population. The true population size is treated as a derived parameter, $N = \sum_{i=1}^{M} z_i$. The linear predictors in the second line are alternative forms of the linear predictors in equation (2.3) such that $\mu + \alpha^* = \alpha$ for $\mathcal{M}_{bh}$, and $\mu + \alpha_t^* = \alpha_t$ for $\mathcal{M}_{th}$ and $\mathcal{M}_{tbh}$. Prior distributions were chosen to be identical to those given by King et al. (2009, pg. 347-350):

$$\begin{aligned} \mu &\sim \mathcal{N}(0, 10) \\ \alpha \text{ or } \alpha_t &\sim \mathcal{N}(0, \sigma_\alpha^2) \\ \gamma &\sim \mathcal{N}(0, \sigma_\gamma^2) \\ v_i &\sim \mathcal{N}(0, \sigma_v^2) \\ \sigma_\alpha^2, \sigma_\gamma^2, \sigma_v^2 &\sim \Gamma^{-1}(4, 3) \end{aligned}$$

and

$$\psi \sim \text{Beta}(0.001, 1)\,.$$

The initial values used for MCMC were: $\mu = 1.0$, α^* or $\alpha_t^* = 0.5$, $\gamma = 0$, $\sigma_\alpha, \sigma_\gamma, \sigma_v = 1$, and $\psi = 0.1$, and a single chain was sampled with $2e^6$ iterations with thinning interval of 20.

As four different models, including $\mathcal{M}_h$ in Chapter 2, were fitted with the same data, model comparison was performed within each method. Specifically, I computed the conditional Akaike information criteria (cAIC) for the h-likelihood (Lee and Nelder, 1996), AIC corrected (AICc) for the frequentist approach using numerical integration (Hurvich and Tsai, 1989), and Watanabe-AIC (WAIC) for the Bayesian method (Vehtari et al., 2017). These quantities cannot be compared across the methods (e.g., comparing two fits of $\mathcal{M}_h$ by the methods based on h-likelihood and numerical integration) and so were only used to identify the best model within each method.

Table 3.3 provides the results of estimating the population size by applying each of the three methods. In general, I found that the point estimates of N were quite similar across the methods, even though the values were more varied from Bayesian method than the other two methods. The frequentist approach via numerical integration and the h-likelihood approach provided almost identical point estimates of N for all models except for $\mathcal{M}_{tbh}$. Point estimates of N were lower for the Bayesian approach than the other two methods for the three MR models that extend from $\mathcal{M}_h$: $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$. The interval estimate of N is narrower for the Bayesian approach than this for the other two approaches, except for the simplest model $\mathcal{M}_h$. Point estimates of the variance of the random effect of the capture probability were consistently smaller for the h-likelihood method, but this seems not to influence the point and interval estimates of N. When the MR models are compared within each method, $\mathcal{M}_{bh}$ was always selected as the top model.

3.5 Discussion

In this chapter, I discuss my approach using the h-likelihood for fitting the MR models extended from the previous chapter by allowing for the capture probability to be dependant on time and behavioural effects as well as individual heterogeneity. I constructed the h-likelihood based on data conditional on individuals captured at least once for each of the three new models, $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$, in which the capture probability is either time-varying or differs between individuals that have and have not been captured previously or both, and I provided the fitting algorithm for these models based on the IRLS algorithm and the bias correction for the h-likelihood estimates. The population size was subsequently estimated by the HT estimator, substituting all estimates of parameters and random effects obtained from the fitting algorithm in place of the true values. The results of the simulation study and the application to data for snowshoe hares demonstrate that my approach is a valid method to fit the extended MR models and that it provides results that are comparable with previous approaches based on integration in the marginal likelihood by quadrature or sampling methods.

As in the previous chapter, one unexpected result I observed is the unsuccessful estimation of σ_v, which tends to cause non-negligible bias when its true value is known to be very small. In the simulation study with the model $\mathcal{M}_{tbh}$, Figures 3.3 and 3.4 clearly show that the smallest value of σ_v (= 0.1) provided the largest bias among all the scenarios, observed at almost 400%. It is also surprising that the estimate of σ_v did not improve significantly even when the median of the true capture probability increased so that more individuals would be captured, and so the sample size would be larger on average. This may imply that the cause of the bias may not be due to the lack of information for σ_v from data, but due to the method used to estimate σ_v through the h-likelihood. In fact, Noh and Lee (2007) observed similar issues when estimating the dispersion parameters for standard GLMMs from the APHL and thus proposed the APHL based on a higher-order Laplace approximation. I did not consider their method as complex mathematical work (e.g., fourth-order derivatives of $\boldsymbol{\mu}_i$) is required to maximize the APHL and because the other parameter estimates, most importantly the estimate of N, did not seem to be severely affected by the bias of $\hat{\sigma}_v$.

Another important result is that the HT estimator may be highly variable and may provide estimates much larger than the true population size when the capture probability is low. I noticed in my simulation study that when the time effect estimate was much lower than the true value of α (e.g., $\hat{\alpha} = -4.0$ while $\alpha = 2.2$), the population size tended to be estimated as an extremely large value. This occurs due to the expression of the HT estimator, to which the inverse of the probability of being captured at least once for each individual contributes. Specifically, when the capture probability for an individual is extremely low (≈ 0), the probability that this individual is captured at least once is overly small as well, while its inverse may contribute to the HT estimator as an extremely large value (e.g., if $p_i = 1e^{-5}$ and $T = 6$, then $1/\pi_i \approx 166667$). This is one main drawback of the HT estimator, and it is suggested that the sample size is increased by adding extra sampling occasions if the original sampling size is small; thus, the chance that an individual is captured at least once becomes higher.

So far, all the MR models I have considered in the previous chapter and this chapter assume a linear relationship between the capture probability and individual covariates. In the next chapter, I consider extending my approach to MR models allowing for the capture probability to be modelled as a non-linear function of individual covariates.

Table 3.3: Results of estimating the population size of snowshoe hares. Three methods based on: h-likelihood, numerical integration via program MARK and Bayesian method (using MCMC) via JAGS are applied to fit $\mathcal{M}_h$, $\mathcal{M}_{th}$, $\mathcal{M}_{bh}$ and $\mathcal{M}_{tbh}$ and estimate the population size. For the methods based on the h-likelihood and numerical integration, two CIs: log-normal CI (left) and Wald CI (right), are reported. For model comparison, values of model measurements according to the methods, cAIC for the h-likelihood, AICc for numerical integration, and WAIC for Bayeisan method, are included.

Method	Model type	$\hat{N}$	95% CI for N		$\hat{\sigma}_v$	Interval	Model
			log-normal	Wald		estimate of $\hat{\sigma}_v$	measure
h-likelihood	$\mathcal{M}_h$	94.0	[80, 124]	[73, 115]	0.78	[0.63, 0.92]	531.5
	$\mathcal{M}_{th}$	90.2	[82, 136]	[74, 124]	0.85	[0.69, 1.01]	530.9
	$\mathcal{M}_{bh}$	87.4	[76, 113]	[70, 105]	0.89	[0.73, 1.06]	515.9
	$\mathcal{M}_{tbh}$	92.2	[72, 199]	[43, 142]	0.88	[0.71, 1.04]	528.1
numerical integration	$\mathcal{M}_h$	91.7	[76, 137]	[64, 119]	0.92	[0.48, 1.76]	516.4
	$\mathcal{M}_{th}$	91.8	[76, 137]	[64, 119]	0.94	[0.49, 1.78]	522.6
	$\mathcal{M}_{bh}$	81.7	[72, 114]	[63, 100]	0.98	[0.55, 1.78]	515.6
	$\mathcal{M}_{tbh}$	82.6	[71, 137]	[64, 119]	0.98	[0.53, 1.81]	524.3
Bayesian (MCMC)	$\mathcal{M}_h$	94.4	[77, 126]		0.95	[0.63, 1.42]	638.1
	$\mathcal{M}_{th}$	75.3	[70, 83]		1.09	[0.62, 2.12]	610.2
	$\mathcal{M}_{bh}$	74.0	[69, 84]		1.03	[0.60, 1.93]	598.3
	$\mathcal{M}_{tbh}$	80.0	[69, 113]		1.16	[0.63, 2.25]	623.2

Chapter 4

Penalized Spline Approach to MR models with heterogeneity via H-likelihood

4.1 Introduction

In a MR experiment, the effect of individual variation on the capture probability may be quite complicated. As an example, in a study that estimates the number of mountain pygmy possums inhabiting in Mount Hotham, Australia (Heinze et al., 2004), the body weight of possums had a non-linear relationship with the capture probability. In several studies analyzing data collected from this study, the capture probability was estimated to be highest when the body weight (g) was between 37 and 40. Its estimate exponentially dropped as the body mass separated from this range. Fitting a MR model assuming a monotone relationship between the capture history observed and the body weight would not capture this pattern, resulting in severe biases in the parameter estimate for the effect of the body weight and the estimate of the population size.

One solution to this issue is to build non-linear relationships by introducing polynomial terms to the MR models. Continuing with the example of the study regarding the mountain pygmy possum, Huggins and Hwang (2007) fitted the MR model with heterogeneity, where the capture probability was linked to multiple polynomial regression models, each of which was defined for different ranges of body weight. The resulting fits of these local models formulated the estimated capture probability much closer to a quadratic rather than linear function of the body weight. A challenge of this approach is to choose the order of each local polynomial, which was fixed at two in this study. However, in practice, it is unknown if this order is adequate to describe the true relationship between the body weight and the underlying capture probability. A common statistical method to choose the order of the local polynomial is comparing model assessment quantities (e.g., AIC) computed for a set of various models fitted with various values of the order. However, this can be time consuming either when data are large or when the number of the models to be fitted is large.

My solution, as well as the solution of Stoklosa and Huggins (2012), is based on an extension of MR models to more complicated models whose structure mimics the framework of GAMs. Instead of expanding the linear predictor of the MR models by adding polynomial terms, employing the framework of GAMs allows the linear predictor to be modelled as a sum of the flexible functions applied to each covariate. Hence, the expansion generalizes the polynomial expression of the linear predictor to more complex functions. Each function is often re-expressed as a sum of basis functions, fixed before the analysis, so that the linear predictor remains as a linear combination of a new set of covariates generated by these basis functions. Here, my focus is B-splines, though many other sets of basis functions can be considered. As discussed in Chapter 1, however, incorporating B-splines can cause overfitting, as the dimension of parameters associated with the new covariates is often rather high. To solve this problem, adding a penalized term to the likelihood function is essential. This results in the MR models being regarded as a type of random effects models, where random effects are defined as coefficients (parameters) attached to new covariates and following a multivariate normal distribution with the variance–covariance matrix depending on smoothing parameters. Consequently, some classical statistical methods for fitting random effects models, such as the MLE by maximizing the marginal likelihood, can be applied to fit the GAM-like MR models with the cross-validation method to determine smoothing parameters (Stoklosa et al., 2011; Yee et al., 2015).

In this chapter, I present the h-likelihood approach to fit the basic MR model for modelling individual heterogeneity, as in Chapter 2, where the linear predictor is extended to B-splince functions, with the penalization of parameters. I first construct the likelihood function of the MR model with a penalized term based on data conditional on individuals captured at least once, which can be regarded as the h-likelihood of a model belonging to the class of GLMMs. Hence, the fitting algorithm based on the h-likelihood for GLMMs proposed by Lee and Nelder (1996, 2001) is directly applied to fit the MR model. The estimation of the population size is subsequently obtained by the HT estimator by substituting parameters estimates obtained from the h-likelihood. To demonstrate my approach, I provide a simulation study with multiple scenarios and apply my approach to real MR data for mountain pygmy possums to estimate their population size.

Table 4.1: Summary of notations used in Chapter 4.

Notation	Definition
N	Unknown population size
n	Number of individuals captured
T	Number of sampling occasions
p	Number of individual covariates
$y_{i.}$	Number of times that individual i is captured
p_i	Capture probability for individual i
α^*	Intercept parameter
$\mathbf{v}^*$	Vector of parameters associated with basis functions
x_{ij}	the j-th individual covariate for individual i
$\mathbf{z}_i^*$	Covariate vector generated by basis functions applied to x_{ij}
$\mathbf{Z}^*$	Design matrix with the i-th row $\mathbf{z}_i^*$
$\boldsymbol{\theta}$	Vector of all unknown quantities in the h-likelihood
$\boldsymbol{\delta}$	Vector of fixed and random effects in the h-likelihood
$\mathbf{I}_a$	$a \times a$ identity matrix
$\mathbf{0}_a$	Vector of 0s with dimension of a
$\mathbf{1}_a$	Vector of 1s with dimension of a

4.2 Method

4.2.1 Description of MR model: $\mathcal{M}_s$ with smoothing functions

I consider the basic MR model for modelling individual heterogeneity by some individual covariates and allowing for non-linearity by applying a smoothing function to each of the individual covariates. The structure of the model originates from that of $\mathcal{M}_h$, proposed by Otis et al. (1978), where no individual covariates were initially included, and the capture probability is treated as a random variable for each individual. Huggins (1989) extended $\mathcal{M}_h$ to the individual covariates without randomizing the capture probability and describes the relationship between the response variable, the capture history, and the individual covariates only through a linear combination of them. Stoklosa and Huggins (2012) broadened this linear relationship by using B-spline, smoother commonly applied to relax the assumption of linearity in many statistical models. In this chapter, I consider the same model of Stoklosa and Huggins (2012) and denote the model by $\mathcal{M}_s$, where the subscript s stands for spline, throughout this chapter.

The framework of $\mathcal{M}_s$ is described as follow using the mathematical notations given in Table 4.1. The observed response variables for the model are $y_{i.}$ for $i = 1, ..., n$ and assumed to follow the binomial distribution with the same density form in equation (2.1). The capture probability, p_i, is linked to a non-linear function for which I specify p_i by

$$\text{logit}(p_i) = \alpha + \sum_{j=1}^{p} f_j(x_{ij}) \, , \tag{4.1}$$

where $f_j(\cdot)$ are smoothing functions. Specifically, I make use of B-splines (de Boor, 1971) so that each function is written as the sum of pre-defined basis functions,

$$f_j(x_{ij}) = \sum_{l=1}^{q_j} v_{jl} b_{jl}(x_{ij}) \, ,$$

where $b_{jl}(\cdot)$ are the so-called basis functions with coefficients v_{jl}, $l = 1, \ldots, q_j$. Computation of the basis functions can be readily provided by modern software package and is defined by the recursive expression

$$b_{jlw}(x_{ij}) = \frac{x_{ij} - r_{jl}}{r_{j,l+w} - r_{jl}} b_{jl,w-1}(x_{ij}) + \frac{r_{j,l+w+1} - x_{ij}}{r_{j,l+w+1} - r_{j,l+1}} b_{j,l+1,w-1}(x_{ij}) \, , \tag{4.2}$$

such that $b_{jl}(\cdot) = b_{jl,w=d}(\cdot)$, when the degree of spline is set at d, and

$$b_{jl,w=0}(x_{ij}) = \begin{cases} 1 & \text{if } r_{jl} \leq x_{ij} < r_{j,l+1} \\ 0 & \text{otherwise} \end{cases} \, .$$

The points, $r_{j1}, ..., r_{j,q_j+1}$, are called the knots of the spline and are selected within the range of x_{ij}. Using basis functions brings about the re-expression of equation (4.1) by

$$\text{logit}(p_i) = \alpha + \mathbf{z}_i' \mathbf{v} \, , \tag{4.3}$$

in which $\mathbf{v} = (\mathbf{v}_1, ..., \mathbf{v}_p)$ with $\mathbf{v}_j = (v_{j1}, ..., v_{j,q_j})'$, and $\mathbf{z}_i$ is the transformed covariate vector composed by the basis functions such that $\mathbf{z}_i'\mathbf{v} = \sum_{l=1}^{q_j} v_{jl} b_{jl}(x_{ij})$. It is noted that the expression in equation (4.2) provides the property that $\sum_l^{q_j} v_{jl} = 1$, so it is necessary to remove one of the coefficients and its basis function for avoiding non-identifiability of the model (i.e., two or more sets of values for α and $\mathbf{v}$ result in the same value of p_i). Hence, I re-write equation (4.3) by

$$\text{logit}(p_i) = \alpha^* + \mathbf{z}_i^{*'} \mathbf{v}^* \, , \tag{4.4}$$

where $\alpha^* = \alpha + \sum_{j=1}^{p} v_{j,q_j}$, $\mathbf{v}^* = (\mathbf{v}_1^*, ..., \mathbf{v}_p^*)'$ with $\mathbf{v}_j^* = \mathbf{v}_{j(-q_j)}$, and $\mathbf{z}_i^{*'}$ is the transformed covariate vector for $\mathbf{v}^*$. A vector with the subscript $(-q_j)$ denotes the vector with dropping the q_j-th element. In the above equation, the unknown quantities in $\mathbf{v}^*$ can be treated as fixed effects associated with new covariate vector, $\mathbf{z}_i^*$. Consequently, the right-hand side of equation (4.4) can be regarded as a linear predictor linked to p_i, which mimics the form of the linear predictor for the class of GLMs. I shall estimate α^* and $\mathbf{v}^*$ with penalizing $\mathbf{v}^*$ from the h-likelihood based on this connection, as described below.

4.2.2 Connection between penalized likelihood and h-likelihood

To fit $\mathcal{M}_s$, I estimate the parameters, α^* and $\mathbf{v}^*$, in equation (4.4) by maximizing the penalized likelihood defined in equation (1.7) with penalty term given by equation (1.8). Similarly with the conditional h-likelihoods constructed in the previous chapters, I build the penalized likelihood with MR data conditional on individuals captured at least once. As a result, the penalized likelihood for $\mathcal{M}_s$ is given by

$$M(\alpha^*, \mathbf{v}^*, \Lambda; \mathbf{y}_{.}) = \log\left(\prod_{i=1}^{n} f(y_{i.}|y_{i.} > 0; p_i)\right) - \sum_{j=1}^{p} \frac{1}{2}\lambda_j \mathbf{v}_j^{*'} \mathbf{P}_j \mathbf{v}_j^*, \tag{4.5}$$

where $\Lambda = (\lambda_1, ..., \lambda_p)$ is the vector of tuning (or smoothing) parameters, and $\mathbf{P}_j$ are known matrices such that the (a, b)-th element is

$$\left[P_j\right]_{a,b} = \int_{\min(k)}^{\max(k)} b_{ja}^{(d)}(k) b_{jb}^{(d)}(k) dk,$$

where the superscript (d) denotes the d-th order derivative. I assume that the knots are equally spaced and so apply the formula

$$\begin{aligned}\mathbf{v}_j^{*'}\mathbf{P}_j\mathbf{v}_j^* &= \int_{\min(k)}^{\max(k)} v_{ja} b_{ja}^{(d)}(k) v_{jb} b_{jb}^{(d)}(k) dk \\ &= \sum_{l=1}^{q_j-d-1} (\Delta^d v_{jl})^2 = \mathbf{D}_j \mathbf{D}_j'\end{aligned}$$

to compute $\mathbf{P}_j$, as shown by Eilers and Marx (1996). Here, $\Delta^d v_{jl}$ denotes the d-th order difference of v_{jl} about the index l and $\mathbf{D}_j$ is the $q_j \times (q_j - d - 1)$ matrix with the l-th column as the coefficients attached to $v_{j1}, ..., v_{j,l+d}$ in the expression of $\Delta^d v_{jl}$ (e.g., $\Delta^2 v_{jl} = v_{j,l+2} - 2v_{j,l+1} + v_{jl}$ and the l-th column of $\mathbf{D}_j$ is $(1, -2, 1)'$). For example, if $d = 2$ and $p_j = 4$, then $\mathbf{v}_j^{*'}\mathbf{P}_j\mathbf{v}_j^* = v_{j3} - 2v_{j2} + v_{j1}$

so that

$$\mathbf{P}_j = \mathbf{D}_j\mathbf{D}_j' = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & 2 \\ 1 & -2 & 1 \end{pmatrix},$$

where $\mathbf{D}_j = (1, -2, 1)'$. In the penalized likelihood in equation (4.5), the first term within the log function is the conditional likelihood of Huggins (1989), which is equivalent to the likelihood function for a GLM with the components described in Section 2.2.2. The second term in the penalized likelihood is the penalty term based on the L_2-norm and multiplied by the tuning parameter, λ_j which controls the smoothness of the non-linear function of p_i. Since this term is also proportional to the log of the density of multivariate normal distribution for $\mathbf{v}_j$ with mean vector, $\mathbf{0}_{q_j}$, and the variance–covariance matrix, $(\lambda_j\mathbf{P}_j)^{-1}$, the penalized likelihood is proportional to the log of the joint density of $y_{i.}$ and $\mathbf{v}$, which is the log of the h-likelihood for a GLMM. The response variables for this GLMM are observed as $y_{i.}$ that are greater than 0, and $\mathbf{v}$ are normal random effects through the linear predictor defined in the same form of equation (4.4). As this h-likelihood conditions on individuals captured at least once, the h-likelihood corresponds to the conditional h-likelihood, and I denote its log as

$$h_c(\boldsymbol{\theta}, \Lambda; \mathbf{y}_{.}, \mathbf{v}) = M(\alpha^*, \mathbf{v}^*, \Lambda; \mathbf{y}_{.}),$$

so that $\boldsymbol{\theta} = (\alpha^*, \mathbf{v}^{*'}, \Lambda')'$ is estimated by maximizing h_c about $\boldsymbol{\theta}$ through the fitting algorithm derived in the following section.

4.2.3 Fitting algorithm

By treating $\mathcal{M}_s$ as a GLMM, the fitting algorithm of Lee and Nelder (2001) can be applied directly to fit $\mathcal{M}_s$. The algorithm is similar to the algorithm provided for fitting $\mathcal{M}_h$ in Chapter 2, but simpler in that some of the bias correction steps are removed. Specifically, the bias correction of MHLEs for fixed and random effects is not considered in this chapter as I found in my simulation study that these MHLEs have negligible biases in general. The algorithm then consists of iterating two main steps: one estimates all fixed and random effects from the h-likelihood while fixing dispersion parameters at values obtained from the other step, and the other step estimates all dispersion parameters from the APHL while fixing fixed and random effects at values obtained from the previous step. The derivation of these two mains steps is based on the h-likelihood property that $\partial^2 h/\partial\boldsymbol{\theta}\partial\Lambda \approx \mathbf{0}$, as described by Lee and Nelder (2001).

The details of the algorithm for fitting $\mathcal{M}_s$ are provided in Algorithm 3. Step 1 of the

Algorithm 3 Fitting algorithm for $\mathcal{M}_s$

```
Set initial value $\hat{\boldsymbol{\theta}}^{(0)} = (\hat{\boldsymbol{\delta}}'^{(0)}, \hat{\Lambda}'^{(0)})'$
Let $r = 0$
while convergence criterion $\hat{\boldsymbol{\theta}}^{(r)} \approx \hat{\boldsymbol{\theta}}^{(r+1)}$ not met do
    Let $\hat{\boldsymbol{\delta}}^{(r,0)} = \hat{\boldsymbol{\delta}}^{(r)}$;
    Let $t = 0$;
    while convergence criterion $\hat{\boldsymbol{\delta}}^{(r,t)} \approx \hat{\boldsymbol{\delta}}^{(r,t+1)}$ not met do
        (Step 1) Given $\hat{\Lambda}^{(r)}$, solve equation (4.6) for $\hat{\boldsymbol{\delta}}^{(r,t)}$;
        $t \leftarrow t + 1$
    end while
    Let $\hat{\boldsymbol{\delta}}^{(r+1)} = \hat{\boldsymbol{\delta}}^{(r,t)}$
    (Step 2) Given $\hat{\boldsymbol{\delta}}^{(r+1)}$, obtain $\hat{\Lambda}^{(r+1)} = (\lambda_1^{(r+1)}, ..., \lambda_p^{(r+1)})'$ by fitting $j = 1, ..., p$ gamma
  GLMs, where each model has
      • response variables observed: $\hat{\mathbf{d}}_j^{(r+1)} = \mathbf{D}_j \mathbf{v}_j^{*(r+1)}$ with $\hat{\mathbf{v}}_j^{*(r+1)} = (\hat{v}_{j1}^{(r+1)}, ..., \hat{v}_{j,q_j-1}^{(r+1)})'$
      • prior weight: the $(l + \mathrm{dim}(\mathbf{d}_{j-1}) + 1)$-th diagonal element of equation (3.10) for $\hat{v}_{jl}^{(r+1)}$,
        given $\hat{\boldsymbol{\delta}}^{(r+1)}$ and $\hat{\lambda}_j^{(r)}$
      • linear predictor: $\tau_j = \log(1/\lambda_j)$
end while
```

algorithm solves the normal equation

$$\frac{\partial h_c}{\partial \boldsymbol{\delta}} = \begin{pmatrix} \mathbf{1}_n'(\mathbf{y}_{\cdot} - \boldsymbol{\mu}) \\ \mathbf{Z}^{*'}(\mathbf{y}_{\cdot} - \boldsymbol{\mu}) - \mathrm{bdiag}(\lambda_1 \mathbf{P}_1, ..., \lambda_p \mathbf{P}_p)\mathbf{v}^* \end{pmatrix} = \mathbf{0}_{\mathrm{dim}(\mathbf{v})+1}$$

with respect to $\boldsymbol{\delta} = (\alpha^*, \mathbf{v}^{*'})'$ for which the IRLS equation for the solution is

$$\mathbf{T}'\mathbf{W}^{(t-1)}\mathbf{T}\hat{\boldsymbol{\delta}}^{(t)} = \mathbf{T}'\mathbf{W}^{(t-1)}\mathbf{z}^{*(t-1)}, \tag{4.6}$$

as shown by Lee and Nelder (1996, 2001). Here,

$$\mathbf{T} = \begin{pmatrix} \mathbf{1}_n & \mathbf{Z}^* \\ \mathbf{0}_{\mathrm{dim}(\mathbf{v}^*)-pd} & \mathbf{D}' \end{pmatrix}$$

is the design matrix such that $\boldsymbol{\eta} = \mathbf{T}\boldsymbol{\delta}$ with $\boldsymbol{\eta} = (\eta_1, ..., \eta_n)' = (\mathrm{logit}(p_1), ..., \mathrm{logit}(p_n))$ and $\mathbf{D} = \mathrm{bdiag}(\mathbf{D}_1, ..., \mathbf{D}_p)$,

$$\mathbf{W}^{(t)} = \left. \begin{pmatrix} \mathrm{diag}\left(\frac{\partial \mu_1}{\partial \eta_1}, ..., \frac{\partial \mu_n}{\partial \eta_n}\right) & \mathbf{0}_{n \times \mathrm{dim}(\mathbf{v}^*)-pd} \\ \mathbf{0}_{(\mathrm{dim}(\mathbf{v}^*)-pd) \times n} & \mathrm{ddiag}(\lambda_1 \mathbf{I}_{q_1-d}, ..., \lambda_p \mathbf{I}_{q_p-d}) \end{pmatrix} \right|_{\boldsymbol{\delta}=\hat{\boldsymbol{\delta}}^{(t)}, \Lambda=\hat{\Lambda}}$$

is the t-th weight matrix, and

$$\mathbf{z}^{*(t)} = \left.\begin{pmatrix} \boldsymbol{\eta} + \mathrm{diag}\Big(\frac{\partial\mu_1}{\partial\eta_1}, ..., \frac{\partial\mu_n}{\partial\eta_n}\Big)(\mathbf{y}_{\cdot} - \boldsymbol{\mu}) \\ \mathbf{0}_{\mathrm{dim}(\mathbf{v}^*)-pd} \end{pmatrix}\right|_{\boldsymbol{\delta}=\hat{\boldsymbol{\delta}}^{(t)},\Lambda=\hat{\Lambda}}$$

is the t-th adjusted response variable of the IRLS equation. For the terms within $\mathbf{T}$, $\mathbf{W}^{(t)}$ and $\mathbf{z}^{*(t)}$, I write

$$\boldsymbol{\mu} = E(\mathbf{y}) = (\mu_1, ..., \mu_n)'$$

and have derived that

$$\mu_i = \frac{Tp_i}{\pi_i}$$

and

$$\frac{\partial\mu_i}{\partial\eta_i} = \frac{Tp_i(1-p_i)}{\pi_i} - \frac{T^2p_i^2(1-\pi_i)}{\pi_i^2}.$$

The IRLS algorithm repeatedly solves equation (4.6) about $\hat{\boldsymbol{\delta}}^{(t)}$ until convergence is achieved for the solution of equation (4.6). In Step 2, the APHL

$$h_c^A(\Lambda; \mathbf{y}_{\cdot}, \hat{\boldsymbol{\delta}}) = h_c - \frac{1}{2}\log\Big[\det\Big(-\frac{1}{2\pi}\frac{\partial^2 h_c}{\partial\boldsymbol{\delta}\partial\boldsymbol{\delta}'}\Big)\Big]\Big|_{\boldsymbol{\delta}=\hat{\boldsymbol{\delta}}},$$

is maximized as suggested by Lee and Nelder (2001), which provides the set of the normal equations,

$$\frac{\partial h_c^A}{\partial\tau_j} = \sum_{l=1}^{q_j-d-1} \frac{\partial\lambda_j^*}{\partial\tau_j}\frac{d_{jl}^2 - (1-q_{jl})\lambda_j^*}{\lambda_j^{*2}}$$

for $j = 1, .., p$ as functions of $\tau_j = \log(\lambda_j^*)$. Here $\lambda_j^* = 1/\lambda_j$, d_{jl} is the l-th element of $\mathbf{d}_j = \mathbf{D}_j\mathbf{v}_j^*$, and q_{jl} is the $(l + \mathrm{dim}(\mathbf{d}_{j-1}) + 1)$-th diagonal element of

$$\mathbf{T}\Big(-\frac{\partial^2 h_c}{\partial\delta\partial\delta'}\Big)^{-1}\mathbf{T}'\mathrm{bdiag}\Big(\frac{\partial^2 h_c}{\partial\mathbf{d}_1\partial\lambda_1}, ..., \frac{\partial^2 h_c}{\partial\mathbf{d}_p\partial\lambda_p}\Big) \tag{4.7}$$

(note that $\mathrm{dim}(\mathbf{d}_0) = 0$). In equation (4.7), the derivatives of h_c are given by

$$-\frac{\partial^2 h_c}{\partial\delta\partial\delta'} = \mathbf{T}'\mathbf{W}\mathbf{T}$$

and

$$\frac{\partial^2 h_c}{\partial\mathbf{d}_j\partial\lambda_j} = \lambda_j\mathbf{I}_{\mathrm{dim}(\mathbf{d}_j)}.$$

Lee and Nelder (2001) showed that each normal equation can be viewed as the normal equation for a gamma GLM such that the response variables are observed as d_{jl} for $l = 1, ..., q_j - d - 1$, each of which has the prior weight, q_{jl}, and the linear predictor is $\tau_j = \log(\lambda_j^*)$. I fit these gamma GLMs through the IRLS algorithm, which can be implemented through existing routines in most modern software.

To compute the variance–covariance matrix of $\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$, I apply the asymptotic property of the h-likelihood described in Chapter 1. The variance–covariance matrix is given by

$$\mathrm{Var}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}) = \begin{pmatrix} \mathrm{Var}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) & \mathbf{0}_{\mathrm{dim}(\boldsymbol{\delta})} \\ \mathbf{0}'_{\mathrm{dim}(\boldsymbol{\delta})} & \mathrm{Var}(\hat{\lambda}_j) \end{pmatrix},$$

where I have derived that

$$\mathrm{Var}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta}) \approx (\mathbf{T}'\mathbf{W}\mathbf{T})^{-1}|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}\,. \tag{4.8}$$

I have computed $\mathrm{Var}(\hat{\lambda}_j)$ by the delta method, such that $\mathrm{Var}(\hat{\lambda}_j) \approx \lambda_j^2 \times \mathrm{Var}(\hat{\tau}_j)$, and obtained $\mathrm{Var}(\hat{\tau}_j)$ directly from the IRLS algorithm for fitting the gamma GLMs as described above. The variance–covariance matrix can be used to draw Wald-typed inferences through the asymptotic properties of the h-likelihood,

$$\hat{\boldsymbol{\theta}} - \boldsymbol{\theta} \mathrel{\dot\sim} \mathcal{N}(\mathbf{0}_{\mathrm{dim}(\boldsymbol{\theta})}, \mathrm{Var}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})) \tag{4.9}$$

if the sample size, n, is large enough. This property shall be used to estimate N in the next section.

4.2.4 Estimation of population size

As in the previous chapters, I estimate N based on the HT estimator, which depends on the MHLE for $\boldsymbol{\delta}$ obtained through Algorithm 3. Under the same idea of Huggins (1989), who proposed the estimate for N as

$$\hat{N}(\boldsymbol{\delta}) = \sum_{i=1}^{N} \frac{1}{\pi_i} = \sum_{i=1}^{n} \frac{1}{\pi_i},$$

where $c_i = I(y_{i.} > 0)$, if the parameters, $\boldsymbol{\delta}$, are known, my estimator for N is given by

$$\hat{N}(\hat{\boldsymbol{\delta}}) = \sum_{i=1}^{n} \frac{1}{\hat{\pi}_i}$$

with the MHLE, $\hat{\boldsymbol{\delta}}$. Even though $\mathbf{v}$ in $\boldsymbol{\delta}$ is regarded as the vector of random effects when the h-likelihood approach is applied, it is originally the parameters generated by B-spline, and so I ignore the variance of $\mathbf{v}$ as the random effects in computing $\hat{N}$, but account for the uncertainty occurred when obtaining $\hat{\boldsymbol{\delta}}$ in computing the variance of $\hat{N}$. Huggins (1989) showed that the expectation of the HT estimator itself is

$$E(\hat{N}(\boldsymbol{\delta})) = N\,,$$

so that by the first-order Taylor expansion of $\hat{N}(\hat{\boldsymbol{\delta}})$ about $\boldsymbol{\delta}$, that is

$$\hat{N}(\hat{\boldsymbol{\delta}}) \approx \hat{N}(\boldsymbol{\delta}) + \frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}'}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})\,, \tag{4.10}$$

and the h-likelihood property in equation (4.9), the expectation of my estimator is given by

$$E(\hat{N}(\hat{\boldsymbol{\delta}})) \approx N\,.$$

Hence, given a large sample size n, my estimator is approximately unbiased. To compute the variance of my estimator, I apply the variance on the expression in equation (4.10), which provides that

$$\mathrm{Var}(\hat{N}(\hat{\boldsymbol{\delta}})) \approx \mathrm{Var}(\hat{N}(\boldsymbol{\delta})) + \frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}'}\mathrm{Var}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})\frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}}\,, \tag{4.11}$$

where Huggins (1989) showed that

$$\mathrm{Var}(\hat{N}(\boldsymbol{\delta})) = \sum_{i=1}^{n} \frac{1}{\pi_i^2} - \frac{1}{\pi_i}\,,$$

$\mathrm{Var}(\hat{\boldsymbol{\delta}} - \boldsymbol{\delta})$ is provided in equation (4.8), and I derived that

$$\frac{\partial \hat{N}(\boldsymbol{\delta})}{\partial \boldsymbol{\delta}} = -\sum_{i=1}^{n} T p_i \Big(\frac{1}{\pi_i^2} - \frac{1}{\pi_i}\Big)(1, \mathbf{z}_i^{*'})'\,.$$

The variance is computed approximately by substituting $\hat{\boldsymbol{\delta}}$ for $\boldsymbol{\delta}$ in equation (4.11). The approximated variance is then used to obtain CIs of N, as described below.

In the same way that CIs for N are obtained in the previous chapters, I consider two types of the CIs: Wald CI and log-normal CI. The Wald CI is based on the assumption that N asymptotically follows a normal distribution and defined as

$$\hat{N}(\hat{\boldsymbol{\delta}}) \pm 1.96\sqrt{\mathrm{Var}(\hat{N}(\hat{\boldsymbol{\delta}}))}\,,$$

when the confidence level is 95%. Alternatively, the log-normal CI is based on the assumption that $N - n$ approximately follows a log-normal distribution and defined as

$$(n + \hat{f}_0/C, n + \hat{f}_0 \times C)\,,$$

where

$$C = \exp\left\{1.96\left[\log\left(1 + \frac{\mathrm{Var}(\hat{N}(\hat{\boldsymbol{\delta}}))}{\hat{f}_0^2}\right)\right]^{1/2}\right\},$$

and $\hat{f}_0 = \hat{N}(\hat{\boldsymbol{\delta}}) - n$. The log-normal CI has been recommended for the cases that N is right-skewed rather than symmetric. In my simulation study, both CIs are shown to perform well, but the log-normal CI outperforms the Wald CI slightly in terms of the coverage probability.

4.3 Simulation Study

I conducted a simulation study to demonstrate my approach using the h-likelihood to fit $\mathcal{M}_s$ with smoothing functions based on B-spline. In particular, I simulated $n_{sim} = 1000$ data sets from $\mathcal{M}_s$, in which p_i was modelled by a non-linear function as in equation (4.1), where a single individual covariate x_i (the index j is dropped for convenience) was generated from $U(-1, 1)$, and $f(x) = 0.7\sin(\pi x_i)$. Each data set was analysed by fitting $\mathcal{M}_s$ with p_i modelled as in equation (4.4), fixing the degree of the B-spline at $d = 3$ and the number of knots at $q = 11$, for which the space between any two sequential knots was equal. The true values of the parameters in $\mathcal{M}_s$ were set at: $\alpha \in \{-2.2, -1.39, -0..85, -0.41\}$, and $(T, N) = \{(5, 100), (8, 250)\}$, which provided eight scenarios in total. The four divisions of α produce four different levels of the capture probability, backtransformed from $E(\mathrm{logit}(p_i))$, $\bar{p}_i = \mathrm{logit}^{-1}(E(\mathrm{logit}(p_i))) \approx 0.1$, 0.2, 0.3 and 0.4, corresponding to $\alpha = -2.2$, -1.39, -0.85 and -0.41, respectively. The setting of the combination (T, N) expects to generate low and moderate number of individuals captured, n, for $(5, 100)$ and $(8, 250)$, respectively.

To assess the performance of my approach, I computed the RB, RRMSE and CP as provided in equations (2.25), (2.26) and (2.27). As in Chapter 2 and 3, I computed both Wald CI and log-normal CI with the confidence level set at 95%.

The results of the RB and RRMSE for N are illustrated in Figure 4.1. The RB and RRMSE for α were not considered as α itself is not estimable due to the issue of non-identifiability with the B-spline. In general, the population size N was estimated with negligible bias when $\alpha > -2.2$ under both settings of (T, N). The RB ranged between 0.15% and 0.53% for the six scenarios such that $\alpha > -2.2$, but it is the largest and observed at the value 9.21% when $\alpha = -2.2$ and $(T, N) = (8, 250)$, and at the extreme value over 1.0^8%, when $\alpha = -2.2$ and

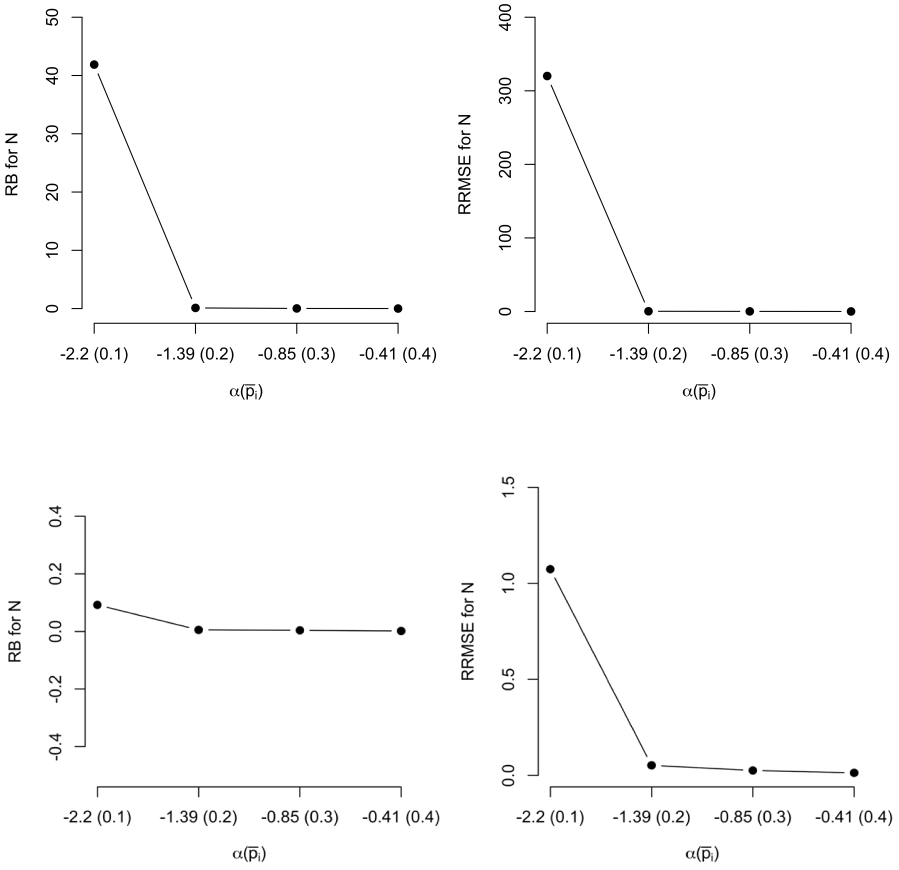

Figure 4.1: Relative bias (RB) (left) and relative root mean square error (RRMSE) (right) for $\hat{N}$ when $(T, N) = (5, 100)$ (upper) and $(T, N) = (8, 250)$ (lower). The scales of the y-axis of the plots for RB and RRMSE were changed to accommodate some extreme values when $\alpha = -2.2$.

$(T, N) = (5, 100)$. The RRMSE ranged between 1.34% and 5.26% when $\alpha > -2.2$, regardless of (T, N) and had an extreme value of 107% and 1.5×10^9% when $\alpha = -2.2$, and $(T, N) = (8, 250)$ and $(5, 100)$, respectively. This indicates that $\hat{N}$ is not accurate when $\alpha = -2.2$ and somewhat alarming, but not too surprising, given that the capture probability is very low and so a larger data set would be required to recover accurate information about the relationship between the capture probability and the covariate.

The results of the CP for log-normal and Wald CIs for N are provided in Table 4.2. All CPs seem to be close to 95% regardless of the type of CI. The log-normal CI slightly outperforms the Wald CI when $\alpha \leq -1.39$ and vice versa when $\alpha > 1.39$. This suggests that the distribution of the HT estimator is more likely right-skewed as α decreases; however, the shape of the distribution seems to negligibly affect the difference in the reliability of the two CIs.

Table 4.2: Percentage coverage for N based on log-normal CI and Wald CI as a function of α which in turn defines the backtransformed capture probability $\bar{p}_i$.

		$\alpha\,(\bar{p}_i)$			
(T, N)	CI type	-2.2 (0.1)	-1.39 (0.2)	-0.85 (0.3)	-0.41 (0.4)
(5, 100)	log-normal	89.9	95.5	91.1	90.4
	Wald	94.0	95.5	95.5	91.2
(8, 250)	log-normal	94.2	94.2	92.8	93.6
	Wald	93.3	92.7	95.7	94.2

4.4 Application

I have applied my approach to analyse MR data for mountain pygmy possum (*Burramys parvus*) from the study of Heinze et al. (2004) conducted at Mount Hotham in the snowfields of Victoria, Australia. The data were collected over $T = 5$ consecutive days of November, 2003. When an individual was captured by traps set in the area that includes the home range of the individual, the body mass was measured when the individual was captured at the first time. In total of $n = 43$ possums captured, 22 of them were captured once, 10 captured twice, 3 three times, 4 four times, and 4 were captured on every occasion. The body mass (g) measured from the possums ranged between 31.0 and 49.0 and with mean and standard deviation 40.5 and 4.3, respectively.

I fitted $\mathcal{M}_s$ with data based on six different approaches for comparison:

1. GLM (no smoothing): I fitted $\mathcal{M}_h$ via the VGAM package in R which uses the IRLS algorithm to maximize the conditional likelihood of $\mathcal{M}_h$ when no smoothing function is applied to the body mass (i.e., $p_i = \alpha + \beta x_i$, where β is a parameter of the body mass of individual i, x_i). This approach was proposed by Yee et al. (2015) who considered $\mathcal{M}_h$ as a GLM and estimated the population size via the HT estimator of Huggins (1989), as defined in equation (1.2).

2. Kernel smoothing: In equation (4.1), the approach of kernel smoothing is non-parametric and estimates the value of $f(x_i)$ by the weighted average of neighboring observed data. I applied this approach of Huggins and Hwang (2007) who particularly used the biweight kernel, $K(w) = (15/16)(1 - w^2)^2$, to obtain $\hat{f}(x_i)$ based on the data conditional on individuals captured at least once. The population size was again estimated via the HT estimator.

3. Backfitting (equivalent degree of freedom (edf) = 2): I applied the backfitting algorithm

(Breiman and Friedman, 1985) to estimate $f(x_i)$, where $f(\cdot)$ was chosen to be a cubic spline smoother, and $\mathcal{M}_s$ was regarded as a GAM through the conditional likelihood. The edf is used as the smoothing parameter in backfitting algorithm and higher edfs allows more flexibility in the relationship between the capture probability and the covariate. I set edf = 2, which seems to be often considered to provide moderate flexibility. The estimates $\hat{p}_i$ are computed with the fitted values $\hat{f}(x_i)$ and substituted into the HT estimator. I implemented this approach through the VGAM package in R that carries out the backfitting algorithm automatically.

4. Backfitting (edf = 12): As described above, the backfitting algorithm with edf = 12 is used to fit $\mathcal{M}_s$ and estimate the population size by the HT estimator. This edf setting allows the function $f(\cdot)$ to be more flexible.

5. P-spline generalized cross-validation (GCV): This approach uses B-spline to fit $\mathcal{M}_s$ with p_i being modelled as in equation (4.3) and maximizes the penalized likelihood in equation (4.5). I applied this approach to fit $\mathcal{M}_s$ with the number of knots set at $q = 15$ and the degree of B-spline $d = 3$. I implemented this approach again using the VGAM package in R that carries out the IRLS algorithm for estimating parameters in the model. This approach selects the optimal value of the tuning parameter, λ, by minimizing the GCV.

6. P-spline h-likelihood: I implemented algorithm 3 in R to fit $\mathcal{M}_s$ and estimate N by the HT estimator, as described in the previous sections. I used the same number of knots and degree of the B-spline in this approach as in the P-spline and GCV approaches. R code for my approach is included in Appendix C.

Table 4.3 presents the results of estimating the population size based on the six approaches. In general, the point estimates but also the interval estimates of the population size are similar for all approaches except for the approach using backfitting with edf = 12. This result implies that allowing for too much flexibility in the function, $f(\cdot)$, is not suitable for describing data as the estimate of the population size is completely nonsensical. Even though I set a large number of knots in the approaches based on the P-spline, by which the model is fitted through GCV or the h-likelihood, the point and interval estimates of the population size are highly comparable to those obtained based on the kernel smoothing and backfitting with edf = 2. This shows that the approaches using the P-spline are able to adapt the flexibility of the curve to the data.

Table 4.3: Results of estimating the population size of mountain pygmy possums. Six approaches for fitting $\mathcal{M}_h$ as: a GLM without a smoothing function on the body mass, a GAM with kernel smoothing, that via backfitting with the equivalent degree of freedom being 2 and 12, that via GCV and my approach using the h-likelihood, where the model is parameterized by basis functions (B-spline) with the degree being 3 and $L2$-norm for the penalization of coefficients of the basis functions. For all approaches, two CIs: log-normal CI (left) and Wald CI (right), are reported.

Method	$\hat{N}$	95% CIs for N	
		log-Normal	Wald
GLM (no smoothing)	49.0	[45, 60]	[42, 56]
Kernel smoothing	55.5	[46, 90]	[37, 74]
Backfitting (edf = 2)	50.9	[46, 65]	[42, 60]
Backfitting (edf = 12)	2857.4	[532, 16250]	[-3237, 8952]
P-spline GCV	58.6	[52, 69]	[51, 67]
P-spline h-likeihood	52.9	[46, 76]	[40, 66]

4.5 Discussion

This chapter considered the basic MR model from Chapter 2, in which individual heterogeneity is modelled by individual covariates, and extended it through a GAM formulation, allowing the capture probability to be described by a non-linear relationship with the individual covariates. I employed B-splines to derive a parametric form that enabled the model to be considered as a GLMM by penalizing the spline coefficients. In this framework, the tuning parameters that control the roughness of the functions of the individual covariates were equivalent to the dispersion parameters in the GLMM. The likelihood function of the model with the added penalization term (i.e., the penalized likelihood) was shown to be equal to the h-likelihood of the GLMM, so I applied the fitting algorithm of Lee and Nelder (2001) that maximizes the h-likelihood of GLMMs via the IRLS algorithm. I demonstrated that my approach provides valid estimates of parameters and population size computed by the HT estimator through the simulation study and the application of my approach to real data for the mountain pygmy possum.

In the simulation study, I found that the performance of my approach was poor when the true capture probability was set as its lowest. This might occur due to a small sample size obtained by the low true capture probability, as discussed in Chapter 2. In particular, the point estimate of the population size has a high bias when the capture probability is low, even though the interval estimates by both Wald and log-normal CIs are highly reliable. This typically indicates that the CIs might be very wide, due to the lack of information in data. Hence, it is suggested to analyze data with a sample size large enough to avoid biased parameter estimates.

My approach has specifically been applied to the basic MR model with GAM formulation but can be easily extended to fit other types of MR models for individual heterogeneity and GAM formulation together. For example, environmental covariates can be included in the expression in equation (4.1) and fit with B-splines to provide flexible and parametric functions that can be regarded as the linear predictor of a GLMM and fitted by the h-likelihood. Moreover, the MR models that include the time and behavioural effects, as in Chapter 3, can be extended to model individual heterogeneity with the GAM structure and fitted fitted by my approach by regarding the MR models as VGLMMs for which I derived the fitting algorithm based on the h-likelihood. For future works, I plan to perform a simulation study for fitting the extended MR models from Chapter 3 with the GAM structure and investigate the performance of my approach.

Chapter 5

Conclusion

The three projects in this thesis all consider problems in modelling individual heterogeneity in the capture probability of models for MR data. In Chapter 2, the capture probability is modelled by random effect describing individual variation, possibly in combination with observed covariates that do not vary by time. This means that the capture probability for each individual is constant, and so the number of times each marked individual is captured is sufficient. This allows me to write the model as an extension of a GLMM that can be fitted by maximizing the conditional h-likelihood. I derive a fitting algorithm based on IRLS, apply the bias correction for the parameter estimates, and then estimate the population size by the HT estimator with estimates obtained from the conditional h-likelihood substituted for the true parameter values. In Chapter 3, I develop my approach using the h-likelihood to fit the extended MR models, which allows additional variation in the capture probability by time-dependant or by individual trap responses. In this case, the capture history for an individual cannot be summarized by a single value; so, it is necessary to consider extensions of VGLMMs. The conditional h-likelihoods for these models are built and maximized to obtain parameter estimates, where the same bias correction method of Chapter 2 is applied. As before, the fitting algorithm for these models is based on the IRLS. In Chapter 4, my objective was to relax the assumption of linearity in the basic model in Chapter 2 by incorporating penalized splines to describe a non-linear relationships between the covariates and the capture probability. The conditional likelihood with a penalizing term of parameters is considered as the h-likelihood of a GLMM and maximized by the fitting algorithm based on the h-likelihood, shown by some authors who considered the general form of GLMMs but not the MR models.

Overall, I conclude that the h-likelihood approach to analyze MR data with individual heterogeneity provides comparable parameter estimates to all the previous methods based on either frequentist or Bayesian approaches. In the analysis of the snowshoe hares and mountain pygmy possum data for the three projects, I found that the h-likelihood approach not only resulted in point estimates and CIs for the population size that were close to those from either of the other approaches, but also computationally efficient. Fitting all types of MR models took time no more than 2.3 minutes, while the frequentist and Bayesian approaches required 1.3 minutes and > 1h, respectively, under the same conditions. This suggests that the h-likelihood approach is computationally more efficient than the Bayesian approach but less efficient than frequentist approach. However, this comparison cannot be interpreted strictly considering that program MARK is written in compiled Fortran code, which will accelerate the computation speed, while the h-likelihood was implemented in R only. The h-likelihood might deliver the same computation efficiency as much as the frequentist approach does if it were rewritten in compiled code. I expect that when the sample sizes are larger, the computation time for program MARK would increase exponentially as the dimension of integral required to compute the likelihood increases with the number of individuals captured. In comparison, my experiences from the simulation studies I performed for all three projects suggest that the runtime of the h-likelihood increases only slightly when the sample size increases. As a result, in practice, I recommend to apply the frequentist approach when a sample size is small and the h-likelihood otherwise to avoid computational issues that might arise from the application of the frequentist approach.

In all the three projects, the MR models assume a closed population, which is often suitable for analyzing data collected over a short study period but may be unrealistic when the time length between sampling occasions is fairly long or the study is conducted over a long period of time. It is possible that individuals migrate throughout a study area, but there could also be birth or death of individuals during a MR experiment. The standard model for open populations is the Jolly–Seber model for analyzing MR data from open populations (see Chapter 5 summary in Seber, 1982, for an overview), which accounts for the survival of individuals over time. This model has been extended in many ways to account for individual differences in both survival and capture probability, as is akin to Pledger et al. (2003) who accounts for both individual covariates and random effects for modelling individual heterogeneity. The use of the h-likelihood is possible for such models, as long as the models depend on any random effects.

When MR models are extended to account for random effects, following distributions another than normal, one challenge would be in identifying the canonical scales required in defining the h-likelihood, as described in Chapter 1. Lee and Nelder (1996, 2001) and Lee et al. (2017) have identified the canonical scales for common distributions of random effects, such as normal, gamma, inverse-gamma and beta distributions; however, the canonical scales for other distributions are still unknown but should be ascertained to avoid any nonsensical estimates from the improperly defined h-likelihood. Lee et al. (2017) have developed the method to find out the canonical scales through a reparameterization of a model into a specific form, which mimics the model with the response variables following distributions in the exponential family. Yet, some distributions assumed on the random effects (e.g., uniform distribution) do not allow the model to be in the family of models that can be written in this specific form, and so the use of the h-likelihood may not be currently available for such cases.

Another problem to consider in the future is that the capture probability may be described by flexible functions of multiple covariates. If the capture probability is assumed to be a linear function of the covariates on some scale (e.g., the logit scale) as in Chapters 2 and 3, then it is simple to incorporate the interaction terms. However, this is not straightforward when the models are combined with a GAM structure as in Chapter 4. In this case, it may be possible to apply the method of thin-plate splines as shown by Wood (2003) to model the capture probability as a flexible and non-additive function of two or more covariates simultaneously. Even in higher dimensions, this method can provide parametric forms of the MR models; so, again, its conditional likelihood with the penalized term can be regarded as the h-likelihood for a GLMM, which can be directly fitted by the algorithm of Lee and Nelder (2001).

There are other diverse types of MR models wherein the h-likelihood can be used to analyze MR data. Examples include the MR models that account for variation due to measurement errors, spatial variation and missing covariates. The three projects in this thesis make the first step toward a novel approach for analyzing MR data, which may come up with solutions for statistical issues that have been encountered in the frequentist and Bayesian approaches. In the end, the three projects may be a significant key to obtaining information from a population more accurately than before, particularly for endangered animals worldwide, and thus lead to more proper management and plans for conserving such animals.

Bibliography

Bonner, S. and Schofield, M. (2014). Mc (mc) mc: exploring m onte c arlo integration within mcmc for mark–recapture models with individual covariates. *Methods in Ecology and Evolution*, 5(12):1305–1315.

Breiman, L. and Friedman, J. H. (1985). Estimating optimal transformations for multiple regression and correlation. *Journal of the American statistical Association*, 80(391):580–598.

Burnham, K. P., Anderson, D. R., White, G. C., Brownie, C., and Pollock, K. H. (1987). *Design and analysis methods for fish survival experiments based on release-recapture*, volume 5. American Fisheries Society Monograph, Maryland, U.S.A.

Burnham, K. P. and Overton, W. S. (1979). Robust estimation of population size when capture probabilities vary among animals. *Ecology*, 60(5):927–936.

Chao, A. (1987). Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, 43(4):783–791.

Chao, A., Lee, S., and Jeng, S. (1992). Estimating population size for capture-recapture data when capture probabilities vary by time and individual animal. *Biometrics*, 48(1):201–216.

Clutton-Brock, T. H. (1988). *Reproductive success: studies of individual variation in contrasting breeding systems*. University of Chicago Press, Chicago.

Coull, B. A. and Agresti, A. (1999). The use of mixed logit models to reflect heterogeneity in capture-recapture studies. *Biometrics*, 55(1):294–301.

de Boor, C. (1971). Subroutine package for calculating with b-splines. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Dulvy, N. K., Fowler, S. L., Musick, J. A., Cavanagh, R. D., Kyne, P. M., Harrison, L. R., Carlson, J. K., Davidson, L. N. K., Fordham, S. V., Francis, M. P., et al. (2014). Extinction risk and conservation of the world's sharks and rays. *elife*, 3:e00590.

Durban, J. W. and Elston, D. A. (2005). Mark-recapture with occasion and individual effects: abundance estimation through Bayesian model selection in a fixed dimensional parameter space. *Journal of Agricultural, Biological, and Environmental Statistics*, 10(3):291–305.

Eilers, P. H. and Marx, B. D. (1996). Flexible smoothing with b-splines and penalties. *Statistical science*, 11(2):89–121.

Gimenez, O., Cam, E., and Gaillard, J.-M. (2018). Individual heterogeneity and capture–recapture models: what, why and how? *Oikos*, 127(5):664–686.

Gimenez, O. and Choquet, R. (2010). Individual heterogeneity in studies on marked animals using numerical integration: capture–recapture mixed models. *Ecology*, 91(4):951–957.

Heinze, D., Broome, L., and Mansergh, I. (2004). A review of ecology and conservation of the mountain pygmy-possum Burramys parvus. *the Biology of Australian Possums and Gliders (eds R. Goldingay and S. Jackson)*, pages 254–267.

Herliansyah, R., King, R., and King, S. (2022). Laplace approximations for capture–recapture models in the presence of individual heterogeneity. *Journal of Agricultural, Biological and Environmental Statistics*, pages 1–18.

Hoffmann, M., Belant, J. L., Chanson, J. S., Cox, N. A., Lamoreux, J., Rodrigues, A. S. L., Schipper, J., and Stuart, S. N. (2011). The changing fates of the world's mammals. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1578):2598–2610.

Holzmann, H., Munk, A., and Zucchini, W. (2006). On identifiability in capture–recapture models. *Biometrics*, 62(3):934–939.

Horvitz, D. G. and Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685.

Huggins, R. and Hwang, W.-H. (2007). Non-parametric estimation of population size from capture–recapture data when the capture probability depends on a covariate. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 56(4):429–443.

Huggins, R. M. (1989). On the statistical analysis of capture experiments. *Biometrika*, 76(1):133–140.

Hurvich, C. M. and Tsai, C.-L. (1989). Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307.

Hwang, W.-H. and Huggins, R. (2005). An examination of the effect of heterogeneity on the estimation of population size using capture-recapture data. *Biometrika*, 92(1):229–233.

Jenni, L. and Kery, M. (2003). Timing of autumn bird migration under climate change: advances in long–distance migrants, delays in short–distance migrants. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 270(1523):1467–1471.

Kalbfleisch, J. D. and Sprott, D. A. (1973). Marginal and conditional likelihoods. *Sankhyā: The Indian Journal of Statistics, Series A*, 35(3):311–328.

Karp, S., Mendenhall, D., Sandí, F., Chaumont, N., Ehrlich, R., Hadly, A., and Daily, C. (2013). Forest bolsters bird abundance, pest control and coffee yield. *Ecology letters*, 16(11):1339–1347.

King, R. and Brooks, S. P. (2008). On the Bayesian estimation of a closed population size in the presence of heterogeneity and model uncertainty. *Biometrics*, 64(3):816–824.

King, R., McClintock, B. T., Kidney, D., and Borchers, D. (2016a). Capture–recapture abundance estimation using a semi-complete data likelihood approach. *The Annals of Applied Statistics*, 10(1):264–285.

King, R., McClintock, B. T., Kidney, D., and Borchers, D. (2016b). Capture–recapture abundance estimation using a semi-complete data likelihood approach. *The Annals of Applied Statistics*, 10(1):264–285.

King, R., Morgan, B. J. T., Gimenez, O., and Brooks, S. P. (2009). *Bayesian analysis for population ecology*. Chapman and Hall, Boca Raton, FL.

Langham, G. M., Schuetz, J. G., Distler, T., Soykan, C. U., and Wilsey, C. (2015). Conservation status of north american birds in the face of future climate change. *PloS one*, 10(9):e0135350.

Lee, Y. (2001). Can we recover information from concordant pairs in binary matched pairs? *Journal of Applied Statistics*, 28(2):239–246.

Lee, Y. and Nelder, J. A. (1996). Hierarchical generalized linear models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(4):619–678.

Lee, Y. and Nelder, J. A. (2001). Hierarchical generalised linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, 88(4):987–1006.

Lee, Y., Nelder, J. A., and Pawitan, Y. (2017). *Generalized linear models with random effects: unified analysis via H-likelihood*, volume 153. Chapman and Hall, Boca Raton, FL, 2 edition.

Link, W. A. (2003). Nonidentifiability of population size from capture-recapture data with heterogeneous detection probabilities. *Biometrics*, 59(4):1123–1130.

Mathai, A. M. and Provost, S. B. (1992). *Quadratic forms in random variables: theory and applications*. Marcel Dekker, Inc, New York, New York.

McCullagh, P. and Nelder, J. A. (2019). *Generalized linear models*, volume 37. Chapman and Hall, New York, 2 edition.

Nelder, J. A. and Wedderburn, R. W. M. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384.

Newton, I. (1989). *Lifetime reproduction in birds*. Academic Press, London.

Noh, M. and Lee, Y. (2007). REML estimation for binary data in GLMMs. *Journal of Multivariate Analysis*, 98(5):896–915.

Otis, D. L., Burnham, K. P., White, G. C., and Anderson, D. R. (1978). Statistical inference from capture data on closed animal populations. *Wildlife Monographs*, (62):3–135.

Patil, G. P. (1962). Maximum likelihood estimation for generalized power series distributions and its application to a truncated binomial distribution. *Biometrika*, 49(1/2):227–237.

Pilliod, D. S., Muths, E., Scherer, R. D., Bartelt, P. E., Corn, P. S., Hossack, B. R., Lambert, B. A., Mccaffery, R., and Gaughan, C. (2010). Effects of amphibian chytrid fungus on individual survival probability in wild boreal toads. *Conservation Biology*, 24(5):1259–1267.

Pledger, S. and Efford, M. (1998). Correction of bias due to heterogeneous capture probability in capture-recapture studies of open populations. *Biometrics*, 54(3):888–898.

Pledger, S., Pollock, K. H., and Norris, J. L. (2003). Open capture-recapture models with heterogeneity: I. Cormack-Jolly-Seber model. *Biometrics*, 59(4):786–794.

Plummer, M. (2003). Jags: A program for analysis of Bayesian graphical models using Gibbs sampling. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*, Vienna, Austria.

Pollock, K. H. (1982). A capture-recapture design robust to unequal probability of capture. *The Journal of Wildlife Management*, 46(3):752–757.

R Core Team (2020). *R: a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria.

Royle, J. A., Dorazio, R. M., and Link, W. A. (2007). Analysis of multinomial models with unknown index using data augmentation. *Journal of Computational and Graphical Statistics*, 16(1):67–85.

Seber, G. A. F. (1982). The estimation of animal abundance and related parameters.

Stoklosa, J. and Huggins, R. M. (2012). A robust p-spline approach to closed population capture–recapture models with time dependence and heterogeneity. *Computational Statistics & Data Analysis*, 56(2):408–417.

Stoklosa, J., Hwang, W.-H., Wu, S.-H., and Huggins, R. (2011). Heterogeneous capture–recapture models with covariates: a partial likelihood approach for closed populations. *Biometrics*, 67(4):1659–1665.

Trull, N., Böhm, M., and Carr, J. (2018). Patterns and biases of climate change threats in the IUCN Red List. *Conservation Biology*, 32(1):135–147.

Van Deusen, P. C. (2002). An em algorithm for capture-recapture estimation. *Environmental and Ecological Statistics*, 9(2):151–165.

Vehtari, A., Gelman, A., and Gabry, J. (2017). Practical Bayesian model evaluation using leave-one-out cross-validation and waic. *Statistics and computing*, 27:1413–1432.

Wegge, P., Pokheral, C. P., and Jnawali, S. R. (2004). Effects of trapping effort and trap shyness on estimates of tiger abundance from camera trap studies. *Animal Conservation*, 7(3):251–256.

White, G. C. and Burnham, K. P. (1999). Program MARK: survival estimation from populations of marked animals. *Bird Study*, 46(sup1):S120–S139.

White, G. C. and Cooch, E. G. (2017). Population abundance estimation with heterogeneous encounter probabilities using numerical integration. *The Journal of Wildlife Management*, 81(2):322–336.

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(1):95–114.

Yee, T. W., Stoklosa, J., and Huggins, R. M. (2015). The VGAM package for capture–recapture data using the conditional likelihood. *Journal of Statistical Software*, 65(5):1–33.

Yun, S. and Lee, Y. (2004). Comparison of hierarchical and marginal likelihood estimators for binary outcomes. *Computational Statistics & Data Analysis*, 45(3):639–650.

Appendix A

R Code for analyzing Snowshoe Hare Data via the H-likelihood in Chapter 2

```r
############ REQUIRED LIBRARY ############
library(Matrix)
library(VGAM)
library(wisp)
##########################################

######### PART 1. FUNCTIONS FOR FITTING ALGORITMS #########

#### function for finding initival values ####
init_val <- function(df, T) {

  # binomial responses
  binom_y <- apply(df, 1, sum)
  df <- cbind(binom_y, T - binom_y)

  # GLM fit without random effects
  fit <- vglm(as.matrix(df) ~ 1, posbinomial(omit.constant = FALSE), epsil = 1e-10,
              maxit = 150)

  # extract coefficients
  result <- as.numeric(coef(fit))

  return(result)
}

#### function for computing design matrix and adjusted response variables ####
aug_data <- function(n, y, design_mat_obs) {
  aug_y <- c(y, rep(0, n)) # y_a
  aug_design_mat <- rbind(design_mat_obs,
                  cbind(Matrix(matrix(0, nrow = n, ncol = ncol(design_mat_obs) - n)),
                               diag(n))) # T

  return(list(aug_y = aug_y, aug_design_mat = aug_design_mat))
}
```

```
#### function for running IRLS in STEP 1 ####
VIRLS_adj <- function(aug_y, aug_design_mat,
                      n, T,
                      cur_delta, cur_lambda,
                      max_iter, tol, stepsize) {

  pre_error = 1e10 # initial error value (must be large to run the next iteration)
  ori_delta = cur_delta # save the initial value of delta
  for (iter in seq_len(max_iter)) {
    print(iter)
    if (iter != 1) {
      error = sum((cur_eta - new_eta)^2) # error calculation
      print(error)
      print(tol * sum(new_eta^2))
      if ((error < tol * sum(new_eta^2)) | (iter == max_iter)) { # if error is small enough

        h <- diag(I_v_inv) / cur_lambda
        h[h >= 1] = 0.9999 # prevent that h is over 1 (possibly by small numerical error)

        v_gamma <- as.vector(I_v_inv %*% (as.vector(tail(cur_delta, n))) / cur_lambda)

        adjust_term = (diag(aug_design_mat[1:n, 2:ncol(aug_design_mat)] %*% I_v_inv %*%
            t(aug_design_mat[1:n, 2:ncol(aug_design_mat)]))) * diagW_v * v_gamma

        q = h - adjust_term
        q[q >= 1] = 0.9999 # prevent that q is over 1 (possibly by small numerical error)

        # d calculation
        d <- as.vector(tail(cur_delta, n))

        break

      } else {
        # if divergence happens in delta estimation, run VIRLS again
        # with smaller step size of Newton's method
        if (pre_error < error) {
          stepsize = stepsize/2
          cur_delta = ori_delta
        } else {
          cur_delta = new_delta
          pre_error = error
        }
      }
    }

    # eta computation
    cur_delta <- matrix(cur_delta, ncol = 1)
    cur_eta <- as.vector(aug_design_mat %*% cur_delta)

    # probability computation
    cur_p <- exp(cur_eta[1:(length(cur_eta) - n)]) /
      (1 + exp(cur_eta[1:(length(cur_eta) - n)]))
    cur_pi <- 1 - sapply(1 - cur_p, function(x) x^T)
```

```
# mean computation
cur_mu_posbin <- T * (cur_p / cur_pi)
cur_mu <- c(cur_mu_posbin, tail(as.vector(cur_eta), n))

# second derivate computation
sigma_inv_diag_vals <- (T * (cur_p - cur_p^2) * (cur_pi^(-1))) -
  ((T * cur_p * (cur_pi^(-2))) * T * cur_p * (1 - cur_pi))
sigma_inv_diag_vals <- c(sigma_inv_diag_vals, rep((1 / cur_lambda), n))

############### adjustment step ###############
# fisher info for random effects
I_v <- t(aug_design_mat[,2:ncol(aug_design_mat)]) %*% diag(sigma_inv_diag_vals) %*%
  (aug_design_mat[,2:ncol(aug_design_mat)])

I_v_inv <- solve(I_v)

# capture probs derivative about eta
p_eta <- cur_p - cur_p^2

# apture probs derivative about random effects
p_v <- p_eta

# pi (prob. of capturing at least once) derivative about eta
pi_eta <- T * cur_p *(1 - cur_pi)

# pi (prob. of capturing at least once) derivative about random effects
pi_v <- pi_eta

# diag(W) derivative about eta
diagW_eta <- (T * p_eta * (cur_pi^(-1))) - (T * cur_p * (cur_pi^(-2)) * pi_eta) -
  2 * (T * cur_p * p_eta * (cur_pi^(-1))) +
  (T * (cur_p^2) * (cur_pi^(-2)) * pi_eta) -
  2 * ((T^2) * cur_p * p_eta * (cur_pi^(-2))) +
  2 * ((T^2) * (cur_p^2) * (cur_pi^(-3)) * pi_eta) +
  2 * ((T^2) * cur_p * p_eta * (cur_pi^(-1))) -
  ((T^2) * (cur_p^2) * (cur_pi^(-2)) * pi_eta)

# diag(W) derivative about random effects
diagW_v <- diagW_eta

# S computation
term1 <- diag(I_v_inv) * diagW_eta
term2 <- diag(I_v_inv) * diagW_v * diag(I_v_inv) * (-1 * sigma_inv_diag_vals[1:n])

S <- (term1 + term2)/2

###############################################

# score function INCLUDING the step of updating random effects through h-likelihood
temp = aug_y - cur_mu - c(S, -1 * cur_lambda * S)
temp[(length(temp) - n + 1):(length(temp))] = tail(temp, n)/cur_lambda
s = t(aug_design_mat) %*% temp

# fisher info
I <- t(aug_design_mat) %*% diag(sigma_inv_diag_vals) %*% aug_design_mat
```

```
    I_inv <- solve(I)

    # new delta
    new_delta <- cur_delta + (stepsize * I_inv %*% s)

    # new and old eta values
    cur_eta = aug_design_mat %*% cur_delta
    new_eta = aug_design_mat %*% new_delta

    if (sum(is.na(new_eta)) > 0) {
      stop(”Diverged.”)
    }

  } # end of for (iter in seq_len(max_iter))

  return(list(new_delta = cur_delta, new_eta = cur_eta,
              pi = cur_pi, p = cur_p,
              I = I, I_inv = I_inv,
              h = h, q = q, d = d, stepsize = stepsize))
}

#### function for fitting gamma GLM in STEP 2 ####
vhglm <- function(aug_y, aug_design_mat,
                  n, T,
                  cur_delta, cur_lambda,
                  max_iter1, max_iter2, tol1, tol2) {
  cur_eta = 0 # initial value for linear predictor (set as 0 to compute error)
  pre_error = 1e10 # initial error value (must be large to run the next iteration)
  stepsize = 1.0
  for (iter in seq_len(max_iter1)) {

    # STEP 1: estimate parameters and random-effects
    new_virls_result <- try(VIRLS_adj(aug_y, aug_design_mat,
                                      n, T,
                                      cur_delta, cur_lambda,
                                      max_iter = max_iter2, tol = tol2, stepsize), TRUE)

    if (class(new_virls_result) == ”try-error”) {
      stop(”Diverged␣in␣VIRLS.”)
    }

    # STEP 2: estimate lambda (variance of random-effects)
    y_d = ((new_virls_result$d)^2) / (1 - new_virls_result$q)
    fit <- tryCatch(glm(y_d ~ 1, family = Gamma(link = ”log”),
                        weights = (1 - new_virls_result$q),
                        maxit = 150),
                    warning = function(w) w)

    if (inherits(fit,”warning”)) {
      stop(”Diverged␣in␣VIRLS␣−−␣diverged␣sigma_v”)
    }

    new_lambda <- exp(coef(fit)) # updated lambda
    print(sqrt(new_lambda))
    # error calculation
    error = sum((cur_eta - new_virls_result$new_eta)^2)
    print(error)
    print((tol1) * sum((new_virls_result$new_eta)^2))

    if (error < (tol1 * sum((new_virls_result$new_eta)^2))) {
      cur_delta = new_virls_result$new_delta
      break
    }

    cur_eta = new_virls_result$new_eta
    cur_delta = new_virls_result$new_delta
    stepsize = new_virls_result$stepsize
    cur_lambda = new_lambda
    pre_error = error

    print(cur_delta[1,])
  }

  return(list(new_eta = new_virls_result$new_eta,
              new_delta = new_virls_result$new_delta, new_lambda = cur_lambda,
              var_info = diag(as.matrix(new_virls_result$I_inv)),
              pi = new_virls_result$pi,
              p =  new_virls_result$p,
              I = new_virls_result$I,
              I_inv = new_virls_result$I_inv,
              log_lambda_std = summary(fit)$coefficients[2]))
}

#### function for estimating population size ####
est_N <- function(pi, p,
                  I, I_inv,
                  T, n, new_delta) {

  # actual var(delta)
  actual_I_inv <- I_inv
  # var(v)
  random_I_inv <- bdiag(0, solve(I[2:nrow(I), 2:nrow(I)]))

  # variance of hat N
  term1 <- sum((pi^(-2))*(1 - pi)) # variance of HT-estimator itself

  # first derivative about intercept
  d <- c(-sum((pi^(-2))*(1 - pi) * T * p))

  # first derivative about random effects
  for (i in 1:n){
    d <- c(d, -((pi[i]^(-2))*(1 - pi[i]) *  T * p[i]))
  }

  term2 <- as.vector(t(d) %*% actual_I_inv %*% d)
  term3 <- 2 * as.vector(t(d) %*% random_I_inv %*% d)

  # second derivative about intercept and intercept
  deriv_pi_intc <-(1 - pi) * T * p
  deriv_intc_intc <- sum(((2/pi^3) * deriv_pi_intc -
                           (1/(pi^2)) * deriv_pi_intc) * T * p +
                          ((1/pi) - (pi^(-2))) * T * (p - p^2))

  # second derivative about intercept and random effects
  deriv_intc_r <- ((2/(pi^3)) * deriv_pi_intc - (1/(pi^2)) * deriv_pi_intc) * T * p +
    ((1/pi) - (pi^(-2))) * T * (p - p^2)
  deriv_intc_r <- as.matrix(deriv_intc_r, ncol = 1)

  # second derivative about random effects and random effects
  deriv_pi_v <- (1 - pi) * T * p
  deriv_r_r <- matrix(0, nrow = n, ncol = n)
  for (i in 1:n) {
    deriv_r_r[i,i] <- ((2/(pi[i]^3)) * deriv_pi_v[i] -
                        (1/(pi[i]^2)) * deriv_pi_v[i]) * T * p[i] +
      ((-(1 - pi[i])/(pi[i]^2)) * T * (p[i] - p[i]^2))
  }

  # second derivative matrix
  second_d <- rbind(cbind(deriv_intc_intc, t(deriv_intc_r)),
                    cbind(deriv_intc_r, deriv_r_r))

  term4 <- sum(diag(second_d %*% actual_I_inv %*% second_d %*% random_I_inv))
  term5 <- 2 * sum(diag(second_d %*% random_I_inv %*% second_d %*% random_I_inv))

  term6 <- 6 * as.vector(t(new_delta) %*% second_d %*% random_I_inv %*%
                          second_d %*% new_delta)

  var_N <- term1 + term2 + term3 + term4 + term5 + term6

  # population size estimator
  adj <- sum(diag(second_d %*% random_I_inv)) # term for bias correction in N estimation
  N_est <- sum(1/pi) + adj

  # log-normal confidence interval for N
  c = exp(1.96 * sqrt(log(1 + (sqrt(var_N) / (N_est - n))^2)))
  N_upper = n + c * (N_est - n)
  N_lower = n + (N_est - n) / c

  # Wald confidence interval for N
  N_upper2 = N_est + 1.96*sqrt(var_N)
  N_lower2 = N_est - 1.96*sqrt(var_N)

  return(list(N_est = N_est, N_lower = N_lower, N_upper = N_upper,
              N_lower2 = N_lower2, N_upper2 = N_upper2, var_N = var_N))
}

############################################################

######### PART 2. SNOWSHOE HARE DATA ANALYSIS #########
# call the data set
data(hare.samp.cr)

# basic settings
snow <- hare.samp.cr$capture # observed capture history
```

```
n = nrow(snow)
T = ncol(snow)

# response variables
y <- apply(snow, 1, sum) # binomial data

# observed design matrix
Z <- diag(n) # for random-effects
design_mat_obs <- cbind(1, Z) # for fixed and random effects

# find initial values
options(warn = -1)
cur_delta <- c(init_val(snow, T), rnorm(n, 0 ,0))

# augmented design matrix and response variables
data_aug <- aug_data(n, y, design_mat_obs)

# run fitting algorithm
max_iter1 = 150
max_iter2 = 150
tol1 = 1e-13
tol2 = 1e-13
lambda_trial = c(1e-3)
fit <- vhglm(data_aug$aug_y, data_aug$aug_design_mat,
             n, T,
             cur_delta, lambda_trial,
             max_iter1, max_iter2, tol1, tol2)

# estimate the population size
N_est <- est_N(fit$pi, fit$p,
                 fit$I, fit$I_inv, T, n, fit$new_delta)
print(N_est)

#cAIC computation
cAIC_h <- 2*(1 + 1 + n) -2 * (sum((as.vector(fit$new_eta[1:n,]) * y) -
                                  log(fit$pi) + T * log(1 - fit$p)) +
                              sum(pnorm(as.vector(tail(fit$new_delta, n)),
                                        mean = 0, sd = sqrt(fit$new_lambda))))
print(cAIC_h)
######################################################
```

Appendix B

R Code for analyzing Snowshoe Hare Data via the H-likelihood in Chapter 3

```
############ REQUIRED LIBRARY ############
library(Matrix)
library(VGAM)
library(wisp)
##########################################

######### PART 1. FUNCTIONS FOR FITTING ALGORITMS #########

#### function for finding initival values ####
init_val <- function(df, type) {

  # GLM fit without random effects
  if (type == "M.h") {
    fit <- vglm(as.matrix(df) ~ 1, posbernoulli.t(parallel.t = TRUE ~ 1),
                epsil = 1e-10, maxit = 150)
  } else if (type == "M.bh") {
    fit <- vglm(as.matrix(df) ~ 1, posbernoulli.b, epsil = 1e-10, maxit = 150)
  } else if (type == "M.th") {
    fit <- vglm(as.matrix(df) ~ 1, posbernoulli.t, epsil = 1e-10, maxit = 150)
  } else {
    fit <- vglm(as.matrix(df) ~ 1, posbernoulli.tb, epsil = 1e-10, maxit = 150)
  }

  # extract coefficients
  result <- as.numeric(coef(fit))
  if (type == "M.b" | type == "M.tbh") {
    result <- c(result[2:length(result)], result[1])
  }

  return(result)
}

#### function for computing design matrix and adjusted response variables ####
aug_data <- function(n, y, design_mat_obs, y_list, design_mat_obs_list) {
  aug_y <- c(y, rep(0, n)) # y_a
  aug_design_mat <- rbind(design_mat_obs, cbind(Matrix(matrix(0, nrow = n,
                                                       ncol = ncol(design_mat_obs) - n)),
                                                         diag(n))) # T

  aug_y_list <- y_list # response variable (in vector form) for each i
  aug_y_list[[n + 1]] <- rep(0, n) # and the augmented part of the response variable
  aug_design_mat_list <- design_mat_obs_list # design matrix for each i
  aug_design_mat_list[[n + 1]] <- cbind(Matrix(matrix(0, nrow = n,
                                                         ncol = ncol(design_mat_obs) - n)),
                                      diag(n)) # and the aumented part of the design matrix

  return(list(aug_y = aug_y, aug_design_mat = aug_design_mat,
              aug_y_list = aug_y_list, aug_design_mat_list = aug_design_mat_list))
}

#### function for running IRLS in STEP 1 ####
VIRLS_adj <- function(aug_y, aug_design_mat, aug_design_mat_list,
                      n, T, z_mat = NULL,
                      cur_delta, cur_lambda,
                      max_iter, tol, type, stepsize) {

  if ((type != "M.tbh") & (type != "M.bh") & (!is.null(z_mat))) {
    stop("have␣any␣Z␣information␣which␣must␣not␣exist")
  }

  # random effects start column index in the design matrix
  if (type == "M.h") {
    rand_start_idx <- 2
  }
  if (type == "M.th") {
    rand_start_idx <- T + 1
  }
  if (type == "M.bh") {
    rand_start_idx <- 3
  }
  if (type == "M.tbh") {
    rand_start_idx <- T + 2
  }

  pre_error = 1e10 # initial error value (must be large to run the next iteration)
  ori_delta = cur_delta # save the initial value of delta
  for (iter in seq_len(max_iter)) {
    print(iter)
    if (iter != 1) {
      error = sum((cur_eta - new_eta)^2) # error calculation
      print(error)
      print(tol * sum(new_eta^2))
      if ((error < tol * sum(new_eta^2)) | (iter == max_iter)) {

        h <- diag(I_v_inv) / cur_lambda
        h[h >= 1] = 0.9999 # prevent that h is over 1 (due to small numerical error)

        v_gamma <- as.vector(I_v_inv %*% (as.vector(tail(cur_delta, n))) / cur_lambda)
    adjust_term <- trace_v * v_gamma # derivative for vhat in adjusted term

    q = h - adjust_term
    q[q >= 1] = 0.9999

    # d calculation
    d <- as.vector(tail(cur_delta, n))

    break

  } else {
    # if divergence happens in delta estimation,
    #run VIRLS again with smaller step size of Newton's method
    if ((pre_error < error)) {
      stepsize = stepsize/2
      cur_delta = ori_delta
    } else {
      cur_delta = new_delta
      pre_error = error
    }
  }
}

# eta computation
cur_delta <- matrix(cur_delta, ncol = 1)
cur_eta <- as.vector(aug_design_mat %*% cur_delta)

# probability computation
cur_p <- exp(cur_eta[1:(length(cur_eta) - n)]) /
  (1 + exp(cur_eta[1:(length(cur_eta) - n)]))
cur_p_mat <- matrix(cur_p, nrow = n, byrow = TRUE)
cur_pi <- 1 - apply(1 - cur_p_mat[,1:T], 1, prod) # prob. captured at least once

# mean computation
if (type == "M.tbh" | type == "M.bh") {
  cur_minus_pi_pi <- (1 - cur_pi) / cur_pi
  cur_minus_pi_pi_mat <- matrix(rep(cur_minus_pi_pi, T), nrow = n)
  cur_minus_pi_pi_mat <- Matrix(cbind(cur_minus_pi_pi_mat,
                                     matrix(0, nrow = n, ncol = T - 1)))
  cur_minus_pi_pi <- as.vector(t(cur_minus_pi_pi_mat))

  cur_mu_posbin <- as.vector(t(cbind((1 - z_mat), z_mat[,2:T]))) *
    cur_p + cur_p * cur_minus_pi_pi
}
if (type == "M.th" | type == "M.h") {
  cur_mu_posbin <- cur_p / rep(cur_pi, each = T)
  cur_mu_posbin_mat <- matrix(cur_mu_posbin, nrow = n, byrow = TRUE)
}

cur_mu <- c(cur_mu_posbin, tail(as.vector(cur_eta), n))

# second derivative computation
sigma_inv_split <- list()
if (type == "M.tbh" | type == "M.bh") {
  for (i in 1:n) {
    cur_pc <- cur_p_mat[i, 1:T]
    cur_pr <- cur_p_mat[i, (T + 1):(2*T - 1)]

    deric_c <- apply(as.matrix(cur_pc / cur_pi[i], ncol = 1), 1,
                     function(x) x * (cur_pc - cur_pc / cur_pi[i]))
    diag_elements <- (1 - cur_pc / cur_pi[i]) * (cur_pc / cur_pi[i])
    diag(deric_c) <- diag_elements
    deric_c <- deric_c + diag(((1 - z_mat)[i,] * (cur_pc - cur_pc^2)) -
                                (cur_pc - cur_pc^2))
    deric_r <- diag(z_mat[i, 2:T] * (cur_pr - cur_pr^2))

    sigma_inv_split[[i]] <- bdiag(deric_c, deric_r)
  }
}
if (type == ”M.th” | type == ”M.h”) {
  for (i in 1:n) {
    cur_deriv_mu_eta <- t((cur_p_mat[i,] - cur_mu_posbin_mat[i,]) %*%
                           t(cur_mu_posbin_mat[i,]))
    new_diag_elements <- cur_mu_posbin_mat[i,] * (1 - cur_mu_posbin_mat[i,])
    diag(cur_deriv_mu_eta) <- new_diag_elements
    sigma_inv_split[[i]] <- cur_deriv_mu_eta
  }
}
sigma_inv_split[[n + 1]] <- (1 / cur_lambda) * diag(n)

############### adjustment step ###############
# fisher info for random effects
I_v <- 0
for (ii in seq_len(n + 1)) {
  cur_fisher <- t(aug_design_mat_list[[ii]][,rand_start_idx:ncol(aug_design_mat)]) %*%
    sigma_inv_split[[ii]] %*%
    (aug_design_mat_list[[ii]][,rand_start_idx:ncol(aug_design_mat)])
  I_v <- I_v + cur_fisher
}

I_v_inv <- solve(I_v)

# initial capture probs derivative about eta
pc_eta <- cur_p_mat[,1:T] - (cur_p_mat[,1:T])^2
if (type == ”M.tbh” | type == ”M.bh”) {
  pr_eta <- cur_p_mat[,(T + 1):(2*T - 1)] - (cur_p_mat[,(T + 1):(2*T - 1)])^2
}

# initial capture probs derivative about random effects
pc_v <- cur_p_mat[,1:T] - (cur_p_mat[,1:T])^2
if (type == ”M.tbh” | type == ”M.bh”) {
  pr_v <- cur_p_mat[,(T + 1):(2*T - 1)] - (cur_p_mat[,(T + 1):(2*T - 1)])^2
}

# pi (prob. of capturing at least once) derivative about eta
pi_eta <- (1 - cur_pi) * cur_p_mat[,1:T]

# pi (prob. of capturing at least once) derivative about random effects
pi_v <- (1 - cur_pi) * apply(cur_p_mat[,1:T], 1, sum)
```

```
# diag(W) derivative about eta
diagW_eta_list <- list()
for (t in 1:T) {
  if (type == "M.tbh" | type == "M.bh") {
    null_mat  <- matrix(0, ncol = T, nrow = n)
    null_mat[,t] <- pc_eta[,t]
    tmp1_eta <- (null_mat / cur_pi) - (cur_p_mat[,1:T] * (pi_eta[,t] / cur_pi^2)) -
      (2 * cur_p_mat[,1:T] * null_mat / cur_pi^2) + (2 * ((cur_p_mat[,1:T])^2) *
                                                        (pi_eta[,t] / (cur_pi^3))) -
      (null_mat) + (2 * cur_p_mat[,1:T] * null_mat) + ((1 - z_mat) * null_mat) -
      (2 * (1 - z_mat) * cur_p_mat[,1:T] * null_mat)
  }
  if (type == "M.th" | type == "M.h") {
    null_mat  <- matrix(0, ncol = T, nrow = n)
    null_mat[,t] <- pc_eta[,t]
    tmp1_eta <- (null_mat / cur_pi) - (cur_p_mat[,1:T] * (pi_eta[,t] / cur_pi^2)) -
      (2 * cur_p_mat[,1:T] * null_mat / cur_pi^2) +
      (2 * ((cur_p_mat[,1:T])^2) *(pi_eta[,t] / (cur_pi^3)))
  }
  diagW_eta_list[[t]] <- tmp1_eta
}

# diag(W) derivative about random effects
if (type == "M.tbh" | type == "M.bh") {
  diagW_v <- (pc_v / cur_pi) - (cur_p_mat[,1:T] * (pi_v / cur_pi^2)) -
    (2 * cur_p_mat[,1:T] * pc_v / cur_pi^2) +
    (2 * ((cur_p_mat[,1:T])^2) *(pi_v / (cur_pi^3))) -
    (pc_v) + (2 * cur_p_mat[,1:T] * pc_v) + ((1 - z_mat) * pc_v) -
    (2 * (1 - z_mat) * cur_p_mat[,1:T] * pc_v)
}
if (type == "M.th" | type == "M.h") {
  diagW_v <- (pc_v / cur_pi) - (cur_p_mat[,1:T] * (pi_v / cur_pi^2)) -
    (2 * cur_p_mat[,1:T] * pc_v / cur_pi^2) +
    (2 * ((cur_p_mat[,1:T])^2) *(pi_v / (cur_pi^3)))
}

# off-diag and diag(W) derivative about eta
W_eta_list <- list()
for (t in 1:T) {
  W_eta <- matrix(NA, nrow = n, ncol = T*T)
  idx = 0
  for (k in 1:T) {
    for (l in 1:T) {
      if (l != k) {
        # nondiag(W) derivative about eta
        idx = idx + 1
        tmp2 <- ((k == t) * pc_eta[,k] * cur_p_mat[,l] + (l == t) * pc_eta[,l] *
                    cur_p_mat[,k]) * (1/cur_pi - (1/cur_pi^2)) +
          (cur_p_mat[,k] * cur_p_mat[,l]) *
          (-1 * pi_eta[,t] / (cur_pi^2) + 2 * pi_eta[,t] / (cur_pi^3))
        W_eta[,idx] <- tmp2
      }
      if (l == k) { # diag(W) derivative about eta
        idx = idx + 1
        W_eta[,idx] <-  diagW_eta_list[[t]][,l]
      }
    }
  }
  W_eta_list[[t]] <- W_eta
}

# W derivative about random effects
W_v <- matrix(NA, nrow = n, ncol = T*T)
idx = 0
for (k in 1:T) {
  for (l in 1:T) {
    if (l != k) {
      # nondiag(W) derivative about random effects
      idx = idx + 1
      tmp2 <- (pc_v[,k] * cur_p_mat[,l] + pc_v[,l] * cur_p_mat[,k]) *
        (1/cur_pi - (1/cur_pi^2)) +
        (cur_p_mat[,k] * cur_p_mat[,l]) * (-1 * pi_v / (cur_pi^2) + 2 *
                                             pi_v / (cur_pi^3))
      W_v[,idx] <- tmp2
    }
    if (l == k) { # diag(W) derivative about random effects
      idx = idx + 1
      W_v[,idx] <- diagW_v[,l]
    }
  }
}

# diag(U) derivative about eta
if (type == ”M.tbh” | type == ”M.bh”) {
  diagW_eta_list2 <- list()

  for (t in 1:(T - 1)) {
    null_mat  <- matrix(0, ncol = T - 1, nrow = n)
    null_mat[,t] <- pr_eta[,t]
    tmp3_eta <- (z_mat[,2:T] * null_mat) - (2 * z_mat[,2:T] *
                                              cur_p_mat[,T + t] * null_mat)
    diagW_eta_list2[[t]] <- tmp3_eta
  }
}

# diag(U) derivative about random effects
if (type == ”M.tbh” | type == ”M.bh”) {
  diagW_v2 <- (z_mat[,2:T] * pr_v) - (2 * z_mat[,2:T] *
                                        cur_p_mat[,(T + 1):(2*T - 1)] * pr_v)
}

# S computation
# **below W_eta, W_eta2, tmp_cap and tmp_recap
# are the tricks to compute
# trace[(Z' I')' ((Z' I') W* (Z' I')')^-1 (Z' I') \partial W* / \partial eta]
#faster**
W_eta <- as.vector(t(Reduce(cbind, W_eta_list)))
if (type == ”M.tbh” | type == ”M.bh”) {
  W_eta2 <- as.vector(t(Reduce(cbind, diagW_eta_list2)))
}
```

```
tmp_cap <- c()
if (type == ”M.tbh” | type == ”M.bh”) {
  tmp_recap <- c()
}

# trace[(Z' I')' ((Z' I') W* (Z' I')')^-1 (Z' I') \partial W* / \partial v]
trace_v <- c()
if (type == ”M.tbh” | type == ”M.bh”) {
  # partial^2 h / partial v partial eta
  h_deriv_v_eta <- matrix(0, nrow = n, ncol = 2*T - 1)
}
if (type == ”M.h” | type == ”M.th”) {
  h_deriv_v_eta <- matrix(0, nrow = n, ncol = T)
}
for (i in 1:n) {

  cur_tmp <- as.matrix(aug_design_mat_list[[i]][,rand_start_idx:
                                                   ncol(aug_design_mat)] %*%
          I_v_inv %*% t(aug_design_mat_list[[i]][,rand_start_idx:
                                                    ncol(aug_design_mat)]))

  tmp_cap <- c(tmp_cap, rep(as.vector(cur_tmp[1:T, 1:T]), T))
  if (type == ”M.tbh” | type == ”M.bh”) {
    tmp_recap <- c(tmp_recap, rep(diag(cur_tmp)[(T + 1):(2*T - 1)], T - 1))
  }

  if (type == ”M.tbh” | type == ”M.bh”) {
    cur_W_block <- bdiag(matrix(W_v[i,], nrow = T,byrow = TRUE),
                         diag(diagW_v2[i,]))
  }
  if (type == ”M.th” | type == ”M.h”) {
    cur_W_block <- matrix(W_v[i,], nrow = T, byrow = TRUE)
  }

  if (type == ”M.tbh” | type == ”M.bh”) {
    tmpp <- -1 * colSums(sigma_inv_split[[i]]) / (colSums(sigma_inv_split[[i]]) +
                                          (1/(cur_lambda * (2*T - 1))))^(-1)
  }
  if (type == ”M.h” | type == ”M.th”) {
    tmpp <- -1 * colSums(sigma_inv_split[[i]]) / (colSums(sigma_inv_split[[i]]) +
                                          (1/(cur_lambda * T)))^(-1)
  }
  h_deriv_v_eta[i,] <- tmpp
  trace_v <- c(trace_v, sum(diag(cur_tmp %*% cur_W_block)))
}

# see equation [?]: this is the first term of it
#(term1 and term1r are blocks of the final matrix)
term1 <- rowSums(matrix(W_eta * tmp_cap, ncol = T*T, byrow = TRUE))
term1 <- matrix(term1, ncol = T, byrow = TRUE)

if (type == ”M.tbh” | type == ”M.bh”) {
  term1r <- rowSums(matrix(tmp_recap * W_eta2, ncol = T - 1, byrow = TRUE))
  term1r <- matrix(term1r, ncol = T - 1, byrow = TRUE)
```

```
      # S values in matrix form
      S_mat <- (cbind(term1, term1r) + cbind(term1, term1r) * h_deriv_v_eta)/2
    }
    if (type == ”M.th” | type == ”M.h”) {
      # S values in matrix form
      S_mat <- (term1 + term1 * h_deriv_v_eta)/2
    }

    # S values in vector form
    S <- as.vector(t(S_mat))

    ################################################

    # score function INCLUDING the step of updating random effects through h-likelihood
    temp = aug_y - cur_mu - c(S, -1 * cur_lambda *
                                  as.vector(t(aug_design_mat[1:length(S),
                                    rand_start_idx:ncol(aug_design_mat)]) %*% S))

    temp[(length(temp) - n + 1):(length(temp))] = tail(temp, n)/cur_lambda
    s = t(aug_design_mat) %*% temp

    # fisher info
    I <- 0
    for (ii in seq_len(n + 1)) {
      cur_fisher <- t(aug_design_mat_list[[ii]]) %*%
        sigma_inv_split[[ii]] %*% aug_design_mat_list[[ii]]
      I <- I + cur_fisher
    }
    I_inv <- solve(I)

    # new delta
    new_delta <- cur_delta + (stepsize * I_inv %*% s)

    # new and old eta values
    cur_eta = aug_design_mat %*% cur_delta
    new_eta = aug_design_mat %*% new_delta

    if (sum(is.na(new_eta)) > 0) {
      stop(”Diverged.”)
    }

  } # end of for (iter in seq_len(max_iter))

  if (type == ”M.tbh” | type == ”M.bh”) {
    pr_mat = cur_p_mat[,(T + 1):(2*T - 1)]
  } else {
    pr_mat = NULL
  }

  return(list(new_delta = cur_delta, new_eta = cur_eta,
              pi = cur_pi, pc_mat = cur_p_mat[,1:T], pr_mat = pr_mat,
              I = I, I_inv = I_inv,
              h = h, q = q, d = d, stepsize = stepsize))
}
```

```
#### function for fitting gamma GLM in STEP 2 ####
vhglm <- function(aug_y, aug_design_mat, aug_design_mat_list,
                  n, T, z_mat,
                  cur_delta, cur_lambda,
                  max_iter1, max_iter2, tol1, tol2, type) {

  cur_eta = 0 # initial value for linear predictor (set as 0 to compute error)
  pre_error = 1e10 # initial error value (must be large to run the next iteration)
  stepsize = 1.0
  for (iter in seq_len(max_iter1)) {

    # STEP 1: estimate parameters and random-effects
    if (type == ”M.th” | type == ”M.h”) {
      new_virls_result <- try(VIRLS_adj(aug_y, aug_design_mat, aug_design_mat_list,
                                        n, T, NULL,
                                        cur_delta, cur_lambda,
                                        max_iter = max_iter2, tol = tol2, type,
                                        stepsize), TRUE)
    }
    if (type == ”M.tbh” | type == ”M.bh”) {
      new_virls_result <- try(VIRLS_adj(aug_y, aug_design_mat, aug_design_mat_list,
                                        n, T, z_mat,
                                        cur_delta, cur_lambda,
                                        max_iter = max_iter2, tol = tol2, type,
                                        stepsize), TRUE)
    }

    if (class(new_virls_result) == ”try-error”) {
      stop(”Diverged␣in␣VIRLS.”)
    }

    # STEP 2: estimate lambda (variance of random-effects)
    y_d = ((new_virls_result$d)^2) / (1 - new_virls_result$q)
    fit <- tryCatch(glm(y_d ~ 1, family = Gamma(link = ”log”),
                        weights = (1 - new_virls_result$q),
                        maxit = 150),
                    warning = function(w) w)

    if (inherits(fit,”warning”)) {
      stop(”Diverged␣in␣VIRLS␣--␣diverged␣sigma_v”)
    }

    new_lambda <- exp(coef(fit)) # updated lambda
    print(sqrt(new_lambda))

    # error calculation
    error = sum((cur_eta - new_virls_result$new_eta)^2)
    print(error)
    print((tol1) * sum((new_virls_result$new_eta)^2))

    if (error < (tol1 * sum((new_virls_result$new_eta)^2))) {
      cur_delta = new_virls_result$new_delta
      break
    }
    cur_eta = new_virls_result$new_eta
    cur_delta = new_virls_result$new_delta
    stepsize = new_virls_result$stepsize
    cur_lambda = new_lambda
    pre_error = error

    if (type == ”M.h”){
      print(cur_delta[(1),])
    }
    if (type == ”M.bh”){
      print(cur_delta[(1:2),])
    }
    if (type == ”M.th”) {
      print(cur_delta[(1:T),])
    }
    if (type == ”M.tbh”){
      print(cur_delta[(1:(T + 1)),])
    }
  }

  return(list(new_eta = new_virls_result$new_eta,
              new_delta = new_virls_result$new_delta, new_lambda = cur_lambda,
              var_info = diag(as.matrix(new_virls_result$I_inv)),
              pi = new_virls_result$pi,
              pc_mat =  new_virls_result$pc_mat,
              pr_mat =  new_virls_result$pr_mat,
              I = new_virls_result$I,
              I_inv = new_virls_result$I_inv,
              log_lambda_std = summary(fit)$coefficients[2]))
}

#### function for estimating population size ####
est_N <- function(pi, pc_mat, I, I_inv, T, n, type, new_delta) {

  # end index for fixed effects in design matrix (ignoring behavioural effect index)
  if (type == ”M.h” | type == ”M.bh”) {
    fixed_end_idx <- 1
  }
  if (type == ”M.th” | type == ”M.tbh”) {
    fixed_end_idx <- T
  }

  # var-cov
  if (type == ”M.h”) {
    # actual var(delta)
    actual_I_inv <- I_inv
    # var(v)
    random_I_inv <- bdiag(0, solve(I[(fixed_end_idx + 1):nrow(I),
                                     (fixed_end_idx + 1):nrow(I)]))
  }
  if (type == ”M.th”) {
    # actual var(delta)
    actual_I_inv <- I_inv
    # var(v)
    random_I_inv <- bdiag(diag(rep(0, fixed_end_idx)),
                         solve(I[(fixed_end_idx + 1):nrow(I),
                               (fixed_end_idx + 1):nrow(I)]))
}
if (type == ”M.bh”) {
  # delta estimate without behavioural effect
  new_delta <- new_delta[-(fixed_end_idx + 1)]
  # actual var(delta)
  actual_I_inv <- I_inv[-(fixed_end_idx + 1), -(fixed_end_idx + 1)]
  # var(v)
  random_I_inv <- bdiag(0, solve(I[(fixed_end_idx + 2):nrow(I),
                                    (fixed_end_idx + 2):nrow(I)]))
}
if (type == ”M.tbh”) {
  # delta estimate without behavioural effect
  new_delta <- new_delta[-(fixed_end_idx + 1)]
  # actual var(delta)
  actual_I_inv <- I_inv[-(fixed_end_idx + 1), -(fixed_end_idx + 1)]
  # var(v)
  random_I_inv <- bdiag(diag(rep(0, fixed_end_idx)),
                         solve(I[(fixed_end_idx + 2):nrow(I),
                                 (fixed_end_idx + 2):nrow(I)]))
}

# variance of hat N
term1 <- sum((pi^(-2))*(1 - pi)) # variance of HT-estimator itself

# first derivative about intercept
if (type == ”M.h” | type == ”M.bh”) {
  d <- c(-sum((pi^(-2))*(1 - pi) * apply(pc_mat, 1, sum)))
}

# first derivative about time effects
if (type == ”M.th” | type == ”M.tbh”) {
  d <- c()
  for (t in 1:T) {
    d <- c(d, -sum((pi^(-2))*(1 - pi) * pc_mat[,t]))
  }
}

# first derivative about random effects
for (i in 1:n){
  d <- c(d, -((pi[i]^(-2))*(1 - pi[i]) * sum(pc_mat[i,])))
}

term2 <- as.vector(t(d) %*% actual_I_inv %*% d)
term3 <- 2 * as.vector(t(d) %*% random_I_inv %*% d)

# second derivative about intercept and intercept
if (type == ”M.h” | type == ”M.bh”) {
  deriv_pi_intc <-(1 - pi) * apply(pc_mat, 1, sum)
  deriv_intc_intc <- sum(((2/pi^3) * deriv_pi_intc - (1/(pi^2)) * deriv_pi_intc) *
                            apply(pc_mat, 1, sum) +
                            ((1/pi) - (pi^(-2))) * apply(pc_mat - pc_mat^2, 1, sum))
}
```

```
# second derivative about time effects and time effects
if (type == "M.th" | type == "M.tbh") {
  # second derivative of N
  deriv_t_t <- matrix(NA, nrow = T, ncol = T)
  for (t1 in 1:T) {
    for (t2 in 1:T) {
      p_prod <- pc_mat[,t1] * pc_mat[,t2]
      pi_terms <- (1 - pi) * ((2/(pi^3)) - (1/(pi^2)))
      if (t1 != t2) {
        deriv_t_t[t1, t2] <- sum(p_prod * pi_terms)
      } else {
        deriv_t_t[t1, t2] <- sum(p_prod * pi_terms) +
          sum((-(1 - pi)/(pi^2)) * (pc_mat[,t1] - pc_mat[,t1]^2))
      }
    }
  }
}

# second derivative about intercept and random effects
if (type == "M.h" | type == "M.bh") {
  deriv_intc_r <- ((2/(pi^3)) * deriv_pi_intc - (1/(pi^2)) * deriv_pi_intc) *
    apply(pc_mat, 1, sum) +
    ((1/pi) - (pi^(-2))) * apply(pc_mat - pc_mat^2, 1, sum)
  deriv_intc_r <- as.matrix(deriv_intc_r, ncol = 1)

}

# second derivative about time effects and random effects
if (type == "M.th" | type == "M.tbh") {
  deriv_t_r <- matrix(NA, nrow = T, ncol = n)
  for (t in 1:T) {
    for (i in 1:n) {
      deriv_t_r[t,i] <- ((((2)*(1 - pi[i]))/(pi[i]^3)) * sum(pc_mat[i,]) -
                           ((1 - pi[i]) * sum(pc_mat[i,]))/(pi[i]^2))*(pc_mat[i, t]) +
        ((-(1 - pi[i])/(pi[i]^2)) * (pc_mat[i, t] - pc_mat[i, t]^2))
    }
  }
}

# second derivative about random effects and random effects
deriv_pi_v <- (1 - pi) * apply(pc_mat, 1, sum)
deriv_r_r <- matrix(0, nrow = n, ncol = n)
for (i in 1:n) {
  deriv_r_r[i,i] <- ((2/(pi[i]^3)) * deriv_pi_v[i] - (1/(pi[i]^2)) *
                       deriv_pi_v[i]) * sum(pc_mat[i,]) +
    ((-(1 - pi[i])/(pi[i]^2)) * sum(pc_mat[i,] - pc_mat[i,]^2))
}

# second derivative matrix
if (type == "M.h" | type == "M.bh") {
  second_d <- rbind(cbind(deriv_intc_intc, t(deriv_intc_r)),
                    cbind(deriv_intc_r, deriv_r_r))
}

if (type == "M.th" | type == "M.tbh") {
    second_d <- rbind(cbind(deriv_t_t, deriv_t_r),
                      cbind(t(deriv_t_r), deriv_r_r))
  }

  term4 <- sum(diag(second_d %*% actual_I_inv %*% second_d %*% random_I_inv))
  term5 <- 2 * sum(diag(second_d %*% random_I_inv %*% second_d %*% random_I_inv))

  term6 <- 6 * as.vector(t(new_delta) %*% second_d %*% random_I_inv %*%
                           second_d %*% new_delta)

  var_N <- term1 + term2 + term3 + term4 + term5 + term6

  # population size estimator
  adj <- sum(diag(second_d %*% random_I_inv)) # term for bias correction in N estimation
  N_est <- sum(1/pi) + adj

  # log-normal confidence interval for N
  c = exp(1.96 * sqrt(log(1 + (sqrt(var_N) / (N_est - n))^2)))
  N_upper = n + c * (N_est - n)
  N_lower = n + (N_est - n) / c

  # Wald confidence interval for N
  N_upper2 = N_est + 1.96*sqrt(var_N)
  N_lower2 = N_est - 1.96*sqrt(var_N)

  return(list(N_est = N_est, N_lower = N_lower, N_upper = N_upper,
              N_lower2 = N_lower2, N_upper2 = N_upper2))
}

############################################################

######### PART 2. SNOWSHOE HARE DATA ANALYSIS #########

# call the data set
data(hare.samp.cr)

# basic settings
snow <- hare.samp.cr$capture # observed capture history
n = nrow(snow)
T = ncol(snow)

# initial capture occasion
# index where the first capture occurs
first_obs <- apply(snow, 1, function(x) match(1, x)[1])
# indicator that an individual is captured once before each occasion
A_mat <- matrix(1, nrow = n, ncol = T)
for (ii in 1:n) {
  A_mat[ii, 1:(first_obs[ii])] <- 0
}

# response variables
# M_th
y_mat <- as.matrix(snow)
y <- as.vector(t(snow)) # vector form
y_list <- Map(function(u,v) y[u:v], seq(1,length(y),T),
                 seq(T, length(y),T)) # list form

# for M_bh and M_tbh
y_mat2 <- cbind((1 - A_mat) * y_mat, (A_mat * y_mat)[,2:T])
y2 <- as.vector(t(y_mat2))
y2_list <- Map(function(u,v) y2[u:v], seq(1,length(y2), 2*T - 1),
                  seq(2*T - 1, length(y2), 2*T - 1))

# observed design matrix
# random effects design matrices
Z <- Matrix(diag(n) %x% rep(1, T)) # for M_h and M_th
Z2 <- Matrix(diag(n) %x% rep(1, 2*T - 1)) # for M_bh and M_tbh

# for M_th
X.t <- Matrix(do.call(rbind, replicate(n, diag(T),
                                        simplify = FALSE))) # for time-varying fixed effects
X_th <- cbind(X.t, Z)
X_th_list <- Map(function(u,v) X_th[u:v,], seq(1,nrow(X_th), T),
                    seq(T, nrow(X_th), T))

# for M_bh
X.b <- rep(c(rep(0, T), rep(1, T - 1)), n) # for behavioural
X_bh <- cbind(1, X.b, Z2)
X_bh_list <- Map(function(u,v) X_bh[u:v,], seq(1,nrow(X_bh), 2*T - 1),
                    seq(2*T - 1, nrow(X_bh), 2*T - 1))

# for M_tbh
X.t <- Matrix(do.call(rbind, replicate(n, rbind(diag(T), diag(T)[2:T,]),
                                        simplify = FALSE))) # for time-varying
X_tbh <- cbind(X.t, X.b, Z2)
X_tbh_list <- Map(function(u,v) X_tbh[u:v,], seq(1,nrow(X_tbh), 2*T - 1),
                     seq(2*T - 1, nrow(X_tbh), 2*T - 1))

# find initial values
options(warn = -1)
cur_delta_th <- c(init_val(snow, ”M.th”), rnorm(n, 0 ,0))
cur_delta_bh <- c(init_val(snow, ”M.bh”), rnorm(n, 0 ,0))
cur_delta_tbh <- c(init_val(snow, ”M.tbh”), rnorm(n, 0 ,0))

# augmented responses and design matrices
data_aug_th <- aug_data(n, y, X_th, y_list, X_th_list)
data_aug_bh <- aug_data(n, y2, X_bh, y2_list, X_bh_list)
data_aug_tbh <- aug_data(n, y2, X_tbh, y2_list, X_tbh_list)

# run fitting algorithm
max_iter1 = 150
max_iter2 = 150
tol1 = 1e-13
tol2 = 1e-13
lambda_trial = c(1e-3)
fit_th <- vhglm(data_aug_th$aug_y, data_aug_th$aug_design_mat,
                  data_aug_th$aug_design_mat_list,
                  n, T, z_mat = NULL,
                  cur_delta_th, lambda_trial,
                  max_iter1, max_iter2, tol1, tol2, ”M.th”)
```

```
fit_bh <- vhglm(data_aug_bh$aug_y, data_aug_bh$aug_design_mat,
                data_aug_bh$aug_design_mat_list,
                n, T, z_mat = A_mat,
                cur_delta_bh, lambda_trial,
                max_iter1, max_iter2, tol1, tol2, ”M.bh”)
fit_tbh <- vhglm(data_aug_tbh$aug_y, data_aug_tbh$aug_design_mat,
                 data_aug_tbh$aug_design_mat_list,
                 n, T, z_mat = A_mat,
                 cur_delta_tbh, lambda_trial,
                 max_iter1, max_iter2, tol1, tol2, ”M.tbh”)

# estimate the population size
N_est_th <- est_N(fit_th$pi, fit_th$pc_mat,
                  fit_th$I, fit_th$I_inv, T, n, ”M.th”, fit_th$new_delta)
print(N_est_th)

N_est_bh <- est_N(fit_bh$pi, fit_bh$pc_mat,
                  fit_bh$I, fit_bh$I_inv, T, n, ”M.bh”, fit_bh$new_delta)
print(N_est_bh)

N_est_tbh <- est_N(fit_tbh$pi, fit_tbh$pc_mat,
                   fit_tbh$I, fit_tbh$I_inv, T, n, ”M.tbh”, fit_tbh$new_delta)
print(N_est_tbh)


#cAIC computation
cAIC_th <- 2*(T + 1 + n) -2 * (sum(rowSums(matrix(fit_th$new_eta, byrow = TRUE,
                                                   ncol = T)[1:n,] * y_mat) -
                                      log(fit_th$pi) + apply(log(1 - fit_th$pc_mat),
                                                             1, sum)) +
                                  sum(pnorm(as.vector(tail(fit_th$new_delta, n)), mean = 0,
                                            sd = sqrt(fit_th$new_lambda))))

print(cAIC_th)

cAIC_bh <- 2*(2 + 1 + n) -2 * (sum(rowSums(matrix(fit_bh$new_eta, byrow = TRUE,
                                                   ncol = 2*T - 1)[1:n,] * y_mat2) -
                                  log(fit_bh$pi) +
                                  apply((1 - A_mat) * log(1 - fit_bh$pc_mat), 1, sum) +
                                  apply((A_mat[,2:T]) * log(1 - fit_bh$pr_mat), 1, sum)) +
                                  sum(pnorm(as.vector(tail(fit_bh$new_delta, n)), mean = 0,
                                            sd = sqrt(fit_bh$new_lambda))))

print(cAIC_bh)

cAIC_tbh <- 2*(T + 1 + 1 + n) -2 * (sum(rowSums(matrix(fit_tbh$new_eta, byrow = TRUE,
                                                        ncol = 2*T - 1)[1:n,] * y_mat2) -
                                       log(fit_tbh$pi) +
                                       apply((1 - A_mat) * log(1 - fit_tbh$pc_mat), 1, sum) +
                                       apply((A_mat[,2:T]) * log(1 - fit_tbh$pr_mat), 1, sum)) +
                                       sum(pnorm(as.vector(tail(fit_tbh$new_delta, n)), mean = 0,
                                                 sd = sqrt(fit_tbh$new_lambda))))

print(cAIC_tbh)
#######################################################
```

Appendix C

R Code for analyzing Snowshoe Hare Data via the H-likelihood in Chapter 4

```
############ REQUIRED LIBRARY ############
library(splines)
##########################################

######### PART 1. FUNCTIONS FOR FITTING ALGORITMS #########

#### b-spline generation function
bspline <- function(x, xl, xr, ndx, bdeg) {
  dx <- (xr - xl) / ndx
  knots <- seq(xl - bdeg * dx, xr + bdeg * dx, by = dx)
  B <- spline.des(knots, x, bdeg + 1, 0 * x, outer.ok = TRUE)$design
  B
}

#### function for running IRLS in STEP 1 ####
VIRLS_wxh_adj_bs <- function(aug_y, aug_design_mat,
                             n, T,
                             cur_delta, cur_lambda, D = D,
                             max_iter, tol, stepsize) {

  pre_error = 1e10 # initial error value (must be large to run the next iteration)
  ori_delta = cur_delta # save the initial value of delta
  for (iter in seq_len(max_iter)) {
    print(iter)
    if (iter != 1) {
      error = sum((cur_eta - new_eta)^2) # error calculation
      print(error)
      print(tol * sum(new_eta^2))
      if ((error < tol * sum(new_eta^2)) | (iter == max_iter)) { # if error is small enough

        I_inv_sub <- I_inv[2:ncol(I_inv), 2:ncol(I_inv)]
        v_star_I_inv <- D %*% I_inv_sub %*% t(D)

        h <- diag(v_star_I_inv) / cur_lambda
    q = h
    q[q >= 1] = 0.9999 # prevent that q is over 1 (possibly by small numerical error)

    # d calculation
    d <- as.vector(tail(cur_eta, nrow(D)))

    break

  } else {
    # if divergence happens in delta estimation, run VIRLS again
    # with smaller step size of Newton's method
    if (pre_error < error) {
      stepsize = stepsize/2
    } else {
      cur_delta = new_delta
      pre_error = error
    }
  }
}

# eta computation
cur_delta <- matrix(cur_delta, ncol = 1)
cur_eta <- as.vector(aug_design_mat %*% cur_delta)

# probability computation
cur_p <- exp(cur_eta[1:(length(cur_eta) - nrow(D))]) /
  (1 + exp(cur_eta[1:(length(cur_eta) - nrow(D))]))
cur_pi <- 1 - sapply(1 - cur_p, function(x) x^T) # prob. captured at least once

# mean computation
cur_mu_posbin <- T * (cur_p / cur_pi)
cur_mu <- c(cur_mu_posbin, tail(as.vector(cur_eta), nrow(D)))

# second derivative computation
sigma_inv_diag_vals <- (T * (cur_p - cur_p^2) * (cur_pi^(-1))) -
  ((T * cur_p * (cur_pi^(-2))) * T * cur_p * (1 - cur_pi))
sigma_inv_diag_vals <- c(sigma_inv_diag_vals, rep((1 / cur_lambda), nrow(D)))

# score function INCLUDING the step of updating random effects through h-likelihood
temp = aug_y - cur_mu
temp[(length(temp) - nrow(D) + 1):(length(temp))] = tail(temp, nrow(D)) /cur_lambda
s = t(aug_design_mat) %*% temp

# fisher info
I <- t(aug_design_mat) %*% diag(sigma_inv_diag_vals) %*% aug_design_mat
I_inv <- solve(I)

# fisher info (no random contained)
I2 <- t(aug_design_mat[1:n,]) %*% diag(sigma_inv_diag_vals[1:n]) %*% aug_design_mat[1:n,]
I_inv2 <- solve(I2)

# new delta
new_delta <- cur_delta + (stepsize * I_inv %*% s)
print(new_delta)
```

```
    # new and old eta values
    cur_eta = aug_design_mat %*% cur_delta
    new_eta = aug_design_mat %*% new_delta

    if (sum(is.na(new_eta)) > 0) {
      stop("Diverged.")
    }

  } # end of for (iter in seq_len(max_iter))

  return(list(new_delta = cur_delta, new_eta = cur_eta,
              pi = cur_pi, p = cur_p,
              I = I, I_inv = I_inv, I_inv2 = I_inv2,
              h = h, q = q, d = d, stepsize = stepsize))
}

#### function for fitting gamma GLM in STEP 2 ####
vhglm_wxh_bs <- function(aug_y, aug_design_mat,
                         n, T,
                         cur_delta, cur_lambda, D = D,
                         max_iter1, max_iter2, tol1, tol2) {
  cur_eta = 0 # initial value for linear predictor (set as 0 to compute error)
  pre_error = 1e10 # initial error value (must be large to run the next iteration)
  stepsize = 1.0
  for (iter in seq_len(max_iter1)) {

    # STEP 1: estimate parameters and random-effects
    new_virls_result <- try(VIRLS_wxh_adj_bs(aug_y, aug_design_mat,
                                             n, T,
                                             cur_delta, cur_lambda, D = D,
                                             max_iter = max_iter2, tol = tol2, stepsize), TRUE)

    if (class(new_virls_result) == "try-error") {
      stop("Diverged␣in␣VIRLS.")
    }

    # STEP 2: estimate lambda (variance of random-effects)
    y_d = ((new_virls_result$d)^2) / (1 - new_virls_result$q)
    fit <- tryCatch(glm(y_d ~ 1, family = Gamma(link = "log"),
    weights = (1 - new_virls_result$q),
                        maxit = 150),
                    warning = function(w) w)

    if (inherits(fit,"warning")) {
      stop("Diverged␣in␣VIRLS␣−−␣diverged␣sigma_v")
    }

    new_lambda <- exp(coef(fit)) # updated lambda
    print(sqrt(new_lambda))

    # error calculation
    error = sum((cur_eta - new_virls_result$new_eta)^2)
    print(error)
    print((tol1) * sum((new_virls_result$new_eta)^2))
    if (error < (tol1 * sum((new_virls_result$new_eta)^2))) {
      cur_delta = new_virls_result$new_delta
      break
    }

    cur_eta = new_virls_result$new_eta
    cur_delta = new_virls_result$new_delta
    stepsize = new_virls_result$stepsize
    cur_lambda = new_lambda
    pre_error = error

    print(cur_delta[(1:ncol(D)),])
  }

  return(list(new_eta = new_virls_result$new_eta,
              new_delta = new_virls_result$new_delta, new_lambda = cur_lambda,
              var_info = diag(as.matrix(new_virls_result$I_inv)),
              pi = new_virls_result$pi,
              p =  new_virls_result$p,
              I = new_virls_result$I,
              I_inv = new_virls_result$I_inv,
              log_lambda_std = summary(fit)$coefficients[2]))
}

#### function for estimating population size ####
est_N_wxh_bs <- function(pi, p, I_inv,
                         Z, T, n, new_delta) {

  # actual var(delta)
  actual_I_inv <- I_inv

  # variance of hat N
  term1 <- sum((pi^(-2))*(1 - pi)) # variance of HT-estimator itself

  # first derivative about intercept
  d <- c(-sum((pi^(-2))*(1 - pi) * T * p))

  # first derivative about spline terms
  d <- c(d, -colSums((pi^(-2))*(1 - pi) *  Z[,-ncol(Z)] * T * p))

  term2 <- as.vector(t(d) %*% actual_I_inv %*% d)

  var_N <- term1 + term2

  # population size estimator
  N_est <- sum(1/pi)

  # log-normal confidence interval for N
  c = exp(1.96 * sqrt(log(1 + (sqrt(var_N) / (N_est - n))^2)))
  N_upper = n + c * (N_est - n)
  N_lower = n + (N_est - n) / c

  # Wald confidence interval for N
  N_upper2 = N_est + 1.96*sqrt(var_N)
  N_lower2 = N_est - 1.96*sqrt(var_N)

  return(list(N_est = N_est, se_N = sqrt(var_N), N_lower = N_lower, N_upper = N_upper,
              N_lower2 = N_lower2, N_upper2 = N_upper2))
}

##############################################################

######### PART 2. SNOWSHOE HARE DATA ANALYSIS #########

# Mountain Pygmy Possum data
mouse <- matrix(c(1, 45, 5, 40, 2, 37, 4, 45, 5, 43, 1, 45, 1, 45,
                  5, 37, 3, 38, 4, 42, 4, 38, 5, 36, 1, 41, 3, 38,
                  4, 37, 2, 41, 1, 37, 2, 45, 2, 33, 2, 41, 2, 42,
                  2, 46, 1, 47, 1, 36, 1, 43, 1, 36, 1, 40, 1, 42,
                  1, 40, 1, 47, 1, 43, 1, 31, 1, 43, 1, 43, 2, 40.22,
                  2, 33, 1, 47, 3, 37, 1, 36, 2, 44, 1, 35, 1, 38,
                  1, 49), ncol = 2, byrow = TRUE)

y <- mouse[,1] # observed count
x.h_obs <- mouse[,2] # body mass

# quadratic spline matrix (random effects matrix)
Z <- bspline(x.h_obs, min(x.h_obs), max(x.h_obs), 12, 3)

# create penalty matrix
D <- diag(ncol(Z)) # difference operator
for (k in 1:2) D <- diff(D)
D <- D[,-ncol(D)] # setting the last basis as redundant
P <- t(D) %*% D

# design matrix for fitting model
design_mat_obs <- cbind(1, Z[,-ncol(Z)])

# data augmentation
aug_y <- c(y, rep(0, nrow(D))) # y_a
aug_design_mat <- rbind(design_mat_obs,
                        cbind(matrix(0, ncol = 1, nrow = nrow(D)), D)) # T

# initial values and setups
n = nrow(mouse)
T = 5
cur_delta = rep(0, ncol(aug_design_mat))
cur_lambda = 1e-3
max_iter1 = 150
max_iter2 = 150
tol1 = 1e-13
tol2 = 1e-13

# run fitting algorithm
fit_hlike <- vhglm_wxh_bs(aug_y, aug_design_mat, n, T, cur_delta,
                          cur_lambda, D = D,
                          max_iter1, max_iter2, tol1, tol2)

# estimate the population size
```

```
est_N_wxh_bs(fit_hlike$pi, fit_hlike$p, fit_hlike$I_inv, Z, T, n, fit_hlike$new_delta)
######################################################
```

Curriculum Vitae

Name:	Han-Na Kim
Post-Secondary Education and Degrees:	Simon Fraser University Burnaby, BC 2010 - 2016 B.Sc. in Statistics
	Western University London, ON 2016 - 2017 M.Sc. in Statistics
	Western University London, ON 2017 - 2022 Ph.D. in Statistics
Related Work Experience:	Teaching Assistant Western University 2016 - 2022
	Statistical Consultant Western Data Science Solutions 2018 - 2018
	Risk Modelling Researcher Canadian Imperial Bank of Commerce 2018 - 2019

Publications:

Bonner, S., Kim, H., Westneat, D., Mutzel, A., Wright, J., & Schofield, M. (2021). dalmatian: A Package for Fitting Double Hierarchical Linear Models in R via JAGS and nimble. *Journal of Statistical Software*, 100, 1-25.

Conference Presentations:

Kim, H. and Bonner, S. (2020) H-likelihood for fitting closed capture-recapture models with unobserved heterogeneity. *Virtual International Statistical Ecology Conference.* (virtual)

Kim, H. and Bonner, S. (2021) H-likelihood for fitting capture-recapture models with heterogeneity. *Annual Meeting of Statistical Society of Canada.* (virtual)