Probability Density Approximation Methods with Applications in Ecology and Population Genetics

Wei Zhang

Department of Statistics
The University of Auckland
New Zealand

A thesis submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy in Statistics

October 2017

Acknowledgements

First, I would like to sincerely and gratefully thank my supervisor, Associate Professor Rachel Fewster, for her supervision and guidance in the past three years. She is such a nice, kind, and knowledgeable person to work with. My deep appreciation is also given to my supervisor Dr Jesse Goodman for his valuable insights and contributions to this PhD project.

My acknowledgement goes to Dr Mark Bravington for his idea of developing a saddlepoint approximation method for fitting latent multinomial models. I thank Dr Jing Liu for his helpful discussions on the application of the maximum entropy principle for the second project in this thesis. Some colleagues in my department, particularly Dr Yu Liu, Abu Zar Md Shafiullah, and Victor Miranda, gave me lots of help both in my research and daily life. They made my life in Auckland colorful and enjoyable. I thank everyone in the Statistics Department, University of Auckland for their kindness and support. It is my honor to be part of this big lovely family.

I greatly thank my family and my friends for their encouragement and support during the last three years. They were always there no matter what happened, although I was living far away from them.

Particularly I thank my wife Sunny, for her unconditional support and understanding during the course of my PhD. Without her, this work might not even have been started.

Funding for my PhD was provided in part by China Scholarship Council and the New Zealand Marsden Fund.

Abstract

Maximum likelihood estimation generally requires finding exact density or mass functions of probability distributions, which are often intractable for complicated statistical models. This PhD thesis shows that probability density approximation can be an effective tool to address this problem. Following this idea, we investigate two specific problems arising in the contexts of ecology and population genetics.

In the first project, we investigate the problem of parameter estimation under latent multinomial models, in which observed data are obtained from a linear transformation of a latent vector of counts arising from a multinomial distribution with unknown parameters. Currently, inference under these models relies primarily on Bayesian methods, which involve long computation times and often require expert implementation. In this thesis, we present a novel likelihood-based approach suitable for all models in the class, using likelihoods constructed by the saddlepoint approximation method. We validate the method by applying it to specific models for which exact or approximate likelihoods are available, by comparing it with other estimation approaches, and by simulation. The saddlepoint method consistently gives accurate inference while being considerably faster than Bayesian methods and more general than other alternative estimation approaches. We show the generality of the approach by applying it to two new models for which no existing likelihood-based approach has been proposed.

In the second project, we propose a new method for estimating the evolutionary parameters of mutation rate and recombination rate from sample data of r^2, which is a common measure of linkage disequilibrium in population genetics. The probability density function of r^2 is an unknown and complicated function of the evolutionary parameters. Our interest is focused on exploring the quantitative properties and sampling distribution of r^2. We demonstrate that a finite sequence of moments of r^2 can be computed without knowing its probability distribution under the diffusion approximation. From the moments obtained, we construct an approximate probability density function of r^2 for a two-locus genetic model using the maximum entropy principle. This density is then used for parameter estimation. The performance of the proposed method is shown by simulation studies and real data analysis.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Problem Statement

Maximum likelihood estimation is a fundamental approach for parameter estimation in statistics, and is widely used because of the attractive limiting properties it possesses such as consistency, asymptotic normality, and efficiency. To apply this approach, one first needs to specify a likelihood function, which is defined as a function of model parameters conditional on observed data. The likelihood is equivalent to the joint density for all observations, which is a function of the data conditional on the model parameters. However, finding probability density functions (PDFs) or probability mass functions (PMFs) for complicated distributions is often intractable, in which case maximum likelihood estimation can be very difficult. Here, we describe two specific problems of real interest, with intractable densities. We will investigate density approximation techniques and study the resulting approximate maximum likelihood estimators.

In the first project of this thesis (Chapters 2 and 3), we investigate the problem of parameter estimation under a class of latent multinomial models (LMMs), which is of widespread interest in many fields, including ecology, epidemiology, and social sciences. For a LMM, observed data $\boldsymbol{y}$ is a linear transformation of a latent vector of counts $\boldsymbol{z}$, where $\boldsymbol{z}$ arises from a multinomial distribution with unknown parameters. The linear relationship can be expressed as $\boldsymbol{y} = T\boldsymbol{z}$, where T is typically a known matrix with more columns than rows. We aim to estimate the parameters from the data $\boldsymbol{y}$. In this context, the density of the random variable underlying the data $\boldsymbol{y}$ is unknown, and is no longer multinomial (Link et al., 2010). Previous authors pointed out that specifying an exact likelihood function for LMMs is difficult (Link et al., 2010; Dobra et al., 2006; Sutherland and Schwarz, 2005). Thus, so far there is no general likelihood-based approach to address the parameter estimation problem for LMMs, although sophisticated tools are available in a Bayesian framework in which the latent vector $\boldsymbol{z}$ is sampled through Markov chain Monte Carlo (MCMC) methods (e.g., Schofield and Bonner, 2015; Bonner and Holmberg, 2013; Higgs et al., 2013; McClintock et al., 2013; Link et al., 2010).

The second project of this thesis (Chapters 4 and 5) addresses the problem of estimating evolutionary parameters such as population-scaled genetic mutation rate θ and recombination rate ρ, from observed data of r^2 at stationarity. Here r^2 is a common measure of linkage disequilibrium (LD) in population genetics. LD refers to the non-random association of alleles at different genetic loci for a given population. Suppose we consider two loci that exhibit alleles A_1, A_2 and B_1, B_2 respectively. Then the four possible types of gamete are A_1B_1, A_1B_2, A_2B_1, and A_2B_2. The LD measure r^2 is defined by

$$r^2 = \frac{D^2}{p\,(1-p)\,q\,(1-q)}, \tag{1.1}$$

where p and q denote the marginal frequencies of alleles A_1 and B_1, and $D = p_1 - pq = p_1 - (p_1 + p_2)\,(p_1 + p_3)$, with p_1, p_2, and p_3 denoting the frequencies of gamete types A_1B_1, A_1B_2, and A_2B_1. Thus, D is the difference between the actual frequency of gamete type A_1B_1, and the frequency that would arise if the two loci were independent. To estimate the parameters θ and ρ from sample observations of r^2 using maximum likelihood, we need the PDF of r^2 at stationarity; however, this is a complicated function of θ and ρ that has not been found so far.

1.2 Motivation

This PhD project was largely motivated by the recent work of Fewster et al. (in prep) and Liu (2012), in approximating densities of complicated probability distributions. Although they considered two problems arising in very different contexts, and applied distinct methods, their work showed that density approximation can act as a powerful tool for statistical estimation.

Fewster et al. (in prep) proposed a hybrid density approximation method for maximum likelihood estimation under the two-source capture-recapture model, which is a specific LMM. The method is of interest because it has excellent inferential performance, and is considerably faster than alternative Bayesian methods for this model (e.g., McClintock, 2015; Bonner and Holmberg, 2013; McClintock et al., 2013). The method is designed specifically for the two-source model, and cannot easily be extended to other models in the latent multinomial class; however, it shows the feasibility of density approximation for LMMs as an alternative approach to Bayesian inference, which involves long computation times and often requires expert implementation. Motivated by the idea underlying the method of Fewster et al. (in prep), we aimed to generalise it or develop a new density approximation method for all LMMs.

Investigating the stationary distribution of r^2 was initially motivated by the pioneering work of Liu (2012), who constructed the stationary distribution of the vector of gametic frequencies (p_1, p_2, p_3) for a two-locus model that will be described in Chapter 4. The underlying idea is that an unknown probability distribution can be approximated using a series of moments from the distribution, which can be computed without needing to know the distribution itself first. The idea was originally from Song and Song (2007), who developed an analytic method of computing $\mathbb{E}\left(r^2\right)$, the expectation of r^2 at stationarity, despite its stationary distribution being unknown. They showed that $\mathbb{E}\left(r^2\right)$ can be written as an infinite sum of expectations expressed in terms of (p, q, D) that can be obtained using a diffusion approximation that we describe in Chapter 4.

Using Liu (2012)'s stationary distribution for the vector (p_1, p_2, p_3), it is possible to derive the approximate stationary distribution of r^2. However, the method is high-dimensional and requires considerable computing power distributed across computer clusters. The new idea in this thesis is to extend Song and Song (2007)'s method to calculate higher moments of r^2 itself, and then directly approximate the PDF of r^2 at stationarity, which is much faster than approximating the multivariate PDF of (p_1, p_2, p_3) and using this to derive the PDF of r^2.

1.3 Objectives

We intend to develop density approximation methods to address the two problems described in Section 1.1. The primary research objectives of this thesis include:

- To develop a general approximate likelihood approach for fitting LMMs;
- To investigate several specific models in the latent multinomial class using the proposed method, and compare our estimation results with those obtained by alternative estimation approaches where these are available;

– To construct the PDF of r^2 at stationarity under a two-locus diallelic model incorporating mutation and recombination;

– To illustrate how to apply the resulting density function for maximum likelihood inference on mutation rate and recombination rate from data of r^2, and to demonstrate the performance of the method by simulation and by applying it to real data analysis.

1.4 Approach

The general idea of density approximation in this thesis is to use information we know about an unknown distribution to approximate its true PDF or PMF. For different problems, different information might be known, and thus we need to apply different approximation methods. The maximum entropy principle and the saddlepoint approximation method are two effective tools we will use in approximating probability distributions in this thesis.

For a latent multinomial variable, we can derive its moment generating function without knowing its mass function. Therefore, we apply the saddlepoint approximation method that converts a moment generating function to an approximate PMF, which is then used for maximum likelihood estimation.

For the genetic model, we extend the method of Song and Song (2007) to compute a series of moments of r^2 at stationarity, and then use the maximum entropy principle to approximate the PDF of r^2. The PDF is then used for estimating mutation rate and recombination rate from a sample of r^2.

1.5 Outline

This thesis is organised as follows. We introduce a set of specific LMMs for studying animal and human populations, and summarise some existing estimation approaches in Chapter 2. Then we derive a novel approximate likelihood approach based on the saddlepoint approximation method for fitting LMMs in Chapter 3. We validate the proposed method by applying it to specific models for which exact or approximate likelihoods are available, by comparing it with other estimation approaches, and by simulation. In Chapter 4, some background related to r^2 is presented, including the two-locus genetic model we use, the maximum entropy principle, the diffusion approximation, and Song and Song (2007)'s method. We construct the stationary PDF of r^2 using the maximum entropy principle in Chapter 5, and illustrate the procedure of how to use the density for maximum likelihood inference on mutation rate and recombination rate. Chapter 6 presents a finite difference method to find the stationary distribution for the vector (p_1, p_2, p_3). Some potential future work is discussed in Chapter 7.

Part I

Parameter Estimation Methods for Latent Multinomial Models

2

Background: Latent Multinomial Models

2.1 Overview

In this chapter, we first define notation for describing a general framework for LMMs. Then we give a brief introduction to classical capture-recapture methods. Following that, a suite of specific LMMs is introduced, including model $M_{t,\alpha}$ and the two-source model in ecology, multi-list models in epidemiology, and models for data augmentation in multi-way contingency tables given known marginal totals in social sciences. In the second half of this chapter, we provide a review of some existing estimation approaches and software for these models in the literature. Finally, we introduce the `R` package `TMB`, which is an important tool for optimisation used in this thesis. The main purpose of this chapter is to give notation and present some background for the first project.

2.2 Models and Notation

2.2.1 Notation

For LMMs, the vector of observed data $\boldsymbol{y}$ is a linear transformation of a latent (unobservable) vector $\boldsymbol{z}$, where $\boldsymbol{z}$ arises from a multinomial distribution. The two data vectors are linked by a known matrix T such that $\boldsymbol{y} = T\boldsymbol{z}$, where T is typically a matrix with more columns than rows.

Define the random vectors underlying the data vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ to be $\boldsymbol{Y} = (Y_1, \ldots, Y_I)$ and $\boldsymbol{Z} = (Z_1, \ldots, Z_J)$, where $I < J$. Then T is an $I \times J$ matrix that connects $\boldsymbol{Y}$ and $\boldsymbol{Z}$ by $\boldsymbol{Y} = T\boldsymbol{Z}$. Suppose $\boldsymbol{Z}$ follows a multinomial distribution with index N and cell probabilities $\boldsymbol{\pi}(\boldsymbol{\theta})$, a known function of $\boldsymbol{\theta}$, where N and $\boldsymbol{\theta}$ are unknown parameters. In different models, $\boldsymbol{\pi}(\boldsymbol{\theta})$ is of different forms. For simplicity, we use $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J)$ to replace $\boldsymbol{\pi}(\boldsymbol{\theta})$ hereafter.

In the context of capture-recapture and multi-list models that will be discussed below, N always denotes the size of an animal or human population, while the parameter vector $\boldsymbol{\theta}$ varies between models. The observed vector $\boldsymbol{Y}$ is a vector of frequencies of observable capture histories, and the latent vector $\boldsymbol{Z}$ is a vector of latent history frequencies. Let $\{\omega_1, \ldots, \omega_I\}$ denote the set of all observable capture histories, and let $\{\lambda_1, \ldots, \lambda_J\}$ denote the set of all latent capture histories. In our notation, Y_i is the count of the observable history ω_i, and Z_j is the count of the latent history λ_j, for $i = 1, \ldots, I$ and $j = 1, \ldots, J$. We may also use notations Y_{ω_i} and Z_{λ_j} interchangeably with Y_i and Z_j in some places. In the context of multi-way contingency tables, $\boldsymbol{Y}$ is a set of marginal totals, $\boldsymbol{Z}$ is a vector of cell entries of the table, and N is the number of entries in the table.

2.2.2 Capture-Recapture

Capture-recapture sampling is a popular method in ecology for estimating the size of animal populations. To apply the method, researchers make repeated attempts to sight or 'capture' animals in the population over a period of time, and record the encounter histories of captured animals. Then the number of animals that have never been sighted can be estimated by analysing the pattern of recaptures of the captured animals. There is considerable literature about statistical models that can be applied in capture-recapture studies (see McCrea and Morgan, 2014; Chao, 2001; Pollock, 2000; Cormack, 1979; Otis et al., 1978; Darroch, 1958, 1959, for a review), and their applications in a variety of areas (e.g., King et al., 2009; Karanth et al., 2006; Link and Barker, 2005).

Capture-recapture data typically consist of a large number of capture histories. For convenience, we use an ID number $i = 1, \dots, n$ to represent each captured animal, and use $t = 1, \dots, K$ to number each capture occasion. The data can be expressed as a matrix

$$\Lambda = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} & \dots & \Lambda_{1K} \\ \Lambda_{21} & \Lambda_{22} & \dots & \Lambda_{2K} \\ \vdots & \vdots & & \vdots \\ \Lambda_{n1} & \Lambda_{n2} & \dots & \Lambda_{nK} \end{bmatrix}, \tag{2.1}$$

where Λ_{it} is 1 if animal i was captured on occasion t, and 0 otherwise. The ith row $(\Lambda_{i1}, \dots, \Lambda_{iK})$ of the data matrix represents the encounter history of animal i, while the tth column $(\Lambda_{1t}, \dots, \Lambda_{nt})$ represents the capture results on occasion t. For example, when $K = 4$, the history $(1, 0, 1, 0)$ or simply 1010, denotes an animal that was captured on occasions 1 and 3, while not captured on occasions 2 and 4.

To analyse the data, we first need an appropriate statistical model, specifying the capture probability p_{it} of animal i on capture occasion t. In different contexts, different models can be used. For example, if we assume $p_{it} = p$ for all possible combinations of i and t, this is the simplest capture-recapture model M_0 (Otis et al., 1978), which has two parameters N and p. The assumption of constant capture probabilities in model M_0 is too ideal, so it has not gained much attention in the literature. To be more realistic, we need to consider more factors, such as heterogeneity in capture probabilities, behavioural response to capture, and variation of capture situations over time. See Otis et al. (1978) for a review of a suite of capture-recapture models incorporating these factors.

In conventional capture-recapture studies, investigators trap animals physically, and mark them with man-made tags that uniquely identify each captured animal. In more recent studies, individual identification relies increasingly on natural features of animals, including genetic markers, scars, and skin patterns, which can be recognised by, for example, DNA profiles (e.g., Vale et al., 2014; Carroll et al., 2011; Wright et al., 2009) and photographs (e.g., Bonner and Holmberg, 2013; McClintock et al., 2013; Higgs et al., 2013; Karanth et al., 2006).

Compared with traditional trapping methods, the use of natural marks has several apparent advantages. First, it requires less effort than physically capturing and marking animals. Second, the risk of causing harm or disturbance to animals is reduced, because they are often detected from a distance. Third, it is possible to obtain more sightings of animals, for example when studying populations that live in large remote areas at low densities, which may result in better precision for population estimation. However, these new techniques also introduce a potential problem of identity uncertainty, i.e., whether two observed samples belong to the same animal or to two distinct animals. Identity uncertainty transforms a model from conventional capture-recapture to a LMM. Here, we investigate two LMMs that arise from uncertain identity in capture-recapture: model $M_{t,\alpha}$ and the two-source model.

2.2.3 Models M_t and $M_{t,\alpha}$

We introduce models M_t and $M_{t,\alpha}$ following Vale et al. (2014) and Link et al. (2010). Consider a closed animal population of N individuals, which are supposed to be captured independently from each other. Model M_t (Otis et al., 1978; Darroch, 1958) assumes that each individual in the population has the same probability p_t of being captured on occasion t for $t = 1, 2, \ldots, K$. Moreover, capture outcomes for a single animal on different occasions are also assumed to be independent.

Under model M_t, each animal has two possible events on a single capture occasion: either it was captured (denoted by code 1), or it was not captured (denoted by code 0). Thus there are 2^K possible capture histories, each of which is denoted by a binary string of length K. For example, an animal with capture history 1101 indicates it was captured on occasions 1, 2, and 4, while not captured on occasion 3.

Classic capture-recapture models, including model M_t, assume that each captured animal is correctly and uniquely identified, so that the number of observed histories is equivalent to the number of distinct animals captured. However, this assumption can be questionable in practice due to misidentification of animals, which is a common problem especially when natural marks are used (see Morrison et al., 2011; Wright et al., 2009, for example). Because of misidentification, one animal's capture history might be observed as histories from two distinct animals. Consider an animal with true capture history 1011. If the animal was misidentified on occasion 3, we would observe two histories 1001 and 0010. In this case, a "ghost" history is observed, corresponding to the history of the misidentified animal, 0010; thus the number of observed histories is higher than the number of different animals captured. It has been shown that even for a low level of identification error, population size can be markedly overestimated under model M_t (Vale et al., 2014; Link et al., 2010; Wright et al., 2009).

Based on the assumptions of model M_t, model $M_{t,\alpha}$ (Link et al., 2010) takes into account the influence of misidentification on estimating population size, by incorporating a new parameter α that denotes the probability of a captured animal being correctly identified on each capture occasion. Model $M_{t,\alpha}$ assumes that each misidentification always spawns a new history with exactly one entry. That is to say, a specific identification error never occurs twice, and an individual is never misidentified as other captured individuals. The validity of this assumption might be questionable in practice (Lukacs and Burnham, 2005), but it provides some convenience for presenting the modelling framework. Extensions to model $M_{t,\alpha}$ that relax this assumption have been discussed in Link et al. (2010), Schofield and Bonner (2015), and Bonner et al. (2016).

Under model $M_{t,\alpha}$, there are three possible outcomes for each animal on occasion t: (i) it was not captured (denoted by code 0), (ii) it was captured and correctly identified (denoted by code 1), and (iii) it was captured but misidentified (denoted by code 2). These events occur with probabilities $1 - p_t$, αp_t, and $(1-\alpha)\,p_t$. The true encounter history for each animal consists of its capture outcomes on all capture occasions. For example, an animal with true capture history 1012 means it was not captured on occasion 2, was captured and correctly identified on occasions 1 and 3, and was captured but misidentified on occasion 4. Thus for K occasions, there are $J = 3^K$ true but unobservable capture histories for model $M_{t,\alpha}$. We call these latent histories.

The latent vector $\boldsymbol{Z}$ of true encounter-history counts for model $M_{t,\alpha}$ follows a multinomial distribution with index N and cell probabilities $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_J)$, where for latent history $j = 1, \ldots, J$,

$$\pi_j = \prod_{t=1}^{K} \left[p_t^{\mathcal{I}\left\{\lambda_{jt}>0\right\}} \left(1-p_t\right)^{\mathcal{I}\left\{\lambda_{jt}=0\right\}} \alpha^{\mathcal{I}\left\{\lambda_{jt}=1\right\}} \left(1-\alpha\right)^{\mathcal{I}\left\{\lambda_{jt}=2\right\}} \right], \tag{2.2}$$

with λ_{jt} denoting the capture code of history λ_j on occasion t and $\mathcal{I}\left\{\cdot\right\}$ denoting the usual indicator function.

Similarly to model M_t, observed histories for $M_{t,\alpha}$ are also in the form of binary strings of length K. Excluding the null history $00\ldots0$, we have $I = 2^K - 1$ observable histories. For convenience, all histories (latent or observable) for model $M_{t,\alpha}$ are ordered lexicographically. Unless otherwise stated, this ordering rule also applies to other models discussed subsequently.

An individual with latent history containing code 2 generates more than one observed history. For example, an animal with latent history 1221 produces three observed histories 1001, 0100, and 0010. However, when the three histories are observed, we cannot determine whether they come from an individual with latent history 1221, or from three distinct individuals with latent histories 1001, 0100, and 0010, or from two individuals with some combination of true and ghost histories. The only exception is that if we observe a history $11\ldots1$, we can be sure that it comes from an animal with the same latent history, and conversely an individual with latent history $11\ldots1$ only produces the same observed history. We call a latent history satisfying this condition a *fully-observed history.* This terminology will also be used in other models. Note that fully-observed histories also appear in the set of observable histories.

Since one latent history might produce one or more observed histories, the observed vector $\boldsymbol{Y}$ is a vector of summary statistics of the latent vector $\boldsymbol{Z}$. We use a simple example with $K = 2$ to illustrate the procedure of finding the matrix T that relates the latent vector $\boldsymbol{Z}$ to the observed vector $\boldsymbol{Y}$. For a more complex example with $K = 3$, see Link et al. (2010). In this case, we have $J = 9$ latent histories and $I = 3$ observable histories. The matrix T is of dimension 3×9. If latent history λ_j gives rise to observable history ω_i, the entry T_{ij} in the ith row and jth column of T is one;

otherwise zero. The relationship $\boldsymbol{Y} = T\boldsymbol{Z}$ can be expressed as

$$\begin{bmatrix} Y_{01} \\ Y_{10} \\ Y_{11} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Z_{00} \\ Z_{01} \\ Z_{02} \\ Z_{10} \\ Z_{11} \\ Z_{12} \\ Z_{20} \\ Z_{21} \\ Z_{22} \end{bmatrix}. \tag{2.3}$$

We can see that the number of 1s in the ith row of T represents the number of latent histories that generate the observed history ω_i. Likewise, the number of 1s in the jth column of T represents the number of observed histories that arise from the latent history λ_j.

2.2.4 The Two-Source Model

The two-source model (Fewster et al., in prep; Bonner and Holmberg, 2013; McClintock et al., 2013) arises in the context of combining two sampling protocols for inference on population size within the same capture-recapture study, for example, genetic samples and photographs. This creates analytic challenges because captures of the same animal from different protocols cannot be matched unless they were obtained simultaneously on at least one capture occasion. One way of addressing this problem is to analyse data from each protocol separately, which is inefficient since information contained in the unused protocol is ignored. In the two-source model, we assume that misidentification of animals does not occur for either of the two protocols.

Table 2.1 Possible latent capture outcomes for each animal and each occasion for the two-source capture-recapture model.

Outcome	Code
Not captured by either method	0
Captured by photo only	1
Captured by genotype only	2
Captured by both methods simultaneously	3
Captured by both methods but non-simultaneously	4

Under the two-source model, each animal in the population has five possible outcomes on each capture occasion. The outcomes and their corresponding capture codes are shown in Table 2.1. There are $J = 5^K$ possible latent histories for K capture occasions. Note that latent histories containing at least one code 3 are fully-observed. For convenience, all latent histories are grouped into: (i) the fully-observed histories; (ii) histories excluding the null history and the fully-observed histories; (iii) the null history. We order the latent histories following this ordering, and within each group histories are ordered lexicographically.

Observed histories for the two-source model include the fully-observed histories, and histories containing only records from either of the protocols. We order all these histories in the same manner: (i) the fully-observed histories; (ii) histories containing records from photographs only excluding the null history, say 10100; (iii) histories containing records from genetic samples only excluding the null history, for example, 02202. Thus, in the observed histories we use codes 1 and 2 to refer to capture by photograph or genotype respectively, but the interpretation of these codes differs from that in the latent histories. The numbers of histories included in the three observable groups are $5^K - 4^K, 2^K - 1$, and $2^K - 1$. Thus the total number of observable histories for the two-source model is $I = 5^K - 4^K + 2(2^K - 1)$.

Individuals with latent histories that consist of records from both the two protocols but without any simultaneous capture (code 3) contribute to two observed histories, one containing records from genetic samples only, and the other containing records from photographs only. For example, an individual with latent history 1204 gives rise to two observed histories 1001 and 0202. However, individuals whose latent histories are 1001 or 0202 are also observed as these two histories. Hence, we cannot distinguish whether the two observed histories come from one individual or two distinct individuals. This generates a similar latent multinomial structure to model $M_{t,\alpha}$.

2.2.5 Multi-List Models

Capture-recapture methods are also widespread in epidemiology. Similarities and differences between capture-recapture and multi-list methods are summarised in Sutherland (2003). Multi-list methods are mainly used to address human population estimation problems, in which members of the population may be present on one or more administrative lists. Different lists may use different tags to identify individuals, such as name, date of birth, or health insurance number. If there exists a tag common to all lists, such that individuals can be matched across all lists, then it is common to use Poisson log-linear models (Cormack, 1989) to estimate the population size.

The assumption of a tag common to all lists is essential for the application of Poisson log-linear models. If it does not hold, Poisson models can only be applied to a subset of the lists that share a common tag, so information contained in unused lists is ignored. To deal with this problem, Sutherland and Schwarz (2005) applied a latent Poisson model. Here, we consider the same problem as discussed by Sutherland and Schwarz (2005), but use an equivalent LMM instead.

In this section, we still use the terminology "capture history", although we do not capture any individuals in a conventional sense. The word "capture" represents being present on a list. The latent history λ_j of an individual is defined as $\left(\lambda_{j1}, \ldots, \lambda_{jK}\right)$ or simply $\lambda_{j1} \ldots \lambda_{jK}$, where λ_{jk} is 1 if the individual is on list k; otherwise 0 for $k = 1, \ldots, K$. We have $J = 2^K$ latent histories for a K-list problem. For example, when $K = 4$, an individual with latent history 1010 is on lists 1 and 3, but not on

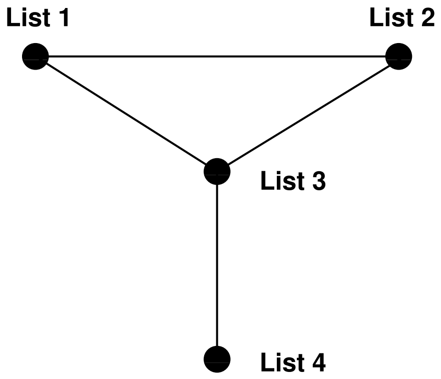


Fig. 2.1 List structure for a four-list, two-tag example.

lists 2 and 4. We assume all individuals are matched correctly between lists whenever those lists share a common tag.

Specifying the set of observable histories requires some work. Given the same number of lists, observable histories can be different for different list structures. Following Sutherland and Schwarz (2005), we use a graph such as that shown in Fig. 2.1 to illustrate the procedure of finding observable histories for multi-list problems. Lists are represented by vertices on the graph. If there exists an edge between two lists, they share a common tag; otherwise they do not. It follows that the example in Fig. 2.1 has at least two different tags, one for matching records on lists 1, 2, and 3, and another for matching records on lists 3 and 4. In this four-list, two-tag example, the record of an individual on list 4 cannot be matched with the individual's records on lists 1 and 2 unless the individual is also included on list 3, which acts as a "bridge". For this reason, latent histories with code 1 for list 3 are fully observed for this list structure. Other latent histories are not observable, but they produce observed histories containing partial information. For example, the latent history 1101 is unobservable. Instead, it is observed as two vague histories 110· and ··01, where "·" means that it is unknown whether or not an individual is on a list.

Sutherland and Schwarz (2005) showed that there are 12 observable histories for the list structure in Fig. 2.1, i.e., {100·, 010·, 110·, 1010, 1011, 0110, 0111, 0010, 0011, 1110, 1111, ··01}. These histories are ordered as they are shown in the set. They also derived the 12×16 matrix T that connects the vectors $\boldsymbol{Y}$ and $\boldsymbol{Z}$.

In the modelling framework of Sutherland and Schwarz (2005), it is assumed that

$$\begin{aligned} \boldsymbol{Y} &= T\boldsymbol{Z} \\ \boldsymbol{Z} &\sim \text{Poisson}\left(\boldsymbol{\mu}_{\boldsymbol{Z}}\right) \\ \log\left(\boldsymbol{\mu}_{\boldsymbol{Z}}\right) &= W\boldsymbol{\beta}, \end{aligned} \tag{2.4}$$

where W is a design matrix, and $\boldsymbol{\beta}$ is a vector of parameters. The vector $\boldsymbol{\beta}$ typically consists of an intercept β_0, main effects β_k for lists $k = 1, \ldots, K$, and potential interaction effects between some pairs of the lists, such as first-order interactions β_{kl} between lists k and l for $k, l \in \{1, \ldots, K\}$ with $k < l$.

Consider an example that includes two first-order interaction effects: interaction β_{12} between lists 1 and 2, and interaction β_{13} between lists 1 and 3. In this example, we have

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \tag{2.5}$$

and $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_4, \beta_{12}, \beta_{13})$. In the matrix W, the first column consists completely of one, which should be the same for most list structures. The second to fifth columns of the matrix are composed of exactly the latent capture histories. The last two columns of W deal with the two list interactions. For example, the last history 1111 includes interactions between lists 1 and 2, and lists 1 and 3, thus the entries in the last row and the last two columns of the matrix are all one. We use some examples to show how the latent vector $\boldsymbol{Z}$ is related to the model parameters:

$$\begin{aligned} \log(\mu_{1000}) &= \beta_0 + \beta_1 \\ \log(\mu_{1101}) &= \beta_0 + \beta_1 + \beta_2 + \beta_4 + \beta_{12} \\ \log(\mu_{1111}) &= \beta_0 + \beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_{12} + \beta_{13} \end{aligned} \tag{2.6}$$

where μ_{1000}, μ_{1101}, and μ_{1111} denote the expectations of Poisson variables Z_{1000}, Z_{1101}, and Z_{1111}.

In our analysis, we apply a multinomial model to describe the latent vector $\boldsymbol{Z}$, which is consistent with the Poisson model of Sutherland and Schwarz (2005):

$$\begin{aligned} \boldsymbol{Z} &\sim \text{Multinomial}\,(N; \boldsymbol{\pi}) \\ \boldsymbol{\pi} &= \frac{\exp\,(W\boldsymbol{\beta})}{\sum \exp\,(W\boldsymbol{\beta})}. \end{aligned} \tag{2.7}$$

The multinomial formulation above gives a natural way of estimating N and of accounting for covariance between elements of Y, both of which have been problematic for authors who used the Poisson formulation (e.g. Sutherland and Schwarz, 2005; Lee, 2002). The probability vector $\boldsymbol{\pi}$ does not depend on the parameter β_0, because the first column of the matrix W consists completely of one-entries so that $\exp\,(\beta_0)$ cancels in the numerator and denominator of equation (2.7). For this reason, the parameters $\boldsymbol{\theta}$ that constitute $\boldsymbol{\pi} = \boldsymbol{\pi}\,(\boldsymbol{\theta})$ in our case include all components of $\boldsymbol{\beta}$ except for the intercept β_0, and thus the two models have the same number of parameters to estimate: namely, $(\beta_0, \boldsymbol{\theta})$ for the Poisson formulation, and $(N, \boldsymbol{\theta})$ for the multinomial formulation. Note that we can also propose a different form for $\boldsymbol{\pi}$ in terms of model parameters. However, neither the Poisson nor the multinomial formulation for $\boldsymbol{Z}$ allows for ready inference based on the observable data $\boldsymbol{Y} = T\boldsymbol{Z}$, because the probability mass function of $\boldsymbol{Y}$ is not known. We use this model in Chapter 3 to compare our method with that of Sutherland and Schwarz (2005) for drawing inference on $(N, \boldsymbol{\theta})$ based on $\boldsymbol{Y}$.

2.2.6 Multi-Way Contingency Tables with Known Marginals

In social sciences, data from a survey or census are often presented in the form of a multi-way contingency table with cell entries to be modelled by a multinomial or Poisson model. However, it often arises that only partial information about the table is available, for example, a subset of marginal totals, for reasons of participant privacy or reporting convenience (Dobra et al., 2006). In this context, LMMs can be applied to model the incomplete data for inference on model parameters and individual cell entries of the table, as proposed by Schofield and Bonner (2015).

We consider a contingency table of counts over a K-dimensional discrete random vector $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_K)$. Each ξ_k represents an observed variable such as level of education or socio-economic class of a participant. For each $k \in \{1, \ldots, K\}$, the random variable ξ_k has I_k possible values, denoted by integers $1, \ldots, I_k$ for convenience. Let $(i_1, \ldots, i_K)$ or simply $i_1 \ldots i_K$ denote a single cell of the table, where i_k is the observation of ξ_k, and takes values from $\{1, \ldots, I_k\}$. Each cell therefore represents one unique combination of the K variables. It follows that the total number of cells in the table is

$$m = \prod_{k=1}^{K} I_k. \tag{2.8}$$

The cell $i_1 \ldots i_K$ has a non-negative integer cell entry $Z_{i_1 \ldots i_K}$ that represents the frequency of the random vector $\boldsymbol{\xi}$ being observed as $(i_1, \ldots, i_K)$, i.e., the number of participants with this combination of observations. For simplicity, let Z_j denote the cell entry of the jth cell of the table for $j = 1, \ldots, m$. The contingency table is an m-dimensional vector consisting of all these cell entries: $\boldsymbol{Z} = \left(Z_{1\ldots1}, \ldots, Z_{I_1 \ldots I_K}\right) = (Z_1, \ldots, Z_m)$.

A marginal table is defined as a vector of summary statistics of the full table, which can be obtained by summation over a subset of the K variables. Consider a subset $D = \{\xi_d \mid d \in \Omega\}$ with $\Omega \subseteq \{1, \ldots, K\}$. The dimension of the marginal vector $\boldsymbol{Y}_D$ corresponding to D is $\prod_{d \in \Omega} I_d$, and this vector is obtained by summing over all variables not included in D. To illustrate the definition, we consider an example with $K = 3$ and $D = \{\xi_1, \xi_2\}$. In this case, we have a two-way marginal table of dimension $I_1 I_2$:

$$\boldsymbol{Y}_D = \boldsymbol{Y}_{\{\xi_1, \xi_2\}} = \left(Z_{11+}, \ldots, Z_{I_1 I_2 +}\right), \tag{2.9}$$

where

$$Z_{i_1 i_2 +} = \sum_{i_3=1}^{I_3} Z_{i_1 i_2 i_3} \tag{2.10}$$

for $i_1 \in \{1, \ldots, I_1\}$ and $i_2 \in \{1, \ldots, I_2\}$. For a set D that includes more variables and for more complex tables, marginal tables can be defined analogously. Note that if $D' \subseteq D$ is a subset of D, the marginal table $\boldsymbol{Y}_{D'}$ can be obtained directly from the marginal table $\boldsymbol{Y}_D$.

It is straightforward to see that any marginal table can be expressed as a linear transformation of the full table. If we consider several marginal tables $\boldsymbol{Y}_{D_1}, \ldots, \boldsymbol{Y}_{D_n}$ over subsets $D_1, \ldots, D_n$ of the K variables, $\boldsymbol{Y} = (\boldsymbol{Y}_{D_1}, \ldots, \boldsymbol{Y}_{D_n})$ can be written as a linear transformation of the full table $\boldsymbol{Z}$, so that $\boldsymbol{Y} = T\boldsymbol{Z}$, where T is a matrix that consists completely of zero and one entries.

As with the formulation for multi-list studies, cell entries of the original table can be modelled by Poisson or multinomial models, thus we can also use formulas (2.4) and (2.7) to investigate inference problems for contingency tables with known marginal totals. We can regard each variable ξ_k here as a "list" to apply those formulas. The only difference is that N is no longer an unknown parameter to be estimated, as it can be obtained by summing up any one of the n marginal tables. Thus this problem differs from capture-recapture scenarios in which we seek to draw inference on N, but it still falls into the general class of LMMs.

2.3 Bayesian Inference

Currently, parameter estimation under LMMs relies mainly on a suite of Bayesian MCMC methods (e.g., Bonner et al., 2016; Schofield and Bonner, 2015; Bonner and Holmberg, 2013; McClintock et al., 2013; Link et al., 2010). The general principle of these methods is to sample the latent vector $\boldsymbol{z}$ using an MCMC sampler instead of enumerating all possible values. Before these methods were developed for LMMs, MCMC methods were already well-established for the more general problem of sampling $\boldsymbol{z}$ in the presence of the linear constraint $\boldsymbol{y} = T\boldsymbol{z}$, where $\boldsymbol{z}$ need not be multinomial. This problem particularly arose in the context of sampling multi-way contingency tables given fixed marginal totals (e.g. Dobra, 2012; Dobra et al., 2006; Chen et al., 2006, 2005; Diaconis and Sturmfels, 1998). In this section, we briefly summarise two MCMC methods for LMMs.

2.3.1 Bayesian Method for Model $M_{t,\alpha}$

Link et al. (2010) first proposed a Bayesian MCMC method for model $M_{t,\alpha}$, and indicated that the method can be extended naturally to other LMMs. Given that $\boldsymbol{y} = T\boldsymbol{z}$ with a known matrix T, their algorithm is to sample from the joint posterior distribution $[\boldsymbol{\theta}, \boldsymbol{z} \mid \boldsymbol{y}]$. The notation $[x]$ denotes $f_X(x)$, the density or mass function of random variable X. Priors are given to $\boldsymbol{\theta}$, so that $[\boldsymbol{\theta} \mid \boldsymbol{z}]$ can be sampled without difficulty. The main challenge of the algorithm lies in drawing samples from the full conditional distribution $[\boldsymbol{z} \mid \boldsymbol{y}, \boldsymbol{\theta}]$. The algorithm of Link et al. (2010) for updating the latent vector $\boldsymbol{z}$ is summarised as follows.

Algorithm 1 Bayesian MCMC method for model $M_{t,\alpha}$ (Link et al., 2010).

```
Choose an initial value $\boldsymbol{z}^0$ satisfying $\boldsymbol{y} = T\boldsymbol{z}^0$
for $i = 1$ to $n$ do
    for $k = 1$ to $m$ do
        Sample $c$ from $\{-C_k, \ldots, -1, 1, \ldots, C_k\}$ with equal probability
        Let $\boldsymbol{z}_{\text{cand}} = \boldsymbol{z}^{i-1} + c\boldsymbol{b}_k$ with $\boldsymbol{b}_k \in \mathcal{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\}$
        Calculate the Metropolis-Hastings ratio $r = \min\left\{1, \frac{[\boldsymbol{z}_{\text{cand}}|\boldsymbol{\theta}]}{[\boldsymbol{z}^{i-1}|\boldsymbol{\theta}]}\right\}$
        Accept $\boldsymbol{z}^i = \boldsymbol{z}_{\text{cand}}$ with probability $r$; otherwise $\boldsymbol{z}^i = \boldsymbol{z}^{i-1}$
    end for
end for
```

Note that the set $\mathcal{B}$ is a basis of $\ker(T)$, the kernel (i.e., null space) of matrix T, so that every element of the feasible set $\mathcal{A} = \{\boldsymbol{z} \mid \boldsymbol{y} = T\boldsymbol{z}\}$ can be expressed as the sum of one feasible solution plus a linear combination of basis elements. For model $M_{t,\alpha}$ with K capture occasions, the cardinality m of the set $\mathcal{B}$ is $3^K - 2^K + 1$. In the algorithm, n is the number of iterations which should be sufficiently large to ensure the convergence of the Markov chain, and C_k is an integer selected by the user. The algorithm therefore involves starting with one feasible solution $\boldsymbol{z}^0$, and then repeatedly sampling random vectors from the null space of matrix T to be added to $\boldsymbol{z}^0$, thereby generating new feasible solutions.

2.3.2 A More General Bayesian Method

Schofield and Bonner (2015) pointed out that Link et al. (2010)'s algorithm is at risk of failing to obtain an irreducible Markov chain in some cases. The problem arises from an incomplete specification for the set $\mathcal{B}$ in Link et al. (2010)'s algorithm. To solve the problem, they suggested to use a Markov basis of the lattice kernel $\ker_{\mathbb{Z}}(T)$ for the set $\mathcal{B}$, where

$$\ker_{\mathbb{Z}}(T) = \ker(T) \cap \mathbb{Z}^J = \left\{\boldsymbol{z} \in \mathbb{Z}^J \mid T\boldsymbol{z} = \mathbf{0}\right\}. \tag{2.11}$$

The advantage of a Markov basis over a simple basis was shown by Schofield and Bonner (2015) using three examples, in which Link et al. (2010)'s algorithm may fail to obtain an irreducible Markov chain. More details about these definitions in algebraic statistics (e.g., Markov basis) can be found in Schofield and Bonner (2015). Moreover, Schofield and Bonner (2015) changed the algorithm of Link et al. (2010) by sampling a single value of k from $\{1, 2, \ldots, m\}$ instead of going through all the m possible values, and sampling c from $\{-1, 1\}$ instead of $\{-C_k, \ldots, -1, 1, \ldots, C_k\}$. Their more general algorithm is shown below.

The two MCMC algorithms of Link et al. (2010) and Schofield and Bonner (2015) are very elegant and general, but still have several disadvantages. First, they both involve long computation times. For example, to fit a data set simulated from model $M_{t,\alpha}$ with the settings $N = 400, \alpha = 0.90$, and $\boldsymbol{p} = (0.3, 0.4, 0.5, 0.6, 0.7)$, Link et al. (2010)'s method cost over half an hour. Computation time is typically proportional to the square of the number of nonzero components of the observed data $\boldsymbol{y}$ (Bonner and Holmberg, 2013). Second, Markov bases are important for the algorithm of Schofield and Bonner (2015), but constructing them is a considerable challenge. Schofield and Bonner (2015) pointed out that the software `4ti2` they used for Markov basis construction may fail for some capture-recapture models with even a moderate number of capture occasions (say $K > 5$). Analytic computation of Markov bases is also a challenge, since substantial knowledge of algebraic statistics is needed. Third, one needs to check the convergence of Markov chains when implementing these methods, but this might not be easy in some circumstances. A typical way of doing this is to use the `R` package `coda` (Plummer et al., 2006), but this might fail to work by crashing with an error when the length of the Markov chain is too large.

Algorithm 2 Bayesian MCMC method for all LMMs (Schofield and Bonner, 2015).

```
Choose an initial $\boldsymbol{z}^0$ satisfying $\boldsymbol{y} = T\boldsymbol{z}^0$
for $i = 1$ to $n$ do
    Sample $k$ from $\{1, 2, \ldots, m\}$ with equal probability
    Sample $c$ from $\{-1, 1\}$ with equal probability
    Let $\boldsymbol{z}_{\text{cand}} = \boldsymbol{z}^{i-1} + c\boldsymbol{b}_k$ with $\boldsymbol{b}_k \in \mathcal{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\}$
    Calculate the Metropolis-Hastings ratio $r = \min\left\{1, \frac{[z_{\text{cand}}|\boldsymbol{\theta}]}{[\boldsymbol{z}^{i-1}|\boldsymbol{\theta}]}\right\}$
    Accept $\boldsymbol{z}^i = \boldsymbol{z}_{\text{cand}}$ with probability $r$; otherwise $\boldsymbol{z}^i = \boldsymbol{z}^{i-1}$
end for
```

2.3.3 The `multimark` Package

The original method of Link et al. (2010) was adapted for the two-source model by Bonner and Holmberg (2013) and McClintock et al. (2013), who also simplified the original algorithm to make it more efficient. McClintock (2015) developed an `R` package `multimark` implementing Bayesian inference to analyse capture-recapture data consisting of up to two natural marks. `multimark` further improves the efficiency of the MCMC algorithms of Bonner and Holmberg (2013) and McClintock et al. (2013), by the use of parallel computing and `C` programming instead of `R` for core algorithms. The package serves as a user-friendly software for practitioners. We describe the models that can be fitted via `multimark` here; for the use of the package, refer to McClintock (2015).

Currently the package can fit a range of models, including the Cormack-Jolly-Seber (CJS) model for open populations, and classical models for closed populations. We focus on the closed population models that can be fitted by the package. The latent vector $\boldsymbol{Z}$ for the two-source model follows a multinomial distribution: $\boldsymbol{Z} \sim \text{Multinomial}\,(N; \boldsymbol{\pi})$. In `multimark`, the jth component of $\boldsymbol{\pi}$, i.e., the cell probability of the latent history λ_j for $j = 1, \ldots, J$, is $\pi_j = \prod_{t=1}^{K} \pi_{jt}$, where

$$\pi_{jt} = \begin{cases} 1 - p_{jt} & \text{if } \lambda_{jt} = 0 \\ p_{jt}\delta_1 & \text{if } \lambda_{jt} = 1 \\ p_{jt}\delta_2 & \text{if } \lambda_{jt} = 2 \\ p_{jt}\left(1 - \delta_1 - \delta_2\right)\alpha & \text{if } \lambda_{jt} = 3 \\ p_{jt}\left(1 - \delta_1 - \delta_2\right)\left(1 - \alpha\right) & \text{if } \lambda_{jt} = 4 \end{cases} \tag{2.12}$$

with λ_{jt} denoting the capture code of the history λ_j on occasion t. Thus, p_{jt} is the overall probability of capture in this cell; conditionally on capture, δ_1, δ_2, and $1 - \delta_1 - \delta_2$ are respectively the probabilities of capture by photograph only, genotype only, or both; and conditionally on capture by both methods, α and $1 - \alpha$ are the probabilities of simultaneous or non-simultaneous capture. For example, when $K = 3$ the cell probability for the history $\lambda_j = 012$ is

$$\pi_j = \prod_{t=1}^{3} \pi_{jt} = \Big(1 - p_{j1}\Big)\, p_{j2}\delta_1 p_{j3}\delta_2. \tag{2.13}$$

We can propose different models to describe p_{jt} for all combinations of $j = 1, \ldots, J$ and $t = 1, \ldots, K$. Letting $p_{jt} = p$ yields the two-source model M_0 that includes parameters $p, \delta_1, \delta_2, \alpha$, and N. If $p_{jt} = p_t$ for all $j = 1, \ldots, J$, this is the two-source model M_t that includes parameters $\delta_1, \delta_2, \alpha, N$, and $\boldsymbol{p} = (p_1, \ldots, p_K)$. Some other models can also be fitted in `multimark`, for example, models incorporating behavioural response to captures, and models with individual heterogeneity.

2.4 Alternative Methods in the Literature

For some models in the latent multinomial class, there are alternative estimation approaches available, which are more efficient but less general compared with the Bayesian methods. Yoshizaki et al. (2011) proposed a least-squares approach for model $M_{t,\alpha}$, although without supplying a variance estimator. Sutherland and Schwarz (2005) applied a quasi-likelihood approach based on estimating functions for fitting latent Poisson models in the context of multi-list studies. However, a general non-Bayesian approach for all LMMs has not been accomplished previously.

2.4.1 Maximum Likelihood Estimation

Mathematically, a general likelihood function for LMMs can be written down easily by summing up the multinomial probabilities of all feasible latent vectors $\boldsymbol{z}$ that are compatible with the observed data $\boldsymbol{y}$:

$$\mathcal{L}\left(N, \boldsymbol{\theta} \mid \boldsymbol{y}\right) = \sum_{z \in \mathcal{A}} \mathbb{P}\left(\boldsymbol{z} \mid N, \boldsymbol{\theta}\right), \tag{2.14}$$

where $\mathcal{A} = \{\boldsymbol{z} \mid T\boldsymbol{z} = \boldsymbol{y}\}$. This looks promising, but it is computationally infeasible to use this summed likelihood for estimation in practice, since specifying or enumerating the set $\mathcal{A}$ is generally intractable except for trivial cases (Link et al., 2010; Yoshizaki et al., 2009; Dobra et al., 2006; Sutherland and Schwarz, 2005). As a consequence, fitting LMMs using maximum likelihood does not follow readily from equation (2.14).

Vale et al. (2014) derived an exact closed-form likelihood function for model $M_{t,\alpha}$, and showed that it can be computed efficiently, but commented that their formulation does not generalise to other models in the LMM class. So far, model $M_{t,\alpha}$ is the only LMM whose exact likelihood function is available by tractable computation. Thus we will use model $M_{t,\alpha}$ to demonstrate in Chapter 3 the performance of our proposed method of likelihood approximation, by comparing inference under our approximate likelihood function with that under the exact likelihood. For more details about the exact likelihood formulation for model $M_{t,\alpha}$, see Vale et al. (2014). Compared with 30 minutes required by the Bayesian method of Link et al. (2010) to fit the instance of model $M_{t,\alpha}$ as mentioned in Section 2.3.2, Vale et al. (2014) reported fitting the same model in 1.2 seconds using their exact likelihood computation.

2.4.2 Hybrid Approximation

Maximum likelihood is highly appealing for fitting LMMs because of its fast computational speed, as demonstrated in Vale et al. (2014). When exact likelihood functions are difficult to specify, approximate likelihoods may be attainable for some models. For example, Fewster et al. (in prep) proposed a hybrid density approximation method to gain an approximate likelihood for the two-source model, which can then be maximised as usual. Here we briefly describe this hybrid approximation method.

Fewster et al. (in prep) showed that the linear relationship $\boldsymbol{Y} = T\boldsymbol{Z}$ for the two-source model can be reformulated as

$$\boldsymbol{Y} = (\boldsymbol{C}, \boldsymbol{X}) = (\boldsymbol{C}, A\boldsymbol{U}), \tag{2.15}$$

where $\boldsymbol{C}$ is a vector of counts of fully-observed histories and histories with zero count, $\boldsymbol{X}$ is a vector of counts of all remaining histories, and the conditional distribution of $\boldsymbol{U}$ given $\boldsymbol{C}$ is multinomial. The multinomial PMF of the vector $\boldsymbol{C}$ is known, so finding the PMF of $\boldsymbol{Y}$ is equivalent to finding the PMF of $\boldsymbol{X} \mid \boldsymbol{C} = A\boldsymbol{U} \mid \boldsymbol{C}$, where A is a submatrix of T.

Using the multivariate central limit theorem (CLT), the probability distribution of the multinomial vector $\boldsymbol{U}$ can be approximated by a multivariate normal distribution. Thus the distribution of $\boldsymbol{X}$ can also be approximated by a multivariate normal distribution, since normality is preserved under the linear transformation $\boldsymbol{X} = A\boldsymbol{U}$. This looks promising, but it is not useful in practice. The reason is that the normal approximation by CLT is only effective asymptotically as $N \to \infty$, which means that cell counts in $\boldsymbol{U}$ should be large, leading to large cell counts in $\boldsymbol{X}$. However, in real applications, many cells of $\boldsymbol{X}$ typically have small counts of 0, 1, 2, $\ldots$, especially when the number of capture occasions is high but capture probabilities are low. In this context, the multivariate normal approximation cannot yield a satisfactory approximation to the PMF of $\boldsymbol{U}$, so the transformed normal approximation to the PMF of $\boldsymbol{X}$ is also unsatisfactory.

Furthermore, Fewster et al. (in prep) proved that, if $\boldsymbol{X}'$ is a multinomial vector with the same mean vector as $\boldsymbol{X}$, the covariance matrix of $\boldsymbol{X}'$ is identical to that of $\boldsymbol{X}$ in many entries but differs in some entries. These discrepancies are minor as long as the cell probabilities of $\boldsymbol{X}$ are all small. This suggests that if $\boldsymbol{X}$ is entirely composed of small counts, a multinomial approximation to $\boldsymbol{X}$ might be satisfactory. Simulation studies showed that treating $\boldsymbol{X}$ as a pseudo-multinomial vector can generate unbiased inference results with reasonable confidence interval coverage when most components of $\boldsymbol{X}$ are sufficiently small, say less than five.

The multivariate normal approximation is effective when $\boldsymbol{X}$ contains only large components, while the multinomial moment-based approximation is effective when $\boldsymbol{X}$ consists only of small counts. The idea of the hybrid approximation method is to combine the two approximations, such that large counts of $\boldsymbol{X}$ are dealt with by the multivariate normal approximation, while small counts are handled by the pseudo-multinomial approximation. This hybrid approximation was studied in extensive simulation studies by Fewster et al. (in prep) and found to give negligible bias and nominal confidence interval coverage across a wide range of scenarios.

2.4.3 Hybrid Approximation compared with multimark

The development of the hybrid approximation method is not part of my original work, but I have created an implementation in TMB (introduced in the next section), and performed comparisons with the multimark package to verify the method. Implementing the method in TMB is faster and more stable than native R implementations, and more user-friendly than the previous implementation in ADMB (Fournier et al., 2012).

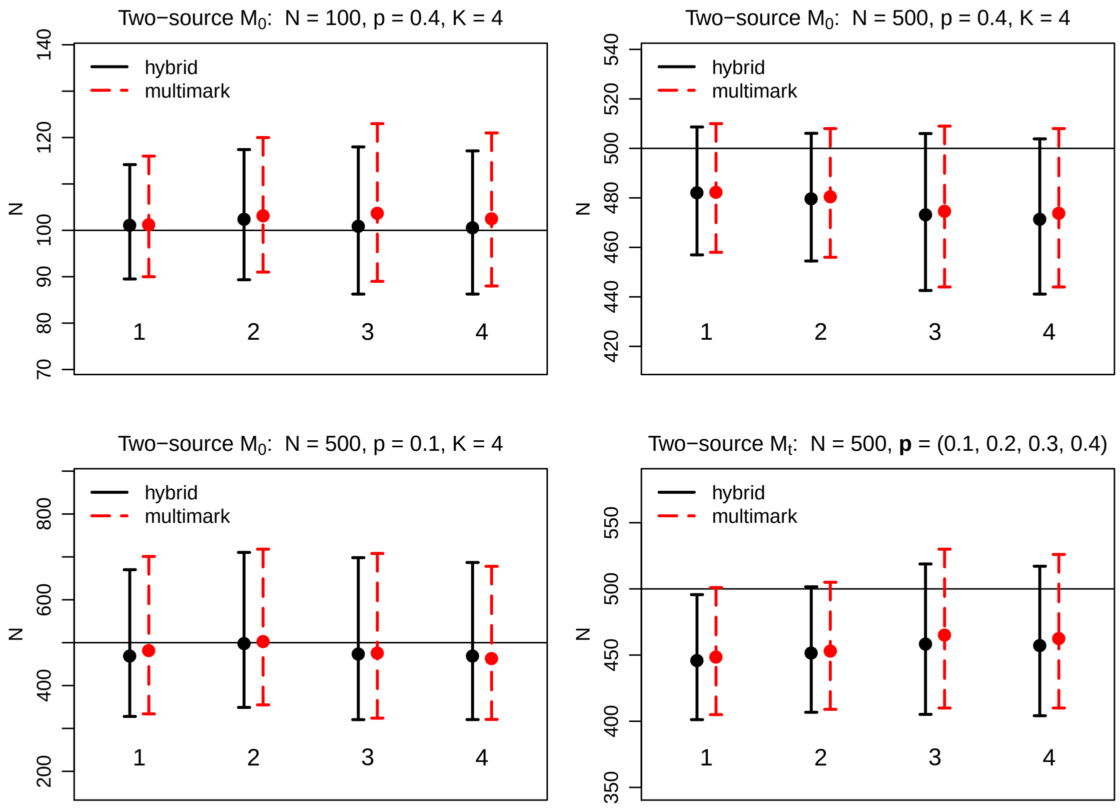


Fig. 2.2 Plots of estimates and 95% confidence intervals (or credible intervals) for the parameter N obtained by multimark (dashed) and the hybrid approximation approach (solid) for a range of settings under the two-source models M_0 and M_t. Each dot represents an estimate from one simulated data set. Horizontal lines across the plots show the true values of N. The parameter δ_1 is fixed at 0.4 for all scenarios. The parameters (δ_2, α) are set to (0.2, 0.1), (0.2, 0.5), (0.4, 0.1), and (0.4, 0.5) respectively for scenarios 1, 2, 3, and 4.

We present comparisons between the hybrid approximation and `multimark` by applying the two methods to simulated data with different parameter settings under the two-source models M_0 and M_t. It is impracticable to run a large number of simulations since fitting one data set using `multimark` even in simple cases (e.g., $K = 4$ capture occasions) can cost over half an hour. Therefore, for each setting we only generated one data set by simulation and calculated parameter estimates and associated confidence intervals using the two methods. Fig. 2.2 shows our results for 16 scenarios. We focus on the parameter N, which is of main interest in closed capture-recapture studies. We can see from the interval plots that the point estimates and confidence intervals of N from the two distinct approaches are always close to each other. This provides evidence for the validity of both methods.

The primary motivation for developing the hybrid approximation method for the two-source model was that it could be considerably faster than the Bayesian MCMC algorithms (e.g., McClintock, 2015; Bonner and Holmberg, 2013; McClintock et al., 2013). Computation times for fitting one data set using the `multimark` package for each of eight scenarios are shown in Table 2.2. All of them are over 30 minutes. In contrast, fitting one data set using the hybrid method cost less than one second in each of the eight scenarios.

We conclude that approximating the likelihood has the potential to deliver inferential performance that matches that of the Bayesian method, at a fraction of the computational cost. We are therefore motivated to explore whether a more general approximation method can be developed for all LMMs. This is the focus of Chapter 3. We will use the exact likelihood for $M_{t,\alpha}$ from Vale et al. (2014) and the hybrid approximation for the two-source model from Fewster et al. (in prep) to validate the new approximation developed there. The purpose of validating the hybrid approximation against `multimark` in this section was to confirm that the hybrid approximation creates a suitable reference method against which our new method can be validated.

Table 2.2 Estimates and 95% confidence intervals (credible intervals) for the parameter N under the two-source model M_0 obtained using `multimark` and the hybrid approximation method, for eight of the settings that are shown by interval plots in Fig. 2.2. Computation times for the hybrid method in all eight scenarios were less than one second on a customary laptop.

$(N, p, \delta_1, \delta_2, \alpha)$	$\widehat{N}_m$	CI_m	$\widehat{N}_h$	CI_h	CI_m/CI_h	`multimark` time / min
$(500, 0.1, 0.4, 0.2, 0.1)$	481	$[334, 701]$	469	$[327, 670]$	1.07	37
$(500, 0.1, 0.4, 0.2, 0.5)$	502	$[355, 718]$	498	$[349, 710]$	1.00	35
$(500, 0.1, 0.4, 0.4, 0.1)$	475	$[324, 708]$	473	$[321, 698]$	1.02	36
$(500, 0.1, 0.4, 0.4, 0.5)$	463	$[321, 678]$	469	$[321, 687]$	1.00	33
$(500, 0.4, 0.4, 0.2, 0.1)$	482	$[458, 510]$	482	$[457, 509]$	1.00	40
$(500, 0.4, 0.4, 0.2, 0.5)$	480	$[456, 508]$	480	$[454, 506]$	1.00	35
$(500, 0.4, 0.4, 0.4, 0.1)$	475	$[444, 509]$	473	$[443, 506]$	1.03	38
$(500, 0.4, 0.4, 0.4, 0.5)$	474	$[444, 508]$	471	$[441, 504]$	1.02	35

$\widehat{N}_m$: posterior mean of N from `multimark`; $\widehat{N}_h$: estimate of N from the hybrid method; CI_m: 95% credible interval for N from `multimark`; CI_h: 95% confidence interval for N from the hybrid method; CI_m/CI_h: ratio of the length of CI_m to the length of CI_h.

2.5 Introduction to TMB

This thesis focuses on maximum likelihood problems, so fast, accurate, and stable optimisation algorithms are indispensable. We primarily rely on the R optimiser nlminb for minimizing negative log-likelihood functions. Like most routines, the performance of nlminb is enhanced if users supply explicit gradient functions along with objectives, especially for complicated problems. However, symbolic differentiation is almost impossible for complex likelihoods, for example, likelihoods for random-effect models. The connection between our methods and random-effect models will become clear later.

Template Model Builder (TMB) is a fast and flexible R package to facilitate optimisation for complex models with or without random effects (Kristensen et al., 2016). It implements automatic differentiation, which repeatedly applies the chain rule to the elementary functions comprising the likelihood, enabling it to calculate highly accurate derivatives of user-defined functions to the same precision as symbolic differentiation, but without needing to do any symbolic differentiation at all. It also implements the Laplace approximation to facilitate fast computations of integrals involved in random-effect models. The TMB package is used for most computations involved in this thesis, and it is particularly valuable for our method of fitting LMMs, so it deserves a brief introduction here. We mainly focus on the use of the package. For more detail about automatic differentiation, refer to Griewank and Walther (2008); the Laplace approximation will be described in Chapter 3.

2.5.1 Example

To use TMB to minimise a negative log-likelihood function, one needs to write the objective function in a C++ template, while completing other steps in R such as data processing. We demonstrate the use of TMB by fitting a simple linear regression model to simulated data, an example taken from the package website https://github.

com/kaskr/adcomp/tree/master/tmb_examples. For more examples, refer to this website and Section 5 of Kristensen et al. (2016).

We consider the linear regression model $Y = a + bX + \epsilon$, where ϵ follows a normal distribution $N\left(0, \sigma^2\right)$. Suppose that a sequence of data pairs $(x_1, y_1), \ldots, (x_n, y_n)$ is simulated from the model $Y_i = a + bX_i + \epsilon_i$, where $\epsilon_1, \ldots, \epsilon_n$ are independent and identically distributed $N\left(0, \sigma^2\right)$. The likelihood function for this example is

$$l\left(a, b, \sigma\right) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - a - bx_i\right)^2\right\}. \tag{2.16}$$

Our first step is to code the negative log-likelihood as a C++ function using the template, which is named as "linreg.cpp". The code is as follows.

```
#include <TMB.hpp>
template<class Type>
Type objective_function<Type>::operator() ()
{
  DATA_VECTOR(y);
  DATA_VECTOR(x);
  PARAMETER(a);
  PARAMETER(b);
  PARAMETER(logSigma);
  Type nll = -sum(dnorm(y, a+b*x, exp(logSigma), true));
  return nll;
}
```

We explain this program line by line. The first four lines and the last line of the program are standard, and should be the same in most cases. The first line of this template is used to include the TMB macros. The following three lines are required by the syntax of a C++ function template. Note that Type is a special type of C++ object defined by TMB that will be replaced by an automatic differentiation type for numerical computations when the template is compiled (Kristensen et al., 2016). The code DATA_VECTOR(y) declares that y is a data vector, and x is declared in the same manner. Note that x and y should be passed from R. The following three lines use PARAMETER to declare scalar parameters a, b, and logSigma. If we have parameter vectors to declare, PARAMETER_VECTOR should be used instead. A normal PDF is given by the function dnorm, which is defined similarly to the corresponding R function. The fourth argument true indicates that the logarithm of the density will be returned, rather than the density itself. The template can be compiled and linked from within R using the following code.

```
library(TMB)
compile("linreg.cpp")
dyn.load(dynlib("linreg"))
dat <- list(y=rnorm(10) + 1:10, x=1:10)
par <- list(a=0, b=0, logSigma=0)
obj <- MakeADFun(dat, par, DLL="linreg")
est <- nlminb(obj$par, obj$fn, obj$gr)
rep <- sdreport(obj)
```

The first line of the R program above loads the TMB package. Then the C++ template is compiled and linked using functions compile and dyn.load through the following two lines. The object dat is a list containing values to be delivered to the data objects declared in the template. Note that the components of the list should have the same names as the objects declared in the C++ code, namely y and x in this example. Likewise, the values contained in par will be assigned to the declared parameters as starting values for the optimisation. Then the function MakeADFun creates an object obj that contains: (i) obj$par, the starting values for optimisation; (ii) obj$fn, the negative log-likelihood function, (iii) obj$gr, the gradient function obtained by automatic differentiation, and (iv) obj$he, a function to evaluate the Hessian matrix of the objective. If some of the parameters are random effects that can be integrated out using the Laplace approximation, we need to assign the parameter names to the argument random of the function MakeADFun. For example, if we use random=c("logSigma") in this problem, the parameter logSigma would be treated as a random effect. Note that for models with random effects, obj$he is not available at present.

The last two lines of the program implement the `R` function `nlminb` to minimise `obj$fn`, and use `sdreport` to return outputs, including parameter estimates, standard errors, and the maximum gradient component to help diagnose whether convergence has been achieved. Note that the ordering of the last two lines makes a difference to the result of `rep`, although it appears that the second last line does not change anything involving `obj`. To see this, we can regard `TMB` as an assistant of `R`. When implementing `nlminb` to find parameter estimates, `R` needs to evaluate functions `obj$fn`, `obj$gr`, and `obj$he` multiple times at a range of different parameter values; however, these computations are done by `TMB` instead of `R`. `TMB` will remember the values of parameters at which these functions are evaluated for the last time, thus when `nlminb` obtains the parameter estimates, `TMB` also knows what the estimates are, and stores them as part of the `obj` object. Then `sdreport` can be used to summarise `obj` at the parameter estimates. If `sdreport` is called before `nlminb` is used for optimisation, the function will return a summary of `obj` at the starting values. Thus, the likelihood optimisation is carried out by native `R` functions, but using `TMB` to create the gradient and Hessian functions and to evaluate each of these throughout the optimisation.

3

A New Approximate Likelihood Method for Latent Multinomial Models

3.1 Overview

In this chapter, we aim to develop a novel approximate likelihood approach to address the problem of parameter estimation for LMMs. We start by describing a likelihood factorization for these models. Then we develop a saddlepoint approximation to the factor of the likelihood that cannot be derived analytically. We then illustrate the implementation of the proposed method in TMB. Finally, we apply the saddlepoint method to the models introduced in Chapter 2, and compare its performance with that of alternative estimation approaches where these are available.

3.2 Likelihood Factorization

In LMMs, given the observed vector $\boldsymbol{y}$, the latent vector $\boldsymbol{z}$ generally has a substantial number of feasible values; otherwise the summed likelihood function (2.14) could be used for estimation. However, it is possible that the components of $\boldsymbol{z}$ can be partially determined by the vector $\boldsymbol{y}$. If $T_{ij} = 1$ is the only nonzero entry in the ith row of the matrix T, it follows that $z_j = y_i$. For example, we have $z_{1\ldots1} = y_{1\ldots1}$ in model $M_{t,\alpha}$, and $z_{\lambda_j} = y_{\omega_i}$ in the two-source model if $\lambda_j = \omega_i$ is a history containing at least one simultaneous capture (i.e., a fully-observed history). Furthermore, if $\boldsymbol{y}$ has a component y_k observed to be zero so that $0 = y_k = z_l + z_m$, the components z_l and z_m of $\boldsymbol{z}$ are known to be zero. In practice, it is commonplace that the vector $\boldsymbol{y}$ has many components observed to be zero for model $M_{t,\alpha}$ and the two-source model, especially when the number of capture occasions is high while capture probabilities are low.

The examples above show the possibility that some components of $\boldsymbol{z}$ can be determined from the observed vector $\boldsymbol{y}$. However, this does not necessarily happen for all models. For example, ideally no cell entry of a multi-way table can be deduced from a marginal table: this is often why marginal tables are reported, so that the individual cell entries cannot be deduced because of privacy concerns (Dobra et al., 2006). Even for a single model, the components of $\boldsymbol{y}$ that equal zero might not be the same in different data realisations, so the components of $\boldsymbol{z}$ known to be zero will differ according to the data.

The likelihood factorization we describe below is important for the validity of the saddlepoint method, and can also improve efficiency. For models in which no component of $\boldsymbol{z}$ can be determined from $\boldsymbol{y}$, we can skip the factorization step and proceed directly to the last two paragraphs of this section. Reasons for the factorization step will be given in more detail later.

Suppose we can determine a unique solution for R elements of $\boldsymbol{z}$ from the observed vector $\boldsymbol{y}$. We reorder the elements of vector $\boldsymbol{z}$ such that it can be written as

$$\boldsymbol{z} = (\boldsymbol{v}, \boldsymbol{u}), \tag{3.1}$$

where $\boldsymbol{v} = (z_1, \ldots, z_R)$ contains all verified elements of $\boldsymbol{z}$, and $\boldsymbol{u} = (z_{R+1}, \ldots, z_J)$ contains all remaining (unverified) elements. Accordingly, we reorder the elements of $\boldsymbol{\pi}$, and the columns of the matrix T. Then, we continue to use the equation $\boldsymbol{y} = T\boldsymbol{z}$, although $\boldsymbol{z}$ and T have been reordered.

We partition matrix T as

$$T = \left[\begin{array}{c:c} B & A \end{array}\right], \tag{3.2}$$

where B is an $I \times R$ matrix that contains the first R columns of T, and A is an $I \times (J - R)$ matrix that contains the remaining $J - R$ columns. It follows that

$$\boldsymbol{y} = T\boldsymbol{z} = B\boldsymbol{v} + A\boldsymbol{u}, \tag{3.3}$$

and thus

$$A\boldsymbol{u} = \boldsymbol{y} - B\boldsymbol{v}. \tag{3.4}$$

For convenience, let $\boldsymbol{x} = \boldsymbol{y} - B\boldsymbol{v}$. Since $\boldsymbol{y}$, $\boldsymbol{v}$ and B are all known, $\boldsymbol{x}$ is a known vector. However, the equation $A\boldsymbol{u} = \boldsymbol{x}$ generally has impractically many solutions for $\boldsymbol{u}$.

The likelihood function for LMMs can be written as

$$\begin{aligned} \mathcal{L}\left(N, \boldsymbol{\theta} \mid \boldsymbol{y}\right) &= \sum_{\boldsymbol{z}:T\boldsymbol{z}=\boldsymbol{y}} \mathbb{P}\left(\boldsymbol{Z} = \boldsymbol{z}\right) \\ &= \sum_{\boldsymbol{u}:A\boldsymbol{u}=\boldsymbol{x}} \mathbb{P}\left(\boldsymbol{Z}_{1:R} = \boldsymbol{v} \bigcap \boldsymbol{Z}_{R+1:J} = \boldsymbol{u}\right) \qquad (3.5) \\ &= \mathbb{P}\left(\boldsymbol{Z}_{1:R} = \boldsymbol{v}\right) \sum_{\boldsymbol{u}:A\boldsymbol{u}=\boldsymbol{x}} \mathbb{P}\left(\boldsymbol{Z}_{R+1:J} = \boldsymbol{u} \mid \boldsymbol{Z}_{1:R} = \boldsymbol{v}\right), \end{aligned}$$

where random vector $\boldsymbol{Z}_{1:R}$ contains the first R components of the multinomial vector $\boldsymbol{Z}$, and $\boldsymbol{Z}_{R+1:J}$ contains all remaining components of $\boldsymbol{Z}$.

By the multinomial marginal property, we have

$$\mathbb{P}\left(\boldsymbol{Z}_{1:R} = \boldsymbol{v}\right) = \frac{N!}{z_1! \dots z_R! \left(N - v^*\right)!} \left(\prod_{j=1}^{R} \pi_j^{z_j}\right) \left(1 - \sum_{j=1}^{R} \pi_j\right)^{N-v^*}, \qquad (3.6)$$

where $v^* = \sum_{j=1}^{R} z_j$ denotes the sum of all elements of vector $\boldsymbol{v}$. Then our problem reduces to finding

$$\sum_{\boldsymbol{u}:A\boldsymbol{u}=\boldsymbol{x}} \mathbb{P}\left(\boldsymbol{Z}_{R+1:J} = \boldsymbol{u} \mid \boldsymbol{Z}_{1:R} = \boldsymbol{v}\right). \qquad (3.7)$$

Define $\boldsymbol{U}_{\boldsymbol{v}}$ to be a random variable following the conditional distribution of $\boldsymbol{Z}_{R+1:J} \mid \boldsymbol{Z}_{1:R} = \boldsymbol{v}$. By the conditional property of multinomial distributions, we have

$$\boldsymbol{U}_{\boldsymbol{v}} \sim \text{Multinomial}\left(\widetilde{N}; \widetilde{\boldsymbol{\pi}}\right), \qquad (3.8)$$

where $\widetilde{N} = N - v^*$ and

$$\widetilde{\boldsymbol{\pi}} = \frac{1}{\sum_{j=R+1}^{J} \pi_j} \left(\pi_{R+1}, \dots, \pi_J\right). \qquad (3.9)$$

It follows that the summed probability (3.7) is equivalent to

$$\sum_{\boldsymbol{u}:A\boldsymbol{u}=\boldsymbol{x}} \mathbb{P}\left(\boldsymbol{U}_{\boldsymbol{v}} = \boldsymbol{u}\right) \; = \; \mathbb{P}\left(A\boldsymbol{U}_{\boldsymbol{v}} = \boldsymbol{x}\right). \qquad (3.10)$$

Now, the problem reduces to finding $\mathbb{P}\left(A\boldsymbol{U_v}=\boldsymbol{x}\right)$, where $\boldsymbol{U_v}$ is specified in (3.8) and (3.9). Let $\boldsymbol{X}=A\boldsymbol{U_v}$. Then the problem is solved if we can find the PMF of $\boldsymbol{X}$ given a known matrix A and a multinomial distribution $\boldsymbol{U_v}$ of dimension $J-R$. For brevity, we use $\boldsymbol{U}$ to replace $\boldsymbol{U_v}$ hereafter. In addition, we use $\widetilde{\boldsymbol{\pi}}=(\widetilde{\pi}_1,\ldots,\widetilde{\pi}_H)$ to replace the original formula (3.9), where $H=J-R$.

The matrix A is of dimension $I\times H$; however, it is often not of full row rank, for example, it may have some rows composed completely of zero entries (see Appendix A for an example). For models where we skip the factorization step and work with the original formulation $\boldsymbol{y}=T\boldsymbol{z}$, we may find similarly that T is not of full row rank. The saddlepoint method does not work properly in these cases; reasons will be given in the following section. Thus, some tuning for the linear relationship $\boldsymbol{x}=A\boldsymbol{u}$ or $\boldsymbol{y}=T\boldsymbol{z}$ is necessary.

The strategy described below is based on the notation $\boldsymbol{x}=A\boldsymbol{u}$. We need to find a submatrix comprising maximally independent rows of matrix A. To accomplish this, we start by using the first non-zero row of matrix A as the submatrix, and add other rows of A one by one into the submatrix to update it. Every time a new row is added, we check the row rank of the new submatrix; this can readily be done with the `R` function `qr` (QR decomposition). If the rank increases by one, we accept the new row and update the submatrix; otherwise we reject it. Accordingly, we can obtain a subset of vector $\boldsymbol{x}$, which is linked to the full vector $\boldsymbol{u}$ by the submatrix. For brevity, we still use the notations A and $\boldsymbol{x}$ to denote the submatrix and the subvector. Assume matrix A is now of dimension $L\times H$ and vector $\boldsymbol{x}$ is of dimension L, where $L\leq I$. Note that this method does not influence estimation results since the data points left out are redundant. This is analogous to the observation that the last cell of a multinomial vector provides no further information if all other cells of the vector are known.

3.3 Saddlepoint Approximation Method

The saddlepoint method was first proposed by Daniels (1954) to pursue an approximation to the PDF of the sum of independent and identically distributed (i.i.d) random variables, when their moment generating function is known. The saddlepoint approximation to the cumulative distribution function (CDF) of the sum of i.i.d variables was introduced in Lugannani and Rice (1980). More details about the derivation of the saddlepoint method can be found in Goutis and Casella (1999) and Reid (1988). It is based on a Taylor expansion of the integrand in the inversion formula used for converting a moment generating function to the corresponding PDF. The Taylor expansion is taken about a 'saddlepoint', $\hat{s}\,(w)$, specific to the value w at which the probability density is to be evaluated. The use of an expansion customized to each point w makes the approximation very accurate, but introduces computational challenges. Butler (2007) provided a comprehensive review of the theory related to the method, and demonstrated its practical value using some real data examples at a more accessible level than previous texts.

Although the initial purpose of the saddlepoint method was to approximate densities or distribution functions of sums of random variables, the same formulas can be applied for any distribution. The approximation is particularly useful in cases where exact distributions of random variables are intractable, but their moment generating functions can readily be found, which is the case in our latent multinomial scenario. The saddlepoint method has gained a range of applications in many areas, including finite population models (Wang, 1993), confidence intervals for bootstrapping (DiCiccio et al., 1992), generalised linear models (Strawderman et al., 1996), and resampling methods (Davison and Hinkley, 1988).

In this section, we aim to provide a saddlepoint approximation to the PMF of $\boldsymbol{X} = A\boldsymbol{U}$, where A is an $L \times H$ matrix, and $\boldsymbol{U} \sim \text{Multinomial}\left(\widetilde{N}; \widetilde{\boldsymbol{\pi}}\right)$ with $\widetilde{\boldsymbol{\pi}} = (\widetilde{\pi}_1, \ldots, \widetilde{\pi}_H)$. The derivation procedure below also applies to approximating the PMF of $\boldsymbol{Y} = T\boldsymbol{Z}$, where $\boldsymbol{Z} \sim \text{Multinomial}\,(N; \boldsymbol{\pi})$ if the likelihood factorization is not needed.

The joint moment generating function (MGF) of $\boldsymbol{X}$ is

$$M_{\boldsymbol{X}}\left(\boldsymbol{s}\right) = \mathbb{E}\left\{\exp\left(\boldsymbol{s}^T \boldsymbol{X}\right)\right\} = \mathbb{E}\left[\exp\left\{\left(A^T \boldsymbol{s}\right)^T \boldsymbol{U}\right\}\right] = M_{\boldsymbol{U}}\left(A^T \boldsymbol{s}\right), \tag{3.11}$$

where $M_{\boldsymbol{U}}$ is the joint MGF of $\boldsymbol{U}$, and $\boldsymbol{s} = (s_1, \ldots, s_L)$ takes values in $\mathbb{R}^L$ for which the expectation of $\exp\left(\boldsymbol{s}^T \boldsymbol{X}\right)$ exists. Let $\boldsymbol{t} = A^T \boldsymbol{s} = (t_1, \ldots, t_H) \in \mathbb{R}^H$. Since $\boldsymbol{U}$ follows a multinomial distribution, whose joint MGF is known, we have

$$M_{\boldsymbol{X}}\left(\boldsymbol{s}\right) = M_{\boldsymbol{U}}\left(\boldsymbol{t}\right) = \left\{\sum_{h=1}^{H} \widetilde{\pi}_h \exp\left(t_h\right)\right\}^{\widetilde{N}} \tag{3.12}$$

and the cumulant generating function (CGF) of $\boldsymbol{X}$, which by definition is the logarithm of $M_{\boldsymbol{X}}\left(\boldsymbol{s}\right)$, is:

$$K_{\boldsymbol{X}}\left(\boldsymbol{s}\right) = \log M_{\boldsymbol{X}}\left(\boldsymbol{s}\right) = \widetilde{N} \log\left\{\sum_{h=1}^{H} \widetilde{\pi}_h \exp\left(t_h\right)\right\}. \tag{3.13}$$

Following the formula provided by Butler (2007), the saddlepoint mass function of any random variable $\boldsymbol{X}$ is

$$\tilde{f}_{\boldsymbol{X}}\left(\boldsymbol{x}\right) = \frac{1}{\left(2\pi\right)^{L/2} \left|K''_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}\right)\right|^{1/2}} \exp\left\{K_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}\right) - \hat{\boldsymbol{s}}^T \boldsymbol{x}\right\}, \tag{3.14}$$

where $\hat{\boldsymbol{s}}$ solves the saddlepoint equation

$$K'_{\boldsymbol{X}}\left(\boldsymbol{s}\right) = \boldsymbol{x}, \tag{3.15}$$

and $\left|K''_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}\right)\right|$ denotes the determinant of the matrix $K''_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}\right)$. Thus, the saddlepoint approximation offers a way of inverting the CGF to generate an approximate PMF at any value of $\boldsymbol{x}$.

For the models discussed in this thesis, matrix T is composed of zero and one entries. Then the support of $\boldsymbol{X}$ is $\mathcal{X} = \left\{\boldsymbol{x} \in \mathbb{N}^L \mid f_{\boldsymbol{X}}(\boldsymbol{x}) > 0\right\}$. Let $\mathcal{I}_{\mathcal{X}} \subseteq \mathbb{R}^L$ denote the interior of the convex hull of $\mathcal{X}$. The saddlepoint mass function $\tilde{f}_{\boldsymbol{X}}(\boldsymbol{x})$ is computable for any vector $\boldsymbol{x} \in \mathcal{I}_{\mathcal{X}}$ (Butler, 2007); however, it is only practically meaningful for integer-valued vectors $\boldsymbol{x} \in \mathcal{I}_{\mathcal{X}} \cap \mathcal{X} \subseteq \mathbb{N}^L$. More properties of the saddlepoint method are listed in Section 3.1 of Butler (2007); detailed discussions can be found in the references therein.

The saddlepoint equation (3.15) can be expanded as

$$\frac{\partial K_{\boldsymbol{X}}(\boldsymbol{s})}{\partial s_l} = \frac{\widetilde{N} \sum_{h=1}^{H} A_{lh} \widetilde{\pi}_h \exp(t_h)}{\sum_{h=1}^{H} \widetilde{\pi}_h \exp(t_h)} = x_l, \quad l = 1, \ldots, L, \tag{3.16}$$

where A_{lh}, the entry in row l and column h of A, comes from the first-order derivative of

$$t_h = \sum_{l=1}^{L} A_{lh} s_l \tag{3.17}$$

with respect to s_l. When the matrix A consists completely of non-negative entries, the saddlepoint equation does not have a finite solution if any component of $\boldsymbol{x}$ equals zero, because equation (3.16) demonstrates that the left-hand part of equation (3.15) is positive for all entries. Matrix A consists of non-negative entries for all of the models in this thesis. For this reason, in cases where $\boldsymbol{y}$ contains zeros we must use the likelihood factorization described in Section 3.2 to ensure that $\boldsymbol{x}$ has no component that equals zero, and therefore enable the saddlepoint method to be applied.

In most cases, the saddlepoint equation cannot be solved analytically. In practice we regard it as an optimisation problem, whose solution can be obtained by numerical methods, such as the Newton-Raphson method. The optimisation formulation is described later in Section 3.4.2. The efficiency and accuracy of solving this inner problem is important for maximum likelihood estimation using the saddlepoint mass function (3.14), because each evaluation of the likelihood involves an optimisation to find $\hat{\boldsymbol{s}}$.

Substituting equations (3.6) and (3.14) into equation (3.5) generates a complete approximation to the likelihood $\mathcal{L}\left(N,\boldsymbol{\theta}\mid\boldsymbol{y}\right)$:

$$\widetilde{\mathcal{L}}\left(N,\boldsymbol{\theta}\mid\boldsymbol{y}\right)=\mathbb{P}\left(\boldsymbol{Z}_{1:R}=\boldsymbol{v}\right)\tilde{f}_{\boldsymbol{X}}\left(\boldsymbol{x}\right), \tag{3.18}$$

where $\mathbb{P}\left(\boldsymbol{Z}_{1:R}=\boldsymbol{v}\right)$ is given by equation (3.6), and $\tilde{f}_{\boldsymbol{X}}\left(\boldsymbol{x}\right)$ approximates the second factor in (3.5) via the formulation in (3.10).

Finally, we explain why we must ensure that matrix A is of full row rank, as mentioned in Section 3.2. If the factorization is not needed, equivalently we must ensure that matrix T is of full row rank. The derivation below for the case of $\boldsymbol{x}=A\boldsymbol{u}$ also applies to the case of $\boldsymbol{y}=T\boldsymbol{z}$.

Butler (2007) pointed out that the CGF $K_{\boldsymbol{X}}\left(\boldsymbol{s}\right)$ must be a strictly convex function, to ensure that the saddlepoint equation has a solution and the square root of $|K''_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}\right)|$ included in (3.14) is defined and strictly positive. It can be seen from the saddlepoint equation (3.16) that if A is not of full row rank, $\partial K_{\boldsymbol{X}}\left(\boldsymbol{s}\right)/\partial s_l$, $l=1,\ldots,L$ are not linearly independent, so that the Hessian matrix $K''_{\boldsymbol{X}}\left(\boldsymbol{s}\right)$ is not of full rank. For example, if the sum of the first row and second row of A equals the third row, then the same linear relationship is inherited by the elements of $K'_{\boldsymbol{X}}\left(\boldsymbol{s}\right)$ according to (3.16):

$$\frac{\partial K_{\boldsymbol{X}}\left(\boldsymbol{s}\right)}{\partial s_1}+\frac{\partial K_{\boldsymbol{X}}\left(\boldsymbol{s}\right)}{\partial s_2}=\frac{\partial K_{\boldsymbol{X}}\left(\boldsymbol{s}\right)}{\partial s_3}; \tag{3.19}$$

thus the third row of $K''_{\boldsymbol{X}}\left(\boldsymbol{s}\right)$ with elements $\frac{\partial^2 K_{\boldsymbol{X}}(\boldsymbol{s})}{\partial s_3 \partial s_j}$ is also the sum of the first two rows, so $|K''_{\boldsymbol{X}}\left(\boldsymbol{s}\right)|=0$. The same applies to any linear relationship among the rows of A. Thus A must have full row rank in order for the saddlepoint mass function to be well defined, because $|K''_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}\right)|=0$ appears in the denominator of (3.14). This also implies A must have no rows consisting entirely of zero entries.

3.3.1 Variance Estimation and Confidence Intervals

The maximum likelihood estimates $\left(\widehat{N}, \widehat{\boldsymbol{\theta}}\right)$ of the parameters $(N, \boldsymbol{\theta})$ are calculated by minimising the negative logarithm of the approximate saddlepoint likelihood in the usual manner:

$$- \log \widetilde{\mathcal{L}} \left(N, \boldsymbol{\theta} \mid \boldsymbol{y}\right) = - \log \mathbb{P} \left(\boldsymbol{Z}_{1:R} = \boldsymbol{v}\right) - \log \tilde{f}_{\boldsymbol{X}} \left(\boldsymbol{x}\right). \tag{3.20}$$

We estimate the variances of the parameter estimates using the main diagonal of the negative inverse of the Hessian matrix of the approximate log-likelihood function evaluated at the maximum likelihood estimates.

Following Vale et al. (2014), lognormal confidence intervals and Normal confidence intervals are calculated for N and $\boldsymbol{\theta}$ respectively. For N, the 95% confidence interval is given by $\left(\widehat{N}/a, a\widehat{N}\right)$, where

$$a = \exp\left[\mu_{0.025}\sqrt{\log\left\{1 + \widehat{\operatorname{var}}\left(\widehat{N}\right)/\widehat{N}^2\right\}}\right] \tag{3.21}$$

with $\widehat{\operatorname{var}}\left(\widehat{N}\right)$ denoting the estimate of the variance of $\widehat{N}$, and $\mu_{0.025}$ denoting the upper 0.025 point of the Normal(0, 1) distribution. For any component θ_i of $\boldsymbol{\theta}$, the 95% confidence interval is $\left(\widehat{\theta}_i - b, \widehat{\theta}_i + b\right)$, where $b = \mu_{0.025}\sqrt{\widehat{\operatorname{var}}\left(\widehat{\theta}_i\right)}$, and $\widehat{\operatorname{var}}\left(\widehat{\theta}_i\right)$ denotes the estimate of the variance of $\widehat{\theta}_i$.

3.4 Implementation in `TMB`

The saddlepoint approximation is a powerful tool in approximating complicated distributions; however, to our knowledge it has not yet contributed greatly to statistical inference in practice. This might be partially due to the fact that specialised software is needed for computation.

The approximate negative log-likelihood function (3.20) consists of two parts. The first part, $-\log \mathbb{P}\left(\boldsymbol{Z}_{1:R} = \boldsymbol{v}\right)$, is a straightforward function and can be easily coded using `TMB`. The difficulty lies in the second part, $-\log \tilde{f}_{\boldsymbol{X}}\left(\boldsymbol{x}\right)$, which cannot be handled directly because of the inner optimisation problem it involves. This optimisation complicates the characterisation of the objective function in terms of elementary functions, as required for automatic differentiation to be performed. In this section, we show how the `TMB` package is designed in such a way that we can overcome this difficulty.

3.4.1 The Laplace Approximation

The Laplace approximation plays a significant role in `TMB` for fitting random-effect models. It is also the key to our implementation of the saddlepoint method using the `TMB` package. We give a brief introduction to the Laplace approximation following Skaug and Fournier (2006) and Kristensen et al. (2016).

Suppose $l\left(\boldsymbol{\mu}, \boldsymbol{\gamma}\right)$ denotes the joint negative log-likelihood function of a statistical model with random effects $\boldsymbol{\mu} \in \mathbb{R}^m$ and fixed parameters $\boldsymbol{\gamma} \in \mathbb{R}^n$. The maximum likelihood estimate of $\boldsymbol{\gamma}$ can be obtained by maximising the marginal likelihood,

$$\mathscr{L}\left(\boldsymbol{\gamma}\right) = \int_{\mathbb{R}^m} \exp\left\{-l\left(\boldsymbol{\mu}, \boldsymbol{\gamma}\right)\right\} d\boldsymbol{\mu}, \tag{3.22}$$

which is a function of only the fixed model parameters after the random effects are integrated out. However, calculating the integral is not easy, especially when the dimension m of the random effects is high.

Define $\hat{\boldsymbol{\mu}}\left(\boldsymbol{\gamma}\right)$ to be the minimiser of $l\left(\boldsymbol{\mu}, \boldsymbol{\gamma}\right)$ with respect to $\boldsymbol{\mu}$, so that

$$\hat{\boldsymbol{\mu}}\left(\boldsymbol{\gamma}\right) = \arg\min_{\boldsymbol{\mu}} l\left(\boldsymbol{\mu}, \boldsymbol{\gamma}\right). \tag{3.23}$$

This minimisation is treated as an inner problem in `TMB`, and can be handled by the classical Newton's method. Then the Laplace approximation to $\mathscr{L}\left(\boldsymbol{\gamma}\right)$ is

$$\mathscr{L}^*\left(\boldsymbol{\gamma}\right) = (2\pi)^{\frac{m}{2}} \det\left\{H\left(\boldsymbol{\gamma}\right)\right\}^{-\frac{1}{2}} \exp\left\{-l\left(\hat{\boldsymbol{\mu}}, \boldsymbol{\gamma}\right)\right\}, \tag{3.24}$$

where $H\left(\boldsymbol{\gamma}\right) = H\left(\hat{\boldsymbol{\mu}}, \boldsymbol{\gamma}\right) = l''_{\boldsymbol{\mu}\boldsymbol{\mu}}\left(\hat{\boldsymbol{\mu}}, \boldsymbol{\gamma}\right)$ is the Hessian matrix of $l\left(\boldsymbol{\mu}, \boldsymbol{\gamma}\right)$ with respect to $\boldsymbol{\mu}$ and evaluated at $\hat{\boldsymbol{\mu}}$. Maximising $\mathscr{L}^*\left(\boldsymbol{\gamma}\right)$ is equivalent to minimising the negative logarithm of $\mathscr{L}^*\left(\boldsymbol{\gamma}\right)$, which is

$$-\log \mathscr{L}^*\left(\boldsymbol{\gamma}\right) = -\frac{m}{2}\log(2\pi) + \frac{1}{2}\log\det\left\{H\left(\boldsymbol{\gamma}\right)\right\} + l\left(\hat{\boldsymbol{\mu}}, \boldsymbol{\gamma}\right). \tag{3.25}$$

Although the joint negative log-likelihood function $l\left(\boldsymbol{\mu}, \boldsymbol{\gamma}\right)$ is coded in the `C++` function template, the `MakeADFun` function from `TMB` returns the objective (3.25) and its gradient function, when $\boldsymbol{\mu}$ is declared as a vector of random effects. Thus, `TMB` has specific functionality for maximising expressions of the form (3.22) based on the Laplace approximation.

3.4.2 Correspondence with the Saddlepoint Approximation

Starting from the saddlepoint mass function $\tilde{f}_{\boldsymbol{X}}\left(\boldsymbol{x}\right)$ in equation (3.14), some algebra gives:

$$-\log \tilde{f}_{\boldsymbol{X}}\left(\boldsymbol{x}\right) = \frac{L}{2}\log(2\pi) + \frac{1}{2}\log\det\left\{K''_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}\right)\right\} - h\left(\hat{\boldsymbol{s}}\right), \tag{3.26}$$

where

$$h\left(\boldsymbol{s}\right) = K_{\boldsymbol{X}}\left(\boldsymbol{s}\right) - \boldsymbol{s}^T\boldsymbol{x}, \tag{3.27}$$

and $\hat{\boldsymbol{s}}$ is the solution to

$$h'\left(\boldsymbol{s}\right) = K'_{\boldsymbol{X}}\left(\boldsymbol{s}\right) - \boldsymbol{x} = \mathbf{0}. \tag{3.28}$$

According to Butler (2007), the Hessian matrix $K''_{\boldsymbol{X}}\left(\boldsymbol{s}\right)$ with respect to $\boldsymbol{s}$ is positive definite after applying the likelihood factorization described in Section 3.2. It follows that finding $\hat{\boldsymbol{s}}$ is equivalent to minimising $h\left(\boldsymbol{s}\right)$ with respect to $\boldsymbol{s}$.

Thus, from equation (3.20) the full saddlepoint objective function that we wish to minimise in TMB is:

$$\begin{aligned} -\log \widetilde{\mathcal{L}}\left(N, \boldsymbol{\theta} \mid \boldsymbol{y}\right) &= -\log \mathbb{P}\left(\boldsymbol{Z}_{1:R} = \boldsymbol{v}\right) + \frac{L}{2}\log(2\pi) + \frac{1}{2}\log\det\left\{K''_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}\right)\right\} - h\left(\hat{\boldsymbol{s}}\right) \\ &= \frac{L}{2}\log(2\pi) + \frac{1}{2}\log\det\left\{K''_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}\right)\right\} - g\left(\hat{\boldsymbol{s}}, N, \boldsymbol{\theta}\right) \\ &= \frac{L}{2}\log(2\pi) + \frac{1}{2}\log\det\left\{K''_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}, N, \boldsymbol{\theta}\right)\right\} - g\left(\hat{\boldsymbol{s}}, N, \boldsymbol{\theta}\right), \end{aligned} \tag{3.29}$$

where

$$g\left(\hat{\boldsymbol{s}}, N, \boldsymbol{\theta}\right) = \log \mathbb{P}\left(\boldsymbol{Z}_{1:R} = \boldsymbol{v}\right) + h\left(\hat{\boldsymbol{s}}, N, \boldsymbol{\theta}\right). \tag{3.30}$$

Through equations (3.9) and (3.13), we have $K_{\boldsymbol{X}}\left(\boldsymbol{s}\right) = K_{\boldsymbol{X}}\left(\boldsymbol{s}, N, \boldsymbol{\theta}\right)$ and thus $h\left(\boldsymbol{s}\right) = h\left(\boldsymbol{s}, N, \boldsymbol{\theta}\right)$. From equation (3.30), we have

$$g''_{\boldsymbol{ss}}\left(\hat{\boldsymbol{s}}, N, \boldsymbol{\theta}\right) = h''_{\boldsymbol{ss}}\left(\hat{\boldsymbol{s}}, N, \boldsymbol{\theta}\right) = K''_{\boldsymbol{X}}\left(\hat{\boldsymbol{s}}\right) \tag{3.31}$$

and

$$\hat{\boldsymbol{s}}\left(N, \boldsymbol{\theta}\right) = \arg\min_{\boldsymbol{s}} h\left(\boldsymbol{s}, N, \boldsymbol{\theta}\right) = \arg\min_{\boldsymbol{s}} g\left(\boldsymbol{s}, N, \boldsymbol{\theta}\right). \tag{3.32}$$

Equation (3.29) is exactly analogous to equation (3.25), with $\hat{\boldsymbol{s}}$ replacing $\hat{\boldsymbol{\mu}}$, $(N, \boldsymbol{\theta})$ replacing $\boldsymbol{\gamma}$, and $g\left(\hat{\boldsymbol{s}}, N, \boldsymbol{\theta}\right)$ replacing $l\left(\hat{\boldsymbol{\mu}}, \boldsymbol{\gamma}\right)$, except for two sign changes in the first and third terms of equation (3.29).

Motivated by this similarity, we copied the source code of the function MakeADFun in TMB, and changed the requisite two signs (see Appendix B for details). By declaring vector $\boldsymbol{s}$ to be a vector of random effects, and defining our objective function to be $g\left(\boldsymbol{s}, N, \boldsymbol{\theta}\right)$, we can therefore deploy the modified version of TMB to generate an efficient optimisation of $-\log \widetilde{\mathcal{L}}$ including a gradient function calculated using automatic differentiation. Consequently, the inner optimisation problem for $\hat{\boldsymbol{s}}$ is entirely taken care of by TMB.

3.5 Method Validation using Model $M_{t,\alpha}$

In this section, we provide an overall validation for the proposed method before it is put into use. First, we assess the performance of the saddlepoint method for parameter estimation by applying it to simulated data from model $M_{t,\alpha}$, whose exact likelihood formulation is available. Second, the computational speed of the saddlepoint method for this model is compared with that of the Bayesian approach of Link et al. (2010). Third, the performance of the saddlepoint method for likelihood approximation is checked, by plotting the approximate log-likelihood curves against the exact curves, which are available for model $M_{t,\alpha}$ from Vale et al. (2014).

3.5.1 Comparison with the Exact Likelihood

Recall that model $M_{t,\alpha}$ involves K capture occasions on a closed population of size N, with capture probabilities $p_1, \ldots, p_K$ and probability α that an animal is correctly identified on each capture.

We simulated data sets in `R` using every combination of parameter values chosen from $K \in \{4, 6, 8, 10\}$, $N \in \{400, 1000\}$, $\alpha \in \{0.8, 0.9, 0.95, 0.97\}$, and $p_1 = \cdots = p_K \in \{0.1, 0.2, 0.3, 0.4\}$. We calculated maximum likelihood estimates and 95% confidence intervals for the parameters using the saddlepoint method and the exact likelihood function of Vale et al. (2014). The saddlepoint computation used `TMB`, and the exact computation used `ADMB` (Fournier et al., 2012), which is a similar tool to `TMB` that predated `TMB`'s development. For each setting, we generated 500 data sets, and found the maximum likelihood estimates by both methods for each data set.

Figure 3.1 exhibits a comparison between parameter estimates obtained using the two methods, in the setting of $N = 400$, $\alpha = 0.97$, and $p_1 = \cdots = p_8 = 0.1$. It shows that the saddlepoint method consistently gives almost identical estimation results to the exact likelihood function in this scenario. Estimates of $\boldsymbol{p}$ and α from the two approaches are typically identical to four or five decimal places, while estimates of N

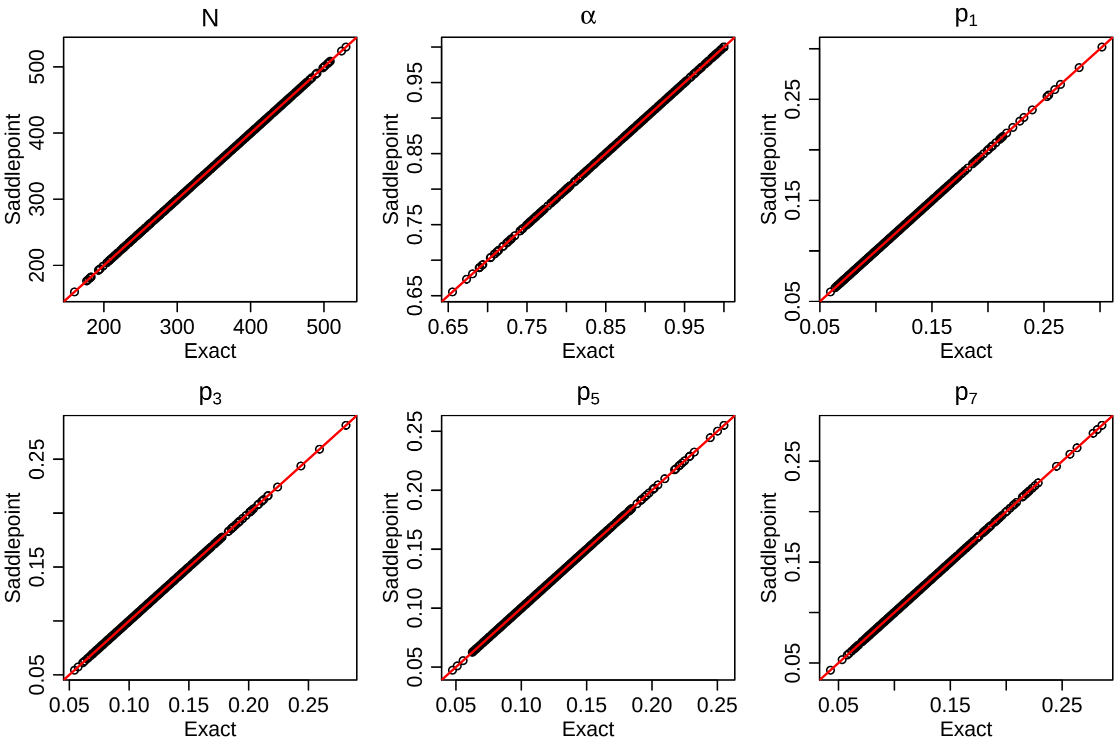


Fig. 3.1 Scatter plots of parameter estimates from 500 simulations obtained using the saddlepoint approximation method and the exact likelihood of Vale et al. (2014), in the setting of $N = 400$, $\alpha = 0.97$, and $p_1 = \cdots = p_8 = 0.1$ under model $M_{t,\alpha}$. Points on straight lines across the plots indicate that estimates from the two approaches are almost identical.

are typically identical to one or two decimal places. The two methods also produced almost the same variances for the estimates, so confidence intervals for all parameters from the two methods were almost identical. For all the other settings listed above, as well as many others outside this range, we always obtained similar scatter plots with the appearance of bold straight lines as shown in Fig. 3.1. Another example is shown in Fig. 3.2. We have not observed any case where the two approaches yielded noticeably different estimates or confidence intervals.

Average fitting time using the saddlepoint method for one simulated data set in the setting of Fig. 3.1 was roughly 10 seconds on a customary laptop with a clock speed of 1.3 GHz, while approximately 4 seconds was needed using the exact likelihood function on the same machine. This is not surprising because every evaluation of the saddlepoint likelihood function involves an optimisation problem, namely solving the

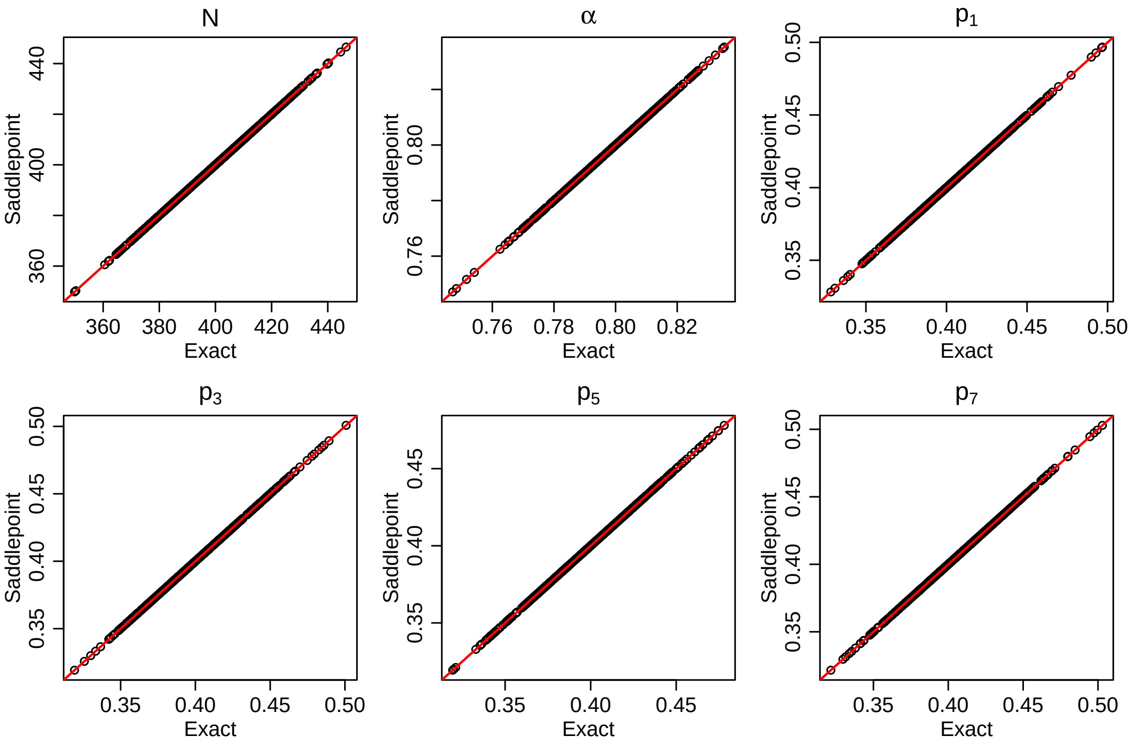


Fig. 3.2 Scatter plots of parameter estimates from the saddlepoint approach and the exact likelihood method for model $M_{t,\alpha}$ when $N = 400, \alpha = 0.8$, and $p_1 = \cdots = p_8 = 0.4$. Other details are the same as Fig. 3.1.

saddlepoint equation (3.15) numerically. Inspection of numerous results indicates that the complexity of the optimisation problem is primarily determined by the dimension of the vector $\boldsymbol{x}$, which may increase when either K or p_t increases. For this reason, in the setting of Fig. 3.2, average fitting time for one data set increased to 100 seconds, since capture probabilities in that case are much higher. However, the performance of the saddlepoint method is still excellent.

3.5.2 Comparison with the Bayesian Approach

Although the saddlepoint method is slightly slower than the exact likelihood approach of Vale et al. (2014) for model $M_{t,\alpha}$, we anticipate it will be significantly faster than the Bayesian method of Link et al. (2010). To demonstrate this, we use an example from Link et al. (2010) with $\boldsymbol{y}$ = (54, 41, 35, 30, 24, 17, 29, 25, 17, 11, 20, 8, 15, 9, 17, 17, 11, 7, 13, 6, 9, 6, 11, 3, 7, 4, 8, 3, 5, 4, 6) for $K = 5$ capture occasions. This

Table 3.1 Parameter estimates from the saddlepoint method and the Bayesian method applied to the data set simulated by Link et al. (2010) under model $M_{t,\alpha}$. Note that Link et al. (2010)'s parameter estimates were posterior means.

Method	$\widehat{N}$ (95% CI)	$\widehat{\alpha}$	$\widehat{\boldsymbol{p}}$
Link et al. (2010)	399.4 (370, 432)	0.91	$(0.302, 0.407, 0.499, 0.596, 0.704)$
Saddlepoint approximation	397.9 (366, 433)	0.91	$(0.302, 0.407, 0.500, 0.598, 0.706)$

data set was generated using $N = 400$, $\alpha = 0.9$, and $(p_1, \ldots, p_5) = (0.3, 0.4, 0.5, 0.6, 0.7)$ under model $M_{t,\alpha}$.

Table 3.1 shows a comparsion between estimation results from the two approaches, which are extremely close to each other. Note that the results from our method are almost identical to those obtained by Vale et al. (2014). The saddlepoint method cost 0.9 seconds on the 1.3 GHz laptop, while Link et al. (2010) indicated that the Bayesian method cost over 30 minutes on a 3.8 GHz machine.

3.5.3 Evaluation of the Saddlepoint Approximation

Here, we investigate the performance of the saddlepoint approximation in reproducing the likelihood curves for model $M_{t,\alpha}$. This verification process is also based on simulations. We first generated data sets by simulation using a broad range of settings. For each setting, we made a comparison between exact and saddlepoint log-likelihoods, which were both expressed only in terms of the parameter N, with $\alpha, p_1, \ldots, p_K$ fixed at their maximum likelihood estimates under the exact method. A suite of examples with different settings is shown in Fig. 3.3.

From Fig. 3.3, and many similar plots that are not presented, we see that in most cases the log-likelihoods obtained using the two methods do not match each other perfectly; however, the two functions differ by an almost constant value. In different settings, the constant may be different. This explains why the two methods generate almost identical estimates and confidence intervals, as shown in the previous simulation studies.

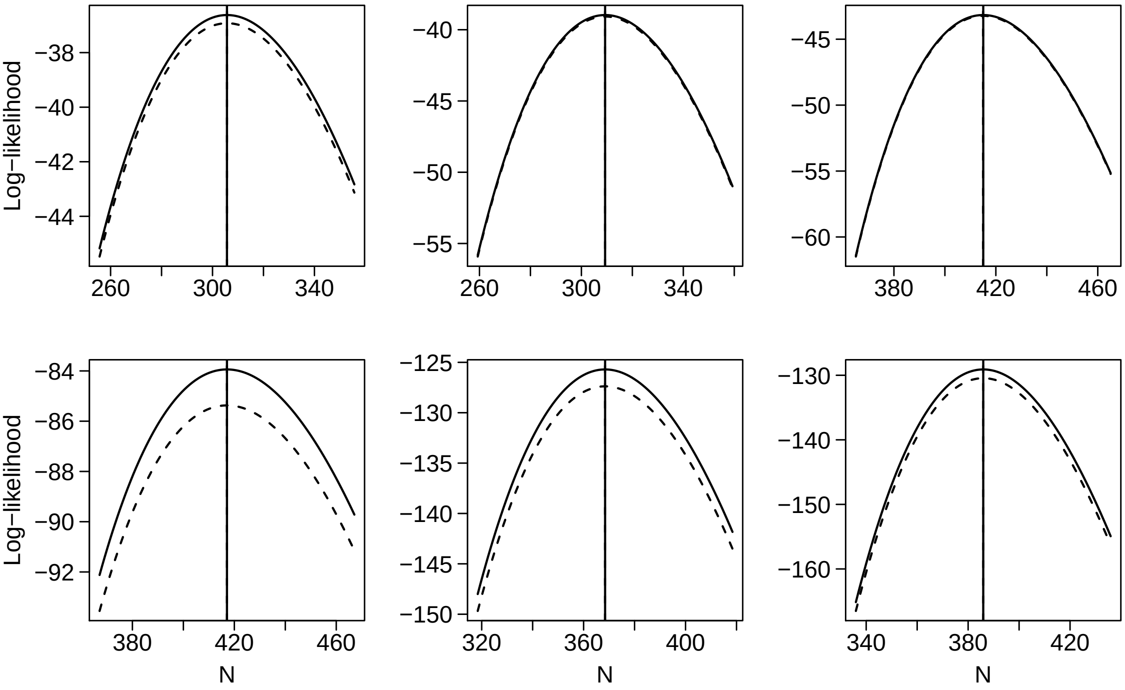


Fig. 3.3 Saddlepoint log-likelihood curves (solid) shown against exact log-likelihood curves (dashed) for model $M_{t,\alpha}$. We used $N = 400$ and $\alpha = 0.97$ for all six panels. We set $p_1 = \cdots = p_4$ to be 0.2, 0.3, and 0.4 for the three panels in the top row (left to right), and $p_1 = \cdots = p_6$ to be 0.2, 0.3, and 0.4 for the three panels in the bottom row (left to right). Vertical lines show the positions of maxima under the saddlepoint computation (solid) and the exact computation (dashed). These cannot be distinguished easily because the saddlepoint estimates and the exact estimates of N are extremely close to one another.

The difference between the saddlepoint and exact log-likelihoods appears to decrease when p_t increases, or when the number K decreases while other parameters remain the same. Each of these scenarios leads to an increase in the proportion of relatively large components in the vector $\boldsymbol{x}$. Inspection of numerous results indicates that if all or most components of the vector $\boldsymbol{x}$ are greater than or equal to five, the saddlepoint method yields an extremely good approximation to the exact likelihood function. For example, the second and third panels from the left in the top row of Fig. 3.3 present two scenarios where all components of $\boldsymbol{x}$ are over five. When the number of capture occasions increases, it is more difficult to observe a vector $\boldsymbol{x}$ with most components larger than five, so differences between the two log-likelihoods are larger for the three scenarios shown in the bottom row.

Therefore, we suggest that the saddlepoint method can be used with confidence for parameter estimation under LMMs without any obvious concerns. However, for approximation of the exact likelihood curves, it can be trusted only if most components of the observed vector are relatively large.

3.6 Application to the Two-Source Model

3.6.1 Two-Source Model M_0

We consider the simplest two-source model M_0 for studying closed populations. Suppose we use data from two protocols, photographs and genetic samples. In the two-source model M_0, capture probabilities for the five capture codes $(0 - 4)$ remain the same for all animals throughout all capture occasions.

We describe a mechanism here for simulating the model following Fewster et al. (in prep). In practice, it is possible that within a single capture occasion, a particular animal is encountered multiple times. For example, when surveying whales from survey boats, one capture occasion might constitute a full day of survey effort, and a "capture" is an approach of a whale sufficiently close to obtain a photograph or

DNA sample. Suppose the number of encounters for animal $i = 1, \ldots, N$ on any occasion is a random variable S_i that follows a Poisson distribution with mean ϕ, i.e., $S_i \sim \text{Poisson}\,(\phi)$. For each encounter of an animal, the probability of getting a photograph is x, and the probability of getting a genetic sample is y. We assume that the two events are independent. For one animal and a single occasion, the probability of obtaining a photograph is $p = 1 - \exp\,(-\phi x)$, the probability of getting a genetic sample is $g = 1 - \exp\,(-\phi y)$, and the probability of getting both samples is $\eta = 1 - \exp\,(-\phi x y)$. We use $\boldsymbol{\theta} = (p, g, \eta)$ to parametrize this model, where η is constrained such that $\eta < p$ and $\eta < g$. The two protocols have substantial overlap if η is large relative to p or g. If η is very small, the two sources are almost independent of each other. Probabilities for the capture codes 0, 1, 2, 3, and 4 within an occasion are $p_0 = (1-p)\,(1-g)\,/\,(1-\eta)\,,\ p_1 = 1 - g - p_0,\ p_2 = 1 - p - p_0,\ p_3 = \eta$, and $p_4 = p + g - \eta - 1 + p_0$.

3.6.2 Simulation Study

Simulation results from the two-source model M_0 in a number of settings are shown in Figures 3.4 and 3.5. For each setting, two-source estimates of N using the saddlepoint method to gain an approximate estimate are shown against estimates obtained using data from each of the two protocols alone, which use straightforward maximisation of the exact likelihood for single-source model M_0. The boxplots in each scenario show estimates of N from 500 simulations.

The figures reveal that the two-source model produces estimates of N with negligible bias and approximately nominal confidence interval coverage for 95% confidence intervals. From simulations in a broad range of other settings not shown here, we always observe similar results. Applying model M_0 to a single source of data also generates almost unbiased estimation with similar confidence interval coverage. Using genetic data only results in a slight positive bias to the results in these simulations, because a low genetic capture probability g was used in all four scenarios. The advantage of applying the two-source model over analysing the two sources of data separately is that it always yields better precision for N, which can be seen from the smaller mean widths of the confidence intervals.

For each of the four simulation studies shown here, applying model M_0 to photographic samples yielded better precision for N than applying the model to genetic samples. This is because we set the photograph capture probability p to be higher than the genetic capture probability g, to test the estimation framework in an asymmetric context.

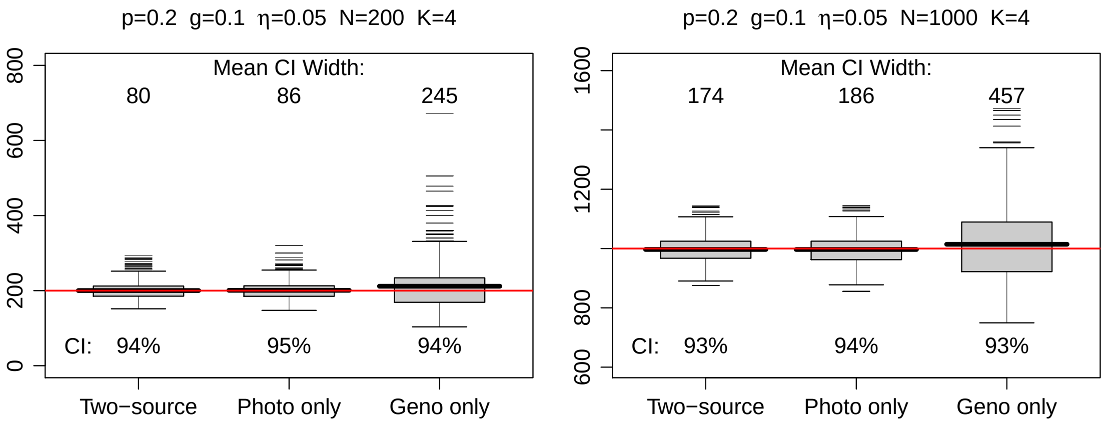


Fig. 3.4 Estimates of N from the two-source model M_0 for the settings $p = 0.2, g = 0.1, \eta = 0.05, K = 4$, and $N = 200$ (left) or $N = 1000$ (right). The red horizontal lines on each plot represent the true values of N. The black lines across each box give the mean values of the 500 estimates. The quantity above and percentage below each box indicate the mean width and coverage of nominal 95% confidence intervals. The three boxes from left to right in each plot show the distributions of estimates of N calculated in three different ways by fitting: the two-source model M_0 to both sources of data using the saddlepoint method to approximate the likelihood function; model M_0 to photographic samples only; and model M_0 to genetic samples only.

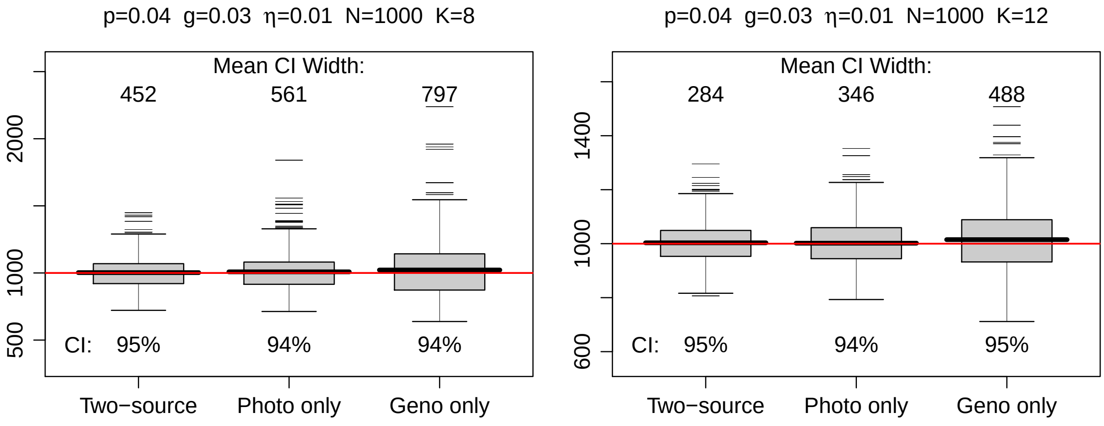


Fig. 3.5 Estimates of N from the two-source model M_0 for the settings $p = 0.04, g = 0.03, \eta = 0.01, N = 1000$, and $K = 8$ (left) or $K = 12$ (right). Other details are the same as those in Fig. 3.4.

3.6.3 Comparison with the Hybrid Approximation

The previous simulation studies showed that the saddlepoint method can produce reasonable estimation results for the two-source model M_0. Here, we further verify the method by comparing it with the hybrid approximation method described in Section 2.4.2. Figures 3.6 and 3.7 serve as two examples of these comparisons.

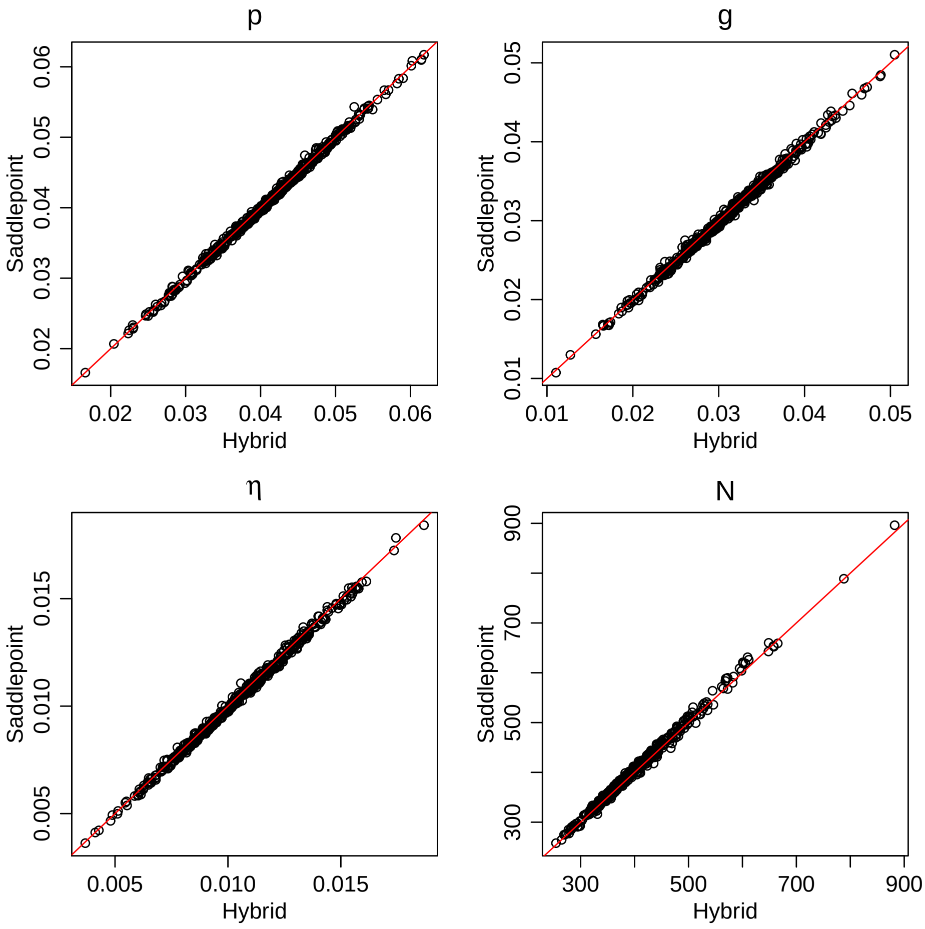


Fig. 3.6 Scatter plots of parameter estimates from 500 simulations calculated using the saddlepoint method and the hybrid approximation method for the setting $p = 0.04$, $g = 0.03$, $\eta = 0.01$, $N = 400$, and $K = 8$. Points distributed on the red lines indicate that estimates from the two approaches are practically identical.

From these two figures, we can see that the two methods consistently yield almost identical estimates for all model parameters in the two scenarios. They also give similar 95% confidence intervals for all parameters. Scatter plots using numerous other settings not presented here followed the same pattern in every case investigated. Since the two methods are based on completely different density approximation techniques, their close accordance presents strong evidence that we can trust both approaches.

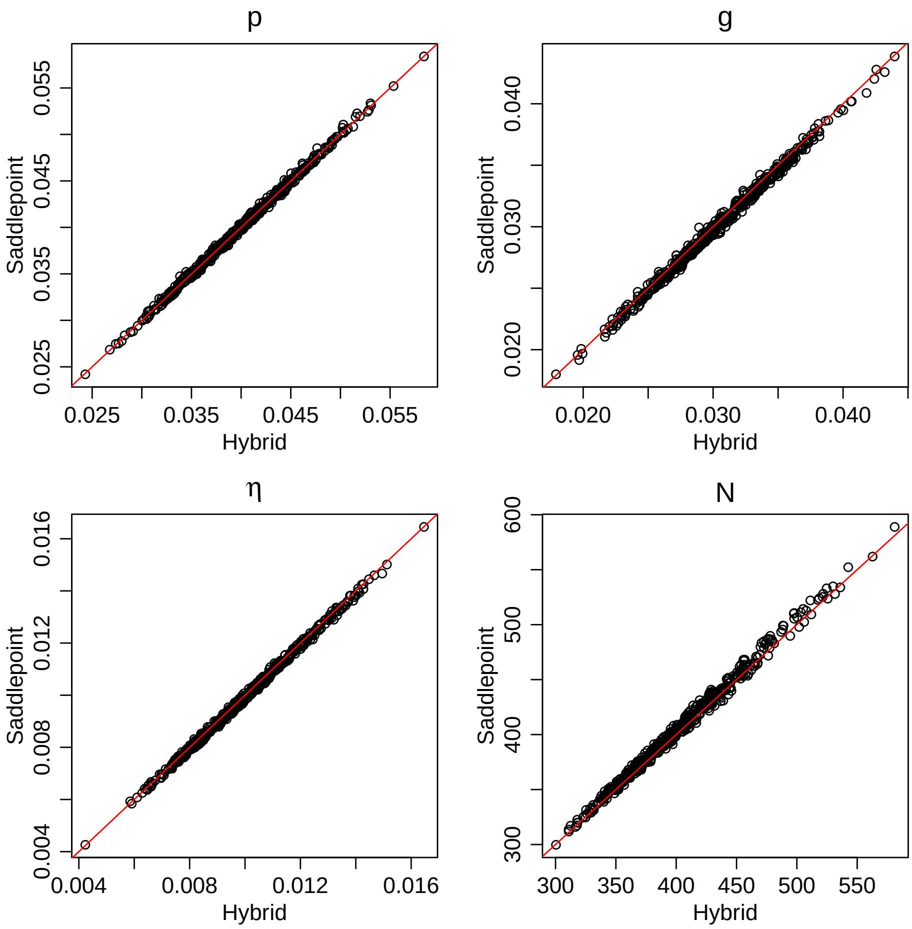


Fig. 3.7 Scatter plots of parameter estimates from 500 simulations calculated using the saddlepoint method and the hybrid approximation method for the setting $p = 0.04$, $g = 0.03$, $\eta = 0.01$, $N = 400$, and $K = 12$. Points distributed on the red lines indicate that estimates from the two approaches are practically identical.

Using the setting of Fig. 3.7, the average fitting time for one data set using the saddlepoint method is around 2 seconds, contrasting with 0.4 seconds required by the hybrid approximation method. The saddlepoint method is slightly slower, but it is still much faster than the Bayesian approach as implemented in `multimark`, which always required over 30 minutes even when $K = 4$ (see Table 2.2). Moreover, the saddlepoint method fits all models in the latent multinomial class, while the hybrid approximation is only suitable for the two-source model. Thus, while there is no exact likelihood available for this model, we have shown that all these approaches give consistent inference with each other, and that the saddlepoint method has the advantage of offering both speed and generality.

3.7 Application to Multi-List Models

3.7.1 Example: Auckland Diabetes Study

We now demonstrate how the saddlepoint method can be used for other models in the literature, for which there is no existing exact-likelihood approach as far as we know. As an example, we consider multi-list data from a study for estimating the prevalence of diabetes in Auckland, New Zealand (see Sutherland, 2003; Huakau, 2002, for more details). This is a four-list study with 1,276 general practitioner records on list G, 1,297 pharmacy records on list P, 12,972 outpatient records on list O, and 3,436 inpatient discharge records on list D. The list structure considered by Sutherland and Schwarz (2005) to analyse these data is shown in Fig. 3.8.

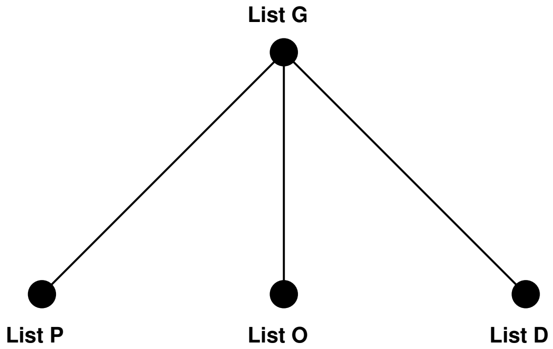


Fig. 3.8 List structure for the Auckland diabetes study data. Lines are drawn between lists that share a common identification tag.

For this list structure, there are $J = 16$ latent capture histories, each of which is presented in the form of $\lambda = (\lambda_G, \lambda_P, \lambda_O, \lambda_D)$, where $\lambda_i = 1$ if the individual is on list

Table 3.2 Estimates and standard errors of all model parameters except for N for the Auckland diabetes study, obtained by the saddlepoint method and the quasi-likelihood approach of Sutherland and Schwarz (2005).

	β_G	β_P	β_O	β_D	β_{GP}	β_{GO}
Sutherland and Schwarz (2005)						
Estimate	−3.76	−3.74	−1.01	−2.95	1.13	0.45
Standard error	0.14	0.14	0.14	0.11	0.10	0.10
Saddlepoint approximation						
Estimate	−3.76	−3.74	−1.00	−2.94	1.13	0.44
Standard error	0.14	0.14	0.14	0.11	0.10	0.10

i and $\lambda_i = 0$ if not, for $i \in \{G, P, O, D\}$. An individual with latent capture history 1011 is on lists G, O and D, but not on list P. The set of observable histories was derived in more detail by Sutherland and Schwarz (2005), and is {01··, 0·1·, 0··1, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111}. The observed data vector for this list structure is $\boldsymbol{y}$ = (1183, 12265, 3276, 654, 51, 366, 91, 40, 4, 5, 14).

Sutherland and Schwarz (2005) selected the Poisson log-linear model $\boldsymbol{\beta}^{(6)} = [GP = OD][GO = GD = PO = PD]$ to describe the latent vector $\boldsymbol{Z}$, because it yielded the lowest QIC$_u$ statistic (Pan, 2001). This model includes all possible first-order interaction effects, but some of them are set to be identical for a more parsimonious model. For example, the notation $[GP = OD]$ means that the interaction effect between lists G and P is set equal to that between lists O and D, i.e. $\beta_{GP} = \beta_{OD}$. Thus the parameter vector for this model is $\boldsymbol{\beta} = (\beta_0, \beta_G, \beta_P, \beta_O, \beta_D, \beta_{GP}, \beta_{GO})$.

The estimate $\widehat{N}$ of the number of diabetes sufferers from the Poisson quasi-likelihood approach of Sutherland and Schwarz (2005) is 45,853. Associated with the estimate are three standard errors, 4530, 4343, and 4008, calculated in different ways (see Sutherland and Schwarz, 2005, for more details). We apply the multinomial saddlepoint method under the same model as described in Section 2.2.5 and obtain a smaller estimate 43,422, with a similar standard error 4303. The two methods yield slightly different estimates of the population size; however, they give almost identical estimates and standard errors for other model parameters that are shown in Table 3.2. This provides some evidence for the validity of the two distinct methods.

Here we investigate the reason why the method of Sutherland and Schwarz (2005) produces a larger estimate of N, while giving similar estimates to the saddlepoint method for the other parameters. In the latent Poisson model of Sutherland and Schwarz (2005), N is not an explicit parameter. It is instead estimated by

$$\widehat{N} = \exp\left(\hat{\beta}_0\right) + \sum_{i=1}^{I} Y_i, \tag{3.33}$$

where $\hat{\beta}_0$ is the estimate of β_0, and $\exp\left(\hat{\beta}_0\right)$ is the estimate of Z_1, the count of the null history 0000. Note that the true parameter N is by definition the sum of all components of the latent vector $\boldsymbol{Z}$:

$$N = \sum_{j=1}^{J} Z_j = Z_1 + \sum_{j=2}^{J} Z_j. \tag{3.34}$$

However, some components of $\boldsymbol{Z}$ are not observable, for example, Z_{0101} and Z_{0000}. Thus, estimating $\sum_{j=2}^{J} Z_j$ using $\sum_{i=1}^{I} Y_i$ as in equation (3.33) is likely to make the estimator (3.33) positively biased, since some components of $\boldsymbol{Z}$ are counted repeatedly in $\sum_{i=1}^{I} Y_i$.

3.7.2 Simulation Study

We conducted simulations to further demonstrate the performance of the saddlepoint method applied to multi-list problems in a wider range of scenarios. We investigated the list structure shown in Fig. 2.1, and explored different types of list dependence following Sutherland and Schwarz (2005).

Firstly, the four lists were assumed to be strictly independent, so we considered models including main effects only. This scenario is unlikely to occur in reality but serves as a validation for the methodology. Secondly, we investigated scenarios with simple list dependence: for example, interactions were fitted between lists 1 and 2 and lists 1 and 3, while all other pairs of lists were independent. For this case, the parameter vector is $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_4, \beta_{12}, \beta_{13})$. Thirdly, scenarios with more complicated list interactions were explored, for example, with dependence between every pair of the four lists. The parameter vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_4, \beta_{12}, \ldots, \beta_{34})$ contains 10 parameters.

Sutherland and Schwarz (2005) indicated that their method gave estimates with negligible bias and high precision in the first two scenarios. In the third scenario, their method produced positively biased estimates. By contrast, we found that the saddlepoint method gave accurate inference in all scenarios we explored.

We present results for one setting of the third scenario to show our main results. Fig. 3.9 shows that for each of the 10 parameters (excluding N), the two methods both generate estimation results with negligible bias and almost the same values for mean confidence interval width and confidence interval coverage for nominal 95% confidence intervals. This means that the two approaches both perform well for estimating these parameters. However, the saddlepoint method also gives estimates for the parameter N with no bias and close to nominal confidence interval coverage for 95% confidence intervals, while Sutherland and Schwarz (2005)'s method is positively biased with very low confidence interval coverage. This could reflect a similar situation to that seen in the Auckland diabetes study.

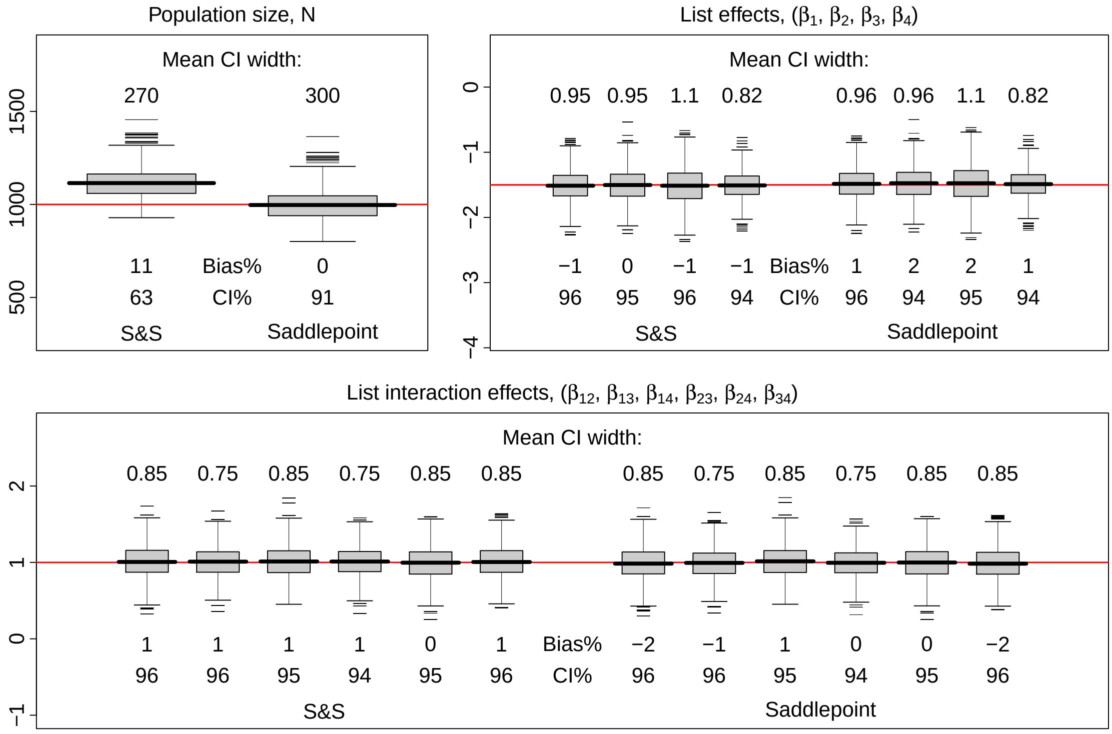


Fig. 3.9 Distributions of parameter estimates from 1000 simulations obtained using the saddlepoint approximation method (right-hand boxplots) and the quasi-likelihood approach of Sutherland and Schwarz (2005). True parameter values are: $N = 1000$, $\beta_1 = \cdots = \beta_4 = -1.5$, and $\beta_{12} = \cdots = \beta_{34} = 1$, and are shown by red horizontal lines across the plots. Black horizontal lines across the boxes give the means of the 1000 estimates. Numbers above each box show the mean width of 95% confidence intervals. Percentages below each box give percentage bias and coverage of 95% confidence intervals.

3.8 Application to Multi-Way Contingency Tables

3.8.1 Example: Czech Autoworkers Data

Finally we showcase the use of the saddlepoint method for drawing inference on parameters underlying multi-way contingency tables when only certain marginal totals are supplied.

We first consider a real data example in the form of a six-way contingency table over six binary variables, each with two categorical levels, as shown in the left-hand panel of Table 3.3. The data come from an epidemiological study involving 1841 workers from a Czechoslovakian car factory, to study potential factors associated with coronary thrombosis (Edwards and Toma, 1985). This table was used by Dobra et al. (2006) to demonstrate their Bayesian approach for inference on model parameters underlying the table, as well as cell entries, given sets of marginal totals.

Assume that the information we are provided with about the original table is a set of marginal totals

$$\boldsymbol{y} = \left(\boldsymbol{y}_{\{A,B,C,D,F\}}, \boldsymbol{y}_{\{A,B,D,E,F\}}, \boldsymbol{y}_{\{A,B,C,E,F\}}\right), \tag{3.35}$$

which consists of three five-way marginal tables. For example, $\boldsymbol{y}_{\{A,B,C,D,F\}}$ sums over the two levels of variable E within each cell, leaving 2^5 cells representing all possible combinations of the remaining factors A, B, C, D, and F.

To describe the original table, we consider a multinomial model incorporating all possible first-order interactions and second-order interactions in addition to six main effects. It is straightforward to see that the marginal vector $\boldsymbol{y}$ in (3.35) specifies all possible one-way, two-way, and three-way marginal tables except for the three-way marginal table $\boldsymbol{y}_{\{C,D,E\}}$, thus the data $\boldsymbol{y}$ are less than sufficient statistics for the model we apply. Dobra et al. (2006) considered more complicated models in their Bayesian framework, but the number of model parameters exceeded the number of independent observations contained in the data they used. For this reason, we do not reproduce their models, so a direct comparison between the two approaches is not available.

Note that likelihood factorization is not needed for applying the saddlepoint method to the LMM for this problem. One reason is that in practice the vector $\boldsymbol{y}$ does not contain any component equal to zero, because such entries would reveal multiple zeros throughout the table and increase the risk of disclosing the raw data, which might compromise participant privacy. Besides, marginal totals are sums of particular components of the full table, so $\boldsymbol{y}$ does not have any component that can be observed directly from the original table. Consequently, we apply the saddlepoint method directly to approximate the density of $\boldsymbol{Y} = T\boldsymbol{Z}$.

We calculate estimates of the model parameters first, and then use them to estimate cell entries of the original table. The 95% confidence intervals for all entries of the table are presented in the right-hand panel of Table 3.3. We only present integer-valued intervals which are practically meaningful, instead of the full-precision real-valued intervals. It can be seen that all the true cell entries lie in the intervals we obtained. Following Dobra et al. (2006), we consider the particular cell $(1, 2, 2, 1, 1, 2)$, whose true entry is one. This cell is marked by a box on the left and right panels of Table 3.3. In general, small cell entries are harder to estimate accurately. Our calculation gives a point estimate of 1.4 with 95% confidence interval $[0, 3]$, which is consistent with the true cell entry.

Table 3.3 Czech autoworkers data from Edwards and Toma (1985) and 95% confidence intervals for all entries calculated by the saddlepoint approximation method under a LMM.

				B	No		Yes		B	No		Yes	
F	E	D	C	A	No	Yes	No	Yes	A	No	Yes	No	Yes
Neg	< 3	< 140	No		44	40	112	67		[26, 70]	[21, 60]	[93, 135]	[54, 90]
			Yes		129	145	12	23		[95, 152]	[118, 170]	[3, 21]	[6, 31]
		≥ 140	No		35	12	80	33		[8, 55]	[0, 28]	[60, 94]	[15, 42]
			Yes		109	67	7	9		[82, 141]	[49, 91]	[3, 18]	[0, 23]
	≥ 3	< 140	No		23	32	70	66		[3, 41]	[9, 49]	[50, 83]	[47, 79]
			Yes		50	80	7	13		[30, 76]	[61, 107]	[0, 17]	[5, 28]
		≥ 140	No		24	25	73	57		[6, 50]	[6, 44]	[58, 91]	[46, 79]
			Yes		51	63	7	16		[23, 72]	[39, 82]	[0, 13]	[1, 23]
Pos	< 3	< 140	No		5	7	21	9		[2, 11]	[2, 11]	[11, 24]	[6, 16]
			Yes		9	17	$\boxed{1}$	4		[5, 15]	[10, 24]	$\boxed{[0, 3]}$	[0, 5]
		≥ 140	No		4	3	11	8		[1, 6]	[0, 5]	[8, 20]	[2, 11]
			Yes		14	17	5	2		[8, 20]	[10, 23]	[1, 5]	[0, 9]
	≥ 3	< 140	No		7	3	14	14		[0, 8]	[1, 9]	[10, 24]	[7, 16]
			Yes		9	16	2	3		[4, 14]	[9, 21]	[0, 5]	[1, 7]
		≥ 140	No		4	0	13	11		[0, 5]	[0, 5]	[7, 19]	[5, 15]
			Yes		5	14	4	4		[3, 11]	[7, 18]	[0, 5]	[1, 9]

A: smoker; B: strenuous mental work; C: strenuous physical work; D: systolic blood presure; E: ratio of β and α lipoproteins; F: family anamnesis of coronary heart disease. Neg is short for negative, and Pos is short for positive.

3.8.2 Simulation Study

For a simulation example, we consider a four-way contingency table over four binary variables $\xi_1, \ldots, \xi_4$. The model underlying the table incorporates 10 parameters, including four main effects and six first-order interaction effects. Suppose we are provided with marginal totals that are less than sufficient statistics for the model, for example, $\boldsymbol{y} = \left(\boldsymbol{y}_{\{\xi_1,\xi_2,\xi_3\}}, \boldsymbol{y}_{\{\xi_2,\xi_3,\xi_4\}}\right)$ that omit the two-way marginal table $\boldsymbol{y}_{\{\xi_1,\xi_4\}}$.

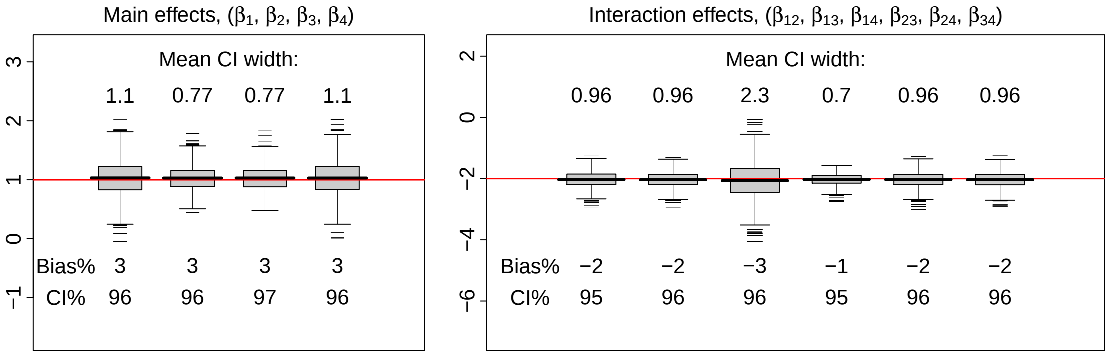


Fig. 3.10 Distributions of parameter estimates obtained using the saddlepoint method for a four-way contingency table with true parameter values: $N = 5000$, $\beta_1 = \cdots = \beta_4 = 1$, and $\beta_{12} = \cdots = \beta_{34} = -2$, which are shown by red horizontal lines across the plots. Black horizontal lines across the boxes indicate the means of the 1000 estimates. The quantity above each box shows the mean width of 95% confidence intervals. Percentages below each box show percentage bias and 95% confidence interval coverage respectively.

Distributions of parameter estimates are shown in Fig. 3.10. The saddlepoint method produces roughly unbiased estimation with approximately nominal confidence interval coverage for 95% confidence intervals for all model parameters. Note that the mean width of the confidence intervals for the parameter β_{14} is much higher than that for the other parameters. This is because the marginal table $\boldsymbol{y}_{\{\xi_1,\xi_4\}}$ is not provided so the data are less informative for this parameter. Moreover, estimation results for the parameters related to variables ξ_1 and ξ_4 are also influenced. For example, the mean width of confidence intervals for the parameter β_{23} is the smallest, and confidence intervals for parameters β_2 and β_3 are also narrower than those for β_1 and β_4 on average.

We further estimated cell entries of the table using the parameter estimates obtained. Consider the first cell entry Z_{1111} of the table as an example. Average percentage bias and 95% confidence interval coverage for this entry were -1.02% and 93% respectively.

3.9 Conclusions and Closing Remarks

We have developed a novel approximate likelihood approach based on the saddlepoint method to address the problem of parameter estimation for LMMs. We have demonstrated the validity of the method by comparing it with the exact likelihood formulation for model $M_{t,\alpha}$ and the hybrid approximation method for the two-source model, which itself was validated against the Bayesian approach in Chapter 2. We further applied the method to two additional LMMs to investigate inference for multi-list studies using incomplete list matching, and inference for multi-way contingency tables given known marginals, which have previously been modelled by latent Poisson models.

Note that in the real example and simulation study for multi-way contingency tables, we always assume that less than sufficient statistics are provided. This is because when sufficient statistics are available LMMs are not needed. Instead, a multinomial model can be fitted. However, this provides another way to verify the saddlepoint method. We found that if a set of marginal totals were provided that are sufficient statistics for the model underlying the table, then fitting a LMM to the marginals gave identical parameter estimates and confidence intervals to those obtained by fitting a multinomial model to the full table. This is reasonable, because sufficient statistics contain full information on the raw data. We do not present the results here, but this observation provides further evidence for the validity of the saddlepoint approximation to LMMs for statistical inference.

Our two key motivations for developing the saddlepoint method were generality and computational speed. The saddlepoint method is considerably faster than the Bayesian approaches for LMMs (e.g., McClintock, 2015; Bonner and Holmberg, 2013; McClintock et al., 2013; Link et al., 2010), and is more general than other alternatives (e.g., Fewster et al., in prep; Vale et al., 2014; Sutherland and Schwarz, 2005). Most inference results shown in this chapter are based on simulations. Two real examples, the Auckland diabetes study and the Czechoslovakian workers data, show that the proposed method can easily be applied for real data analysis.

The `TMB` package is vital for the computations involved in applying the saddlepoint approximation method for maximum likelihood estimation. Motivated by the similarities between the equations of the Laplace and saddlepoint approximations, we illustrated how to make minor changes to the source code of `TMB` so that it can be used conveniently for saddlepoint-based estimation. We considered a saddlepoint approximation to the probability mass function of LMMs in this thesis; however, the same modification of the `TMB` code could be applied to saddlepoint approximations for any other models.

In the contexts of multi-list studies and multi-way contingency tables, many models can be applied to describe the latent vector $\boldsymbol{Z}$, as long as the number of model parameters is no larger than the number of independent components of $\boldsymbol{Y}$. Questions of model selection arise in this situation. For example, Sutherland and Schwarz (2005) selected a model including seven parameters for analysing the Auckland diabetes data using the QIC_u statistic. Many other criteria for model selection are well developed such as the Akaike information criterion (Akaike, 1974) and Bayesian information criterion (Burnham and Anderson, 2003). Our investigation of the saddlepoint likelihood for model $M_{t,\alpha}$, compared against the exact likelihood for this model, suggests that the saddlepoint likelihood may be suitable as a basis for the usual suite of likelihood-based model selection tools when most components of the observed vector $\boldsymbol{y}$ are sufficiently large. The saddlepoint likelihood function might otherwise need to be normalized if $\boldsymbol{y}$ contains many small components. See Section 3.1 of Butler (2007) for more details about the normalization of saddlepoint approximations. Full investigation of this is beyond the scope of this thesis.

The framework we have proposed for saddlepoint-based estimation may be suitable for a much wider class of models, where the latent variable $\boldsymbol{Z}$ need not be multinomial, as long as its moment generating function is efficiently computable. For example, $\boldsymbol{Z}$ might follow a multivariate Poisson distribution (Dobra et al., 2006; Sutherland and Schwarz, 2005; Lee, 2002). However, each different model class implies a different saddlepoint approximation, which may have different properties from our LMM case, so new approximations should be tested extensively by simulation, as we have done for the multinomial case. It is particularly helpful if there is a specific model for which the exact likelihood function is available for comparsion.

Part II

Inference for Two-Locus Linkage Models in Population Genetics

4

Preliminaries: Models and Methods

4.1 Overview

In this chapter, we start by giving a brief introduction to linkage disequilibrium. Then we describe the classical Wright-Fisher model in population genetics, and a generalisation of it, namely the two-locus diallelic model that incorporates mutation and recombination. In the following section, we introduce the diffusion approximation, which is useful for studying genetic models in large populations. In the second half of this chapter, we provide a review of Song and Song (2007)'s method to compute the expectation of r^2 at stationarity under the diffusion approximation. Finally, we discuss the maximum entropy principle, and some related issues, including numerical solutions to maximum entropy problems and trust region optimisation methods. The main purpose of this chapter is to present a review of models and methods related to constructing the stationary distribution of r^2.

4.2 Linkage Disequilibrium

Linkage disequilibrium (LD) refers to the non-random combination of alleles at two or more loci in population genetics (Lewontin, 1964; Lewontin and Kojima, 1960). It can be used for many purposes, for example, understanding the evolutionary history of a species (e.g., Slatkin, 2008; Tishkoff et al., 1996), gene mapping in association studies (e.g., Horikawa et al., 2000; Pritchard and Rosenberg, 1999), and detecting recombination hotspots (e.g., Auton et al., 2014; Li and Stephens, 2003).

A range of statistics are defined for measuring LD in the literature (see Pritchard and Przeworski, 2001; Jorde, 2000, for a review). The suitablity of each definition depends on the context of specific problems. A popular and convenient measure of LD is the squared correlation coefficient r^2, which has been studied and applied in many papers (e.g., Gupta et al., 2005; Mueller, 2004; Hill and Robertson, 1968).

Consider two biallelic loci with allele types A_1 or A_2 present at the first locus, and B_1 or B_2 at the second locus. Let p_1, p_2, and p_3 denote the frequencies of gametes A_1B_1, A_1B_2, and A_2B_1. The LD measure r^2 defined in terms of $\boldsymbol{p} = (p_1, p_2, p_3)$ is

$$r^2 = \frac{D^2}{p(1-p)q(1-q)}, \tag{4.1}$$

where

$$\begin{aligned} p &= p_1 + p_2 \\ q &= p_1 + p_3 \\ D &= p_1 - pq = p_1 - (p_1 + p_2)(p_1 + p_3). \end{aligned} \tag{4.2}$$

Thus, D measures the difference between the actual frequency of gamete-type A_1B_1, and the frequency that would be obtained if the two loci were independent. The LD measure r^2 is then the square of the correlation coefficient between the indicator random variables I_{A_1} and I_{B_1}:

$$r^2 = \operatorname{corr}^2\left(I_{A_1}, I_{B_1}\right) = \frac{\operatorname{cov}^2\left(I_{A_1}, I_{B_1}\right)}{\operatorname{var}\left(I_{A_1}\right)\operatorname{var}\left(I_{B_1}\right)}. \tag{4.3}$$

The distribution of r^2 is influenced by many evolutionary factors, such as selection, mutation, and recombination (Pritchard and Przeworski, 2001). Every pair of loci at a particular generation time in a finite reproducing population generates a single realisation of r^2. The distribution of these r^2 values across locus pairs encapsulates information on the evolutionary factors acting on the population. Since it is possible to generate sample observations of r^2 at equilibrium, by observing sample correlations between alleles at multiple locus pairs in a single generation, it is plausible that we could estimate those evolutionary factors (parameters) from the data using maximum likelihood. To accomplish this we need to find the probability distribution of r^2 at stationarity. This is a complex function of the parameters, which to our knowledge has not been characterised before. So far, most research emphasis has focused on exploring the expectation of r^2 (Song and Song, 2007; Ohta and Kimura, 1969b; Hill and Robertson, 1968) or its empirical distributions under different models (Hudson, 1985; Golding, 1984). As far as we know, no method has been proposed to obtain the density function of r^2 at stationarity.

In this project, our focus is to approximate the stationary PDF of r^2, and then use it for estimating evolutionary parameters such as mutation rate and recombination rate from sample observations of r^2. Models and methods related to this goal will be introduced in this chapter.

4.3 Models and Notation

4.3.1 Wright-Fisher Model

The Wright-Fisher model (Fisher, 1999; Wright, 1931) is an idealised stochastic model to study genetic drift in population genetics. Genetic drift is a basic mechanism of evolution, in which relative frequencies of different alleles or gametes for a population vary from one generation to the next.

The original Wright-Fisher model is based on a series of ideal assumptions for the population of interest: (1) individuals in the population are monoecious and diploid; (2) mating between individuals is assumed to be random; (3) population size is assumed to be a finite constant for all generations, which are discrete and non-overlapping; (4) individuals are able to produce a large number of gametes, each of which is equally likely to be inherited to the next generation; and (5) other evolutionary factors including selection, mutation, and recombination are not considered.

We first give an explanation of the terminology used in these assumptions. Individuals are said to be *diploid* if they have two copies of each chromosome, one from each parent. A *monoecious* individual has both male and female reproductive organs, so that reproduction can be completed within one individual or between two individuals. A *gamete* is a reproductive cell, such as a sperm or egg, that contains a single set of chromosomes. *Selection* refers to a genetic process by which specific alleles or genotypes are favoured over alternatives for reproductive success. Mutation and recombination will be explained in the following section.

Consider a particular locus that possesses two possible alleles A and a. In genetics, one locus simply represents a position on a chromosome. Define X_t to be the number of copies of the allele A in generation $t = 1, 2, 3, \ldots$. For a diploid population of N individuals, there are $2N$ gametes in total in each generation. It follows that X_t takes values from $\{0, 1, \ldots, 2N\}$. Note that although a large number of gametes are produced, the model assumes that only $2N$ of them will be inherited to the next generation. The $2N$ gametes are sampled randomly and independently from the gamete pool.

Under the Wright-Fisher model, X_{t+1} follows a binomial distribution:

$$X_{t+1} \sim \text{Bin}\left(2N; p_{t+1}\right), \tag{4.4}$$

where $p_{t+1} = X_t/2N$ denotes the expected proportion of A in generation $t+1$ and is equal to the actual proportion of allele A in generation t. The one-step transition probability of going from $X_t = i$ to $X_{t+1} = j$ for $i, j \in \{0, 1, \ldots, 2N\}$ is

$$p_{ij} = \mathbb{P}\left(X_{t+1} = j \mid X_t = i\right) = \binom{2N}{j} p_{t+1}^{j} (1 - p_{t+1})^{2N-j}. \tag{4.5}$$

It is clear that the stochastic process defined by the Wright-Fisher model is a discrete-time, discrete-state Markov chain, because the distribution of X_{t+1} is only determined by the current state X_t without any reference to previous states.

For large t, the Wright-Fisher model is guaranteed to reach fixation, as there are two absorbing states for the Markov chain, namely 0 and $2N$, for which $p_{ii} = 1$ and $p_{ij} = 0$ for $i \neq j$ when $i = 0$ or $2N$. This means that once an allele is lost in some generation, it never returns. Thus the population will consist entirely of the alternative allele and will not show any further genetic variation in future generations.

4.3.2 Two-Locus Diallelic Model

For a more realistic model that does not guarantee fixation in the long run, we consider a generalisation of the original Wright-Fisher model, namely the two-locus diallelic (TLD) model that incorporates mutation and recombination. More details on this model, along with other generalisations of the Wright-Fisher model, are given by Liu (2012) and references therein.

In the TLD model, we consider two genetic loci, each of which possesses two possible alleles. Let A_1, A_2 denote the two possible alleles for locus 1, and let B_1, B_2 denote the two possible alleles for locus 2. There are four possible types of gamete, A_1B_1, A_1B_2, A_2B_1, and A_2B_2, whose counts in generation t are X_{t1}, X_{t2}, X_{t3}, and X_{t4}, where $X_{t1} + X_{t2} + X_{t3} + X_{t4} = 2N$. Since X_{t4} can be derived using $2N$ to deduce the other three counts, we use $\boldsymbol{X}_t = (X_{t1}, X_{t2}, X_{t3})$ to denote the Markov chain specified by the TLD model. Note that if gametic frequencies are of interest, we may equivalently use $\boldsymbol{X}_t/2N$ to define the TLD model. We do not distinguish the two definitions in this thesis.

The *genotype* of a diploid individual consists of two gamete-types, one on each chromosome, so it is of the form A_iB_j/A_mB_n, where $m, n, i, j \in \{1, 2\}$. It follows that there are 10 possible genotypes for the TLD model. *Recombination* refers to a process in which two gametes exchange their alleles during meiosis. For example, two gametes A_1B_1 and A_2B_2 generally lead to genotype A_1B_1/A_2B_2; however, in a recombination event the positions of alleles B_1 and B_2 might be exchanged so that genotype A_1B_2/A_2B_1 is generated. We define the recombination rate C between these two loci to be the probability that an odd number of exchanges occurs. It is clear that the parameter C varies between 0 and 0.5. When the two loci are very close together, it is very unlikely that a recombination event will take place between them, so the parameter C will be close to 0. For distant loci, C may approach 0.5.

In addition to recombination, endogenous and environmental factors may cause alleles to change from one type to another. This process is called *mutation.* In the TLD model, the same mutation rate μ is assumed for the alleles in both loci, that is

$$A_1 \underset{\mu}{\overset{\mu}{\rightleftharpoons}} A_2 \qquad \text{and} \qquad B_1 \underset{\mu}{\overset{\mu}{\rightleftharpoons}} B_2. \tag{4.6}$$

Under the original Wright-Fisher model with independent loci, the expected composition of the four types of gametes in the next generation is only determined by the current composition. In the TLD model, the mutation rate μ and recombination rate C also influence the expected counts of the four gamete-types in the next generation. The one-step transition probability for this model was derived in detail by Liu (2012). We only present the results here.

Suppose two states $\boldsymbol{x} = (x_1, x_2, x_3)$ and $\boldsymbol{y} = (y_1, y_2, y_3)$ are taken arbitrarily from the state space of the model, i.e.,

$$\mathcal{S} = \left\{(s_1, s_2, s_3) \in \mathbb{N}^3 \mid s_1 + s_2 + s_3 \leq 2N\right\}. \tag{4.7}$$

Assume $x_4 = 2N - x_1 - x_2 - x_3$ and $y_4 = 2N - y_1 - y_2 - y_3$. Then the one-step transition probability of going from $\boldsymbol{x}$ to $\boldsymbol{y}$ is

$$\begin{aligned} p_{\boldsymbol{xy}} &= \mathbb{P}\left(\boldsymbol{X}_{t+1} = \boldsymbol{y} \mid \boldsymbol{X}_t = \boldsymbol{x}\right) \\ &= \frac{(2N)!}{y_1!y_2!y_3!y_4!}\Phi_1^{y_1}\Phi_2^{y_2}\Phi_3^{y_3}\left(1 - \sum_{i=1}^{3}\Phi_i\right)^{y_4}, \end{aligned} \tag{4.8}$$

where

$$\begin{aligned} \Phi_1 &= \frac{x_1}{2N}(1-\mu)^2 + \left(\frac{x_2}{2N} + \frac{x_3}{2N}\right)\mu\left(1-\mu\right) + \frac{x_4}{2N}\mu^2 - CD\left(1-2\mu\right)^2 \\ \Phi_2 &= \frac{x_2}{2N}(1-\mu)^2 + \left(\frac{x_1}{2N} + \frac{x_4}{2N}\right)\mu\left(1-\mu\right) + \frac{x_3}{2N}\mu^2 + CD\left(1-2\mu\right)^2 \\ \Phi_3 &= \frac{x_3}{2N}(1-\mu)^2 + \left(\frac{x_1}{2N} + \frac{x_4}{2N}\right)\mu\left(1-\mu\right) + \frac{x_2}{2N}\mu^2 + CD\left(1-2\mu\right)^2 \end{aligned} \tag{4.9}$$

and

$$D = \frac{x_1}{2N}\frac{x_4}{2N} - \frac{x_2}{2N}\frac{x_3}{2N}. \tag{4.10}$$

The Markov chain defined by the TLD model is irreducible and aperiodic, thus a unique stationary distribution $\boldsymbol{\pi}$ exists, which can be obtained by solving the following equation:

$$\boldsymbol{\pi}^T\left(\Sigma - I\right) = \mathbf{0}, \tag{4.11}$$

where I is an identity matrix and Σ is the transition matrix of the Markov chain that consists of all transition probabilities of the form (4.8). Once the stationary distribution is known for the state space $\mathcal{S}$ consisting of vectors of gametic counts $\boldsymbol{x}$, the stationary distribution of gametic frequencies $\boldsymbol{p} = (p_1, p_2, p_3) = \boldsymbol{x}/2N$ follows immediately. Then the distribution of r^2 at stationarity can be constructed, where r^2 is defined in equations (4.1) and (4.2).

Solving the stationary distribution of the TLD model analytically is infeasible except for trivial cases. Instead, it is common to solve equation (4.11) numerically. However, even for a small population size, say $N = 10$, finding the stationary distribution numerically generates a huge computational burden. This is due to the fact that even a small population size generates a large state space $\mathcal{S}$, so the matrix Σ is of high dimension. As N increases, numerical methods rapidly lose both efficiency and accuracy. An alternative normal approximation provides a possible solution to the problem in some cases when N is very large; however, the performance of this method is not stable and might change from case to case. The latter two approaches to the problem were discussed by Liu (2012), including several examples.

4.4 Diffusion Approximation

To facilitate computation of the stationary distribution of r^2 under the TLD model for large populations, we use the diffusion approximation, which is a well-established technique in population genetics. For a review of the approach, refer to Kimura (1964) and Watterson (1996). The application of the diffusion approximation has a long history in this field and can be traced back to Fisher et al. (1922) and Wright (1931).

The diffusion approximation is a mathematical technique that approximates a discrete stochastic process using a diffusion process which is continuous in both space and time. A diffusion process is a strong Markov process with a continous sample path

(Glynn, 1990). A classical example of a diffusion process is Brownian motion (Doob, 1942). Diffusion processes have many convenient properties that discrete stochastic processes do not possess. The underlying idea of the diffusion approximation is similar to that of the central limit theorem which approximates an intractable sum of random variables by a proper normal random variable under some conditions.

Models for genetic drift in population genetics are usually discrete Markov chains. By diffusion theory, we can reformulate the time space and state space of the model, to ensure that the gap between two successive time points or states is infinitesimal when the population size N is sufficiently large. This can be achieved by establishing links between the time and state units and the population size. For instance, we can scale the discrete state space $\{0, 1, \ldots, 2N\}$ by a factor of $(2N)^{-1}$, so that the difference between two successive states is $(2N)^{-1}$, which tends to 0 as N goes to infinity. Likewise, we also scale the time space by the factor of $(2N)^{-1}$, so that the scaled time space is $\{0, \delta t, 2\delta t, \ldots\}$, where $\delta t = (2N)^{-1}$. In this case, the discrete process can be regarded as a continuous one.

Associated with a diffusion process is typically a Kolmogorov equation (a partial differential equation), and a diffusion generator. The solution of the Kolmogorov equation is the stationary distribution of the diffusion process, but the equation is notoriously difficult to solve. We will discuss this further in Chapter 6. Details on the derivation of the diffusion approximation to the TLD model can be found in established literature, for example, Kimura (1964), Ewens (2004), and Liu (2012). In the remainder of this section, we focus on the diffusion generator of the TLD model, which is particularly valuable for this project.

Ohta and Kimura (1969a,b) first showed that computing certain expectations at stationarity can be greatly facilitated by the diffusion approximation for models in population genetics. Song and Song (2007) extended the method of Ohta and Kimura (1969b) to compute the expectation of r^2 at stationarity. The diffusion generator used by Song and Song (2007) differs from that used by Ohta and Kimura (1969b) by a factor of 2. In this thesis, we follow the formulation of Song and Song (2007).

Let $\boldsymbol{X}_t = (p, q, D)_t$ denote the diffusion process corresponding to the TLD model. Note that using parameters p, q, D as defined in equation (4.2) instead of the original gametic frequencies p_1, p_2, p_3 is more convenient for the calculation of certain expectations. The diffusion generator for the TLD model is

$$\begin{aligned}\mathcal{L} \;=\; & \frac{1}{2}p\left(1-p\right)\frac{\partial^2}{\partial p^2} + \frac{1}{2}\left\{p\left(1-p\right)q\left(1-q\right) + D\left(1-2p\right)\left(1-2q\right) - D^2\right\}\frac{\partial^2}{\partial D^2} \\ & + \frac{1}{2}q\left(1-q\right)\frac{\partial^2}{\partial q^2} + D\frac{\partial^2}{\partial p\partial q} + D\left(1-2p\right)\frac{\partial^2}{\partial p\partial D} + D\left(1-2q\right)\frac{\partial^2}{\partial q\partial D} \\ & + \frac{\theta}{4}\left(1-2p\right)\frac{\partial}{\partial p} + \frac{\theta}{4}\left(1-2q\right)\frac{\partial}{\partial q} - D\left(1+\frac{\rho}{2}+\theta\right)\frac{\partial}{\partial D}\end{aligned} \tag{4.12}$$

where $\rho = 4NC$ and $\theta = 8N\mu$ denote the population-scaled recombination rate and mutation rate. Here, N should be interpreted as the genetic effective population size, also called N_e.

The diffusion generator $\mathcal{L}$ is defined such that

$$\frac{\partial}{\partial t}\mathbb{E}\left\{f\left(\boldsymbol{X}_t\right)\right\} = \mathbb{E}\left\{\mathcal{L}f\left(\boldsymbol{X}_t\right)\right\}, \tag{4.13}$$

for any twice continuously differentiable function f that has compact support. Note that at stationarity, $\mathbb{E}\left\{f\left(\boldsymbol{X}_t\right)\right\}$ does not depend on the time parameter by definition, so its rate of change on the left-hand side of (4.13) is zero. Thus, for any suitable function f, and for $\boldsymbol{X}_t = (p, q, D)_t$ distributed over the continuous state space according to the stationary distribution $\boldsymbol{\pi}$, we have

$$\mathbb{E}\left\{\mathcal{L}f\left(\boldsymbol{X}_t\right)\right\} = 0. \tag{4.14}$$

Equation (4.14) is called the master equation, and it provides considerable power for computing expectations of certain forms at stationarity for the TLD model.

4.5 Analytic Computation of the Expectation of r^2

In this section, we introduce the method of Song and Song (2007) for computing the expectation of r^2 at stationarity under the TLD model. Recall from equation (4.1) that

$$r^2 = \frac{D^2}{p\,(1-p)\,q\,(1-q)}. \tag{4.15}$$

The main strategy of Song and Song (2007)'s method is to write r^2 as an infinite sum of monomials in terms of (p, q, D), for which stationary expectations can be computed using the master equation (4.14). We will extend this method in Chapter 5 to compute higher-order stationary moments of r^2, and thereby construct the density of r^2 using the maximum entropy principle that we will introduce in Section 4.6. Here, we describe Song and Song (2007)'s work only.

Since the infinite series $\sum_{k=0}^{\infty} y^k$ converges to $1/(1-y)$ for $0 < y < 1$, we have the following results for $0 < p, q < 1$:

$$\begin{aligned}\frac{1}{1-p} &= \sum_{k=0}^{\infty} p^k \\ \frac{1}{p} &= \sum_{k=0}^{\infty} (1-p)^k \\ \frac{1}{1-q} &= \sum_{k=0}^{\infty} q^k \\ \frac{1}{q} &= \sum_{k=0}^{\infty} (1-q)^k\,.\end{aligned} \tag{4.16}$$

It follows that

$$\begin{aligned} r^2 &= \frac{D^2}{p\,(1-p)\,q\,(1-q)} \\ &= D^2 \left(\frac{1}{p} + \frac{1}{1-p}\right)\left(\frac{1}{q} + \frac{1}{1-q}\right) \\ &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \Big\{ D^2 p^m q^n + D^2 (1-p)^m q^n \\ &\qquad\qquad + D^2 p^m (1-q)^n + D^2 (1-p)^m (1-q)^n \Big\}. \end{aligned} \tag{4.17}$$

Because the TLD model treats all alleles A_1, A_2, B_1, B_2 identically with respect to mutation, recombination, and drift, we must have:

$$\begin{aligned} \mathbb{E}\left(D^2 p^m q^n\right) &= \mathbb{E}\left\{D^2\,(1-p)^m\,q^n\right\} \\ &= \mathbb{E}\left\{D^2 p^m\,(1-q)^n\right\} \\ &= \mathbb{E}\left\{D^2\,(1-p)^m\,(1-q)^n\right\}. \end{aligned} \tag{4.18}$$

Thus it follows that

$$\mathbb{E}\left(r^2\right) = 4 \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \mathbb{E}\left(D^2 p^m q^n\right). \tag{4.19}$$

The problem of calculating $\mathbb{E}\left(r^2\right)$ therefore reduces to calculating $\mathbb{E}\left(D^2 p^m q^n\right)$ for all pairs of non-negative integers m and n.

Before introducing Song and Song (2007)'s method of computing $\mathbb{E}\left(D^2 p^m q^n\right)$, we first use some examples to illustrate the procedure of calculating expectations using the master equation.

If $f\,(p, q, D) = D$ is used in the master equation (4.14), we obtain

$$\mathcal{L} f\,(p, q, D) = -D\left(1 + \frac{\rho}{2} + \theta\right), \tag{4.20}$$

and therefore (4.14) gives $\mathbb{E}\left(D\right) = 0$ at stationarity. If $f\left(p, q, D\right) = p^n$ is used, we obtain

$$\mathbb{E}\left(p^n\right) = \left(\frac{\theta/2 + n - 1}{\theta + n - 1}\right) \mathbb{E}\left(p^{n-1}\right). \tag{4.21}$$

Since $\mathbb{E}\left(p^0\right) = \mathbb{E}\left(1\right) = 1$, it follows that

$$\mathbb{E}\left(p^n\right) = \frac{\theta/2\left(\theta/2 + 1\right)\cdots\left(\theta/2 + n - 1\right)}{\theta\left(\theta + 1\right)\cdots\left(\theta + n - 1\right)}. \tag{4.22}$$

The same formula also applies to $\mathbb{E}\left(q^n\right)$. Similarly, inserting different forms of $f\left(p, q, D\right)$ such as pq, p^2q, pq^2, Dp^n and Dq^n into the master equation gives the expectations of these items. See Song and Song (2007) for more details. These preliminary expectations are needed in the algorithm below.

Computing $\mathbb{E}\left(D^2p^mq^n\right)$ for all combinations of m and n requires more work. Since $\mathbb{E}\left(D^2p^mq^n\right) = \mathbb{E}\left(D^2p^nq^m\right)$ for the TLD model, we focus on expectations with $m \geq n$. For each pair of (m, n) where $m \geq n$, inserting $f\left(p, q, D\right) = D^k p^{m+2-k} q^{n+2-k}$ into the master equation with $k = 0, 1, \ldots, n+2$ generates a system of $n+3$ linear equations. This system includes more than $n+3$ unknown expectations, so it does not have a unique solution unless the computations are carried out in a particular order. Song and Song (2007)'s innovative idea was to specify an order of computation such that the number of unknown expectations is reduced to $n+3$ at every iterative step.

According to Song and Song (2007)'s algorithm, the computations are executed along an increasing level of $\ell = m + n$, while within each level the algorithm follows an increasing order of n. The computation order is as follows:

$$(m, n) : \overbrace{(0,0)}^{\ell=0} \rightarrow \overbrace{(1,0)}^{\ell=1} \rightarrow \overbrace{(2,0) \rightarrow (1,1)}^{\ell=2} \rightarrow \overbrace{(3,0) \rightarrow (2,1)}^{\ell=3} \rightarrow \overbrace{(4,0) \rightarrow (3,1)}^{\ell=4} \cdots. \tag{4.23}$$

When this order is followed, for a system of equations generated with a specific pair (m, n), some expectations have already been derived in the preliminary exercise above or can be obtained from the solutions of previous equation systems earlier in the ordering. Song and Song (2007) showed that each system of linear equations then includes exactly $n+3$ unknown expectations:

$$\mathbb{E}\left(p^{m+2}q^{n+2}\right),\ \mathbb{E}\left(Dp^{m+1}q^{n+1}\right),\ \boxed{\mathbb{E}\left(D^2p^mq^n\right)},\ \ldots,\ \mathbb{E}\left(D^{2+n}p^{m-n}\right), \tag{4.24}$$

one of which is the expectation $\mathbb{E}\left(D^2p^mq^n\right)$ needed for the computation of $\mathbb{E}\left(r^2\right)$ in (4.19). As byproducts, other expectations are found that are not needed for computing $\mathbb{E}\left(r^2\right)$, but we find later that these can be used to compute higher-order moments of r^2. We will demonstrate this in Chapter 5.

In practice, we need to truncate the infinite sum (4.19) at some finite value to compute $\mathbb{E}\left(r^2\right)$. Given a value of $\ell = m+n$, the truncated summation

$$\mathbb{E}\left(r^2\right)_\ell = 4\sum_{m,n\geq 0}^{m+n=\ell}\mathbb{E}\left(D^2p^mq^n\right) \tag{4.25}$$

serves as an approximation to $\mathbb{E}\left(r^2\right)$. As ℓ varies from 0 to ∞, we obtain a monotonically increasing sequence of partial sums $\left\{\mathbb{E}\left(r^2\right)_\ell\right\}_{\ell=0}^{\infty}$. Since $\mathbb{E}\left(r^2\right)$ is bounded, the convergence of the series is guaranteed. Practical results show that the sequence converges quite fast as $\ell = m+n$ increases, although the rate of convergence tends to be faster for smaller ρ and larger θ. Song and Song (2007) showed that for all settings of ρ and θ, $\mathbb{E}\left(r^2\right)_{\ell_{\max}}$ serves as a good approximation to $\mathbb{E}\left(r^2\right)$ when the truncation level is $\ell_{\max} = 700$.

Song and Song (2007)'s computation indicates that $\mathbb{E}\left(r^2\right)$ is a function of ρ and θ for the TLD model. It can also be readily seen that the stationary PDF of r^2 is a function of the same two parameters, although the function is complicated and does not have an analytic formula or any approximations so far. Throughout, we maintain the definitions of ρ and θ as the population-scaled recombination and mutation rates under the TLD model: $\rho = 4NC$ and $\theta = 8N\mu$.

4.6 Maximum Entropy Principle

In the previous section, we showed that $\mathbb{E}\left(r^2\right)$ can be calculated using Song and Song (2007)'s method. However, our interest is not just in $\mathbb{E}\left(r^2\right)$, but in the entire stationary distribution of r^2. We therefore wish to obtain an approximate density function $\pi\left(r^2\right)$. Here, we describe the maximum entropy (Maxent) principle, by which density functions can be constructed based on known expectations and higher moments.

The Maxent principle was first proposed by Jaynes (1957a,b), who specified the correspondence between information theory and statistical mechanics. It is a powerful tool for approximating the probability density or mass function of an unknown distribution given limited information about it, for example a finite sequence of moments of the distribution. The Maxent principle has gained a large number of applications in many fields, such as ecology (Banavar et al., 2010; Phillips et al., 2006), linguistics (Berger et al., 1996), econometrics (Wu, 2003; Zellner and Highfield, 1988), physics (Robert, 1991), and population genetics (Liu, 2012).

The term "entropy" has two related definitions in both thermodynamics and statistical mechanics. There are many ways to understand the basic concept of entropy. In this thesis, we follow the interpretation given by Shannon (1948a,b) in the context of information theory, which is known as Shannon's entropy or information entropy. In this interpretation, entropy is used to measure probabilistic uncertainty. For example, a random variable with more than one possible value has uncertainty associated with it. By contrast, the entropy of a probability distribution with only one possible value is zero.

Mathematically, the entropy of a probability distribution is defined to be the expectation of the negative logarithm of the density function of the distribution. Given certain information about an unknown distribution, the Maxent principle specifies a density that is compatible with the known constraints, while allowing for maximal uncertainty (entropy) with regard to the unknown part of the distribution.

Suppose the true PDF $\pi\left(x\right)$ of a continuous random variable X is unknown, but a set of power moments $m_i = \mathbb{E}\left(X^i\right), i = 0, 1, 2, \ldots, n$ can be obtained. We aim to find a density $\widetilde{\pi}_n\left(x\right)$ that maximises the entropy of the distribution subject to the constraints that $\mathbb{E}\left(X^i\right) = m_i$ for all $i = 0, \ldots, n$. Denote Ω to be the support of X. In this thesis, our focus is the squared correlation coefficient r^2, thus we have $\Omega = [0, 1]$ for the range of r^2.

Constructing the Maxent density $\widetilde{\pi}_n\left(x\right)$ is equivalent to finding the function $\widetilde{\pi}_n$ that maximises the entropy defined below:

$$I\left(\widetilde{\pi}_n\right) = -\int_\Omega \widetilde{\pi}_n\left(x\right) \log\left\{\widetilde{\pi}_n\left(x\right)\right\} dx \tag{4.26}$$

subject to the moment constraints

$$m_i = \int_\Omega x^i\, \widetilde{\pi}_n\left(x\right) dx \quad \text{for} \quad i = 0, 1, \ldots, n. \tag{4.27}$$

The Lagrange function corresponding to this constrained optimisation problem is

$$L = -\int_\Omega \widetilde{\pi}_n\left(x\right) \log\left\{\widetilde{\pi}_n\left(x\right)\right\} dx - \sum_{i=0}^{n} \lambda_i \left(\int_\Omega x^i\, \widetilde{\pi}_n\left(x\right) dx - m_i\right). \tag{4.28}$$

By the Euler-Lagrange equation, the Maxent solution is (Liu, 2012):

$$\widetilde{\pi}_n\left(x\right) = \exp\left(\lambda_0 + \lambda_1 x + \lambda_2 x^2 + \cdots + \lambda_n x^n\right), \tag{4.29}$$

where the vector $\boldsymbol{\lambda} = \left(\lambda_0, \lambda_1, \ldots, \lambda_n\right)$ solves the unconstrained minimisation problem below:

$$\arg\min_{\boldsymbol{\lambda}} \left\{\int_\Omega \exp\left(\sum_{i=0}^{n} \lambda_i x^i\right) dx \; - \; \sum_{i=0}^{n} \lambda_i m_i\right\}. \tag{4.30}$$

This is the univariate maximum entropy principle for a continuous random variable, as needed for our problem. The multivariate principle can be derived in a similar manner (see Liu, 2012, for more details). Note that a more general Maxent density formula can also be derived given a set of constraints of the form $\mathbb{E}\left\{g_k\left(X\right)\right\}$, where g_k are arbitrary functions for $k = 1, 2, \ldots, n$, for which the expectations exist. For more details about the Maxent principle, see Jaynes (1982) and Cover and Thomas (2012).

4.6.1 Numerical Maximum Entropy Solution

It can be seen that finding the Maxent density (4.29) is equivalent to solving the unconstrained optimisation problem (4.30). The problem does not have an analytic solution for $n \geq 2$, so numerical optimisation methods are generally used. However, minimising (4.30) directly using current numerical methods is a substantial challenge, especially when the number of moment constraints is large. In this section, we discuss some strategies to address this problem following Liu (2012).

The first difficulty of the problem lies in calculating the definite integral involved in the objective function of (4.30). When the number of moments used is large, it is notoriously difficult to evaluate the integral with high accuracy because the order of the polynomial $\sum_{i=0}^{n} \lambda_i x^i$ is high. The benefit of using more moments in Maxent problems is that it can result in a more accurate approximation to the true density function $\pi\left(x\right)$, but this means that more computation time and computing power is required.

A typical solution to the problem of integral evaluation is to use Gaussian-Legendre quadrature to compute the integral numerically (see Hildebrand, 1987, for more details). Gaussian-Legendre quadrature is an approach to approximating the definite integral of a function $h\left(z\right)$ defined over the interval $[-1, 1]$ by summing up weighted function values at a set of values of z. More specifically, using quadrature nodes z_j

and weights β_j, where j varies between 1 and the total number of nodes, we have

$$\int_{-1}^{1} h\left(z\right) dz \approx \sum_j \beta_j h\left(z_j\right). \tag{4.31}$$

In this thesis, we need quadrature nodes over $[0,1]$ rather than $[-1,1]$. Gaussian-Legendre quadrature nodes over $[0,1]$ can be obtained by a linear transformation of the usual nodes. It is straightforward to see that

$$\begin{aligned} \int_{0}^{1} h\left(z\right) dz &= \frac{1}{2}\int_{-1}^{1} h\left(\frac{1}{2}z+\frac{1}{2}\right) dz \\ &\approx \frac{1}{2}\sum_j \beta_j h\left(\frac{1}{2}z_j+\frac{1}{2}\right), \end{aligned} \tag{4.32}$$

where z_j and β_j represent the usual weights and nodes. Then

$$\begin{aligned} \beta_j' &= \frac{1}{2}\beta_j \\ z_j' &= \frac{1}{2}(1+z_j) \end{aligned} \tag{4.33}$$

are weights and nodes for our problem over interval $[0,1]$. Refer to Clason and von Winckel (2012) and Liu (2012) for more details.

For our Maxent problem, suppose the Gaussian-Legendre quadrature nodes and weights are x_j and ω_j. Let x_j^i denote the jth node x_j to the power of i. Using the quadrature nodes and weights, the optimisation problem (4.30) reduces to

$$\arg\min_{\boldsymbol{\lambda}} \left\{ \sum_j \omega_j \exp\left(\sum_{i=0}^{n} \lambda_i x_j^i\right) - \sum_{i=0}^{n} \lambda_i m_i \right\}. \tag{4.34}$$

However, this is an ill-posed problem due to the power moments involved inside the objective function of (4.34). This is the second difficulty with using the Maxent principle. When $x_j \in (0,1)$, the term x_j^i becomes exceedingly close to zero for a large value of i, say 1,000. For this reason, many significant digits are required in the computation; otherwise x_j does not contribute to the result. However, the level of machine precision imposes a restriction on the number of digits. Consequently, the use of Gaussian-Legendre quadrature nodes has not resolved the problem completely.

A typical solution is to transform the original power moments $m_i = \mathbb{E}\left(X^i\right)$ into corresponding Chebyshev moments, and then apply the Maxent principle to the shifted moments. See Wheeler et al. (1974) for more details about the transformation between power moments and modified moments. Silver and Röder (1997) found that using Chebyshev moments greatly improves the solvability of the optimisation problem compared with the use of the original moments in Maxent problems. For more studies of the Maxent principle using Chebyshev moments, refer to Bandyopadhyay et al. (2005), Biswas and Bhattacharya (2010), and Liu (2012).

We first give the definition of the usual Chebyshev polynomial of the first kind, which is expressed by a recurrence relation as below:

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \tag{4.35}$$

where $x \in [-1, 1]$ and $n \geq 1$ denotes the degree of the Chebyshev polynomial. In addition, $T_0\left(x\right) = 1$ and $T_1\left(x\right) = x$. Since our problem is formulated over the interval $[0, 1]$, we need the shifted Chebyshev polynomial of the first kind, which is linked to the usual polynomial by

$$T_i^*(x) = T_i(2x - 1) \quad \text{for} \quad x \in [0, 1]. \tag{4.36}$$

Let $\{m_i^c \mid i = 0, 1, \ldots, n\}$ denote the set of shifted Chebyshev moments, where the ith moment is defined by

$$m_i^c = \mathbb{E}\left\{T_i^*\left(X\right)\right\}. \tag{4.37}$$

The numeric values of the shifted Chebyshev moments can be obtained from the original moments $m_i = \mathbb{E}\left(X^i\right)$ $(i = 0, 1, \ldots, n)$, via the expression

$$\begin{bmatrix} m_0^c \\ m_1^c \\ \vdots \\ \vdots \\ m_n^c \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & \cdots & \cdots & a_{0n} \\ a_{10} & a_{11} & \cdots & \cdots & a_{1n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n0} & a_{n1} & \cdots & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ \vdots \\ m_n \end{bmatrix}, \tag{4.38}$$

where a_{ij} is the coefficient of the term x^j in the formulation of $T_i^*(x)$. Let $\boldsymbol{A}$ denote the $(n+1) \times (n+1)$ matrix in the equation above. The entries of matrix $\boldsymbol{A}$ can be computed using equation (4.36) and the recurrence relation (4.35). We use some examples to illustrate the computation. Because $T_0(x) = 1$, we have $T_0^*(x) = T_0(2x-1) = 1$, so $a_{00} = 1$ and $a_{0j} = 0$ for $j = 1, 2, \ldots, n$. Since $T_1^*(x) = T_1(2x-1) = 2x-1$, it follows that $a_{10} = -1$ and $a_{11} = 2$ are the only two non-zero entries in the second row of $\boldsymbol{A}$.

After obtaining the set of shifted Chebyshev moments, we have the following modified optimisation problem to replace problem (4.34):

$$\arg\min_{\boldsymbol{\lambda}^c} \left\{ \sum_j \omega_j \exp\left(\sum_{i=0}^n \lambda_i^c T_{ij}^* \right) - \sum_{i=0}^n \lambda_i^c m_i^c \right\}, \tag{4.39}$$

where $T_{ij}^* = T_i^*(x_j)$ and $\boldsymbol{\lambda}^c = (\lambda_0^c, \lambda_1^c, \ldots, \lambda_n^c)$. Equation (4.39) is easier to solve using numerical methods than equation (4.34). Then, we back-transform $\boldsymbol{\lambda}^c$ to obtain the original vector $\boldsymbol{\lambda}$ by

$$\boldsymbol{\lambda} = \boldsymbol{A}^T \boldsymbol{\lambda}^c \tag{4.40}$$

and substitute $\boldsymbol{\lambda}$ into (4.29) to obtain the Maxent density in its original form.

Note that the Maxent density using the shifted Chebyshev moments is

$$\widetilde{\pi}_n^c(x) = \exp\left\{ \lambda_0^c + \lambda_1^c T_1^*(x) + \cdots + \lambda_n^c T_n^*(x) \right\}. \tag{4.41}$$

The two densities (4.29) and (4.41) provide the same Maxent approximation to the true density of X given the n power moment constraints. It is more convenient to use the density (4.41) in practice, since no back transformation is needed.

4.6.2 Trust Region Optimisation Method

The key to finding a numerical Maxent solution for the target density $\pi(x)$ is solving the optimisation problem (4.39). However, this can still be a computational challenge even though the calculation has been greatly expedited using the shifted Chebyshev moments and Gaussian-Legendre quadrature nodes. To tackle this problem, in this thesis we use the trust region optimisation method following Liu (2012). This method provides stable and fast computation for our problem. We do not describe the algorithm here; see Fletcher (1987), Wright and Nocedal (1999), and Liu (2012) for more details. An `R` package `trust` is also available (Geye, 2015). Note that the `TMB` package can also be applied to this problem, and its performance is similar to that of the trust region algorithm.

4.7 Stationary Distribution for the TLD Model

The diffusion approximation approach greatly improves the possibility of finding the stationary distribution for genetic models. Here, the genetic models in question represent diffusion processes corresponding to the original discrete Markov chain models. For example, the stationary distribution of a single-locus model with mutation was found under the diffusion approximation by Wright (1968). The stationary distributions for certain multi-locus models without linkage are also available (Wright, 1949, 1937). For a summary of genetic models whose stationary distributions are computable under the diffusion approximation, see Section 3.2 of Liu (2012).

More recently, Liu (2012) approximated the stationary distribution of the TLD model parameterised in terms of $\boldsymbol{p} = (p_1, p_2, p_3)$ by applying a three-dimensional Maxent approach to a set of moments of the form $\mathbb{E}\left(p_1^{i_1} p_2^{i_2} p_3^{i_3}\right)$, where i_1, i_2, and i_3 are non-negative integers. To prove the accuracy of the multivariate Maxent approach, Liu (2012) computed the expectation of r^2 for different settings of ρ and θ using his Maxent density $\widetilde{\pi}\left(\boldsymbol{p}\right)$:

$$\mathbb{E}\left(r^2\right) = \int_{\Omega} r^2 \left(\boldsymbol{p}\right) \widetilde{\pi}\left(\boldsymbol{p}\right) d\boldsymbol{p}, \tag{4.42}$$

and compared the results with those obtained by Song and Song (2007). Liu (2012) pointed out that the multivariate Maxent approach loses accuracy for some settings of ρ and θ, for which the density function has sharp edges. Incorporating more moments could address this problem; however, computing time increases rapidly when the number of moments increases. Therefore, the accuracy of the multivariate Maxent approach is limited by current computing power. By contrast, the accuracy and computability of a univariate Maxent approach is shown in Liu (2012) by constructing the Maxent density of a single-locus model whose exact distribution is available for comparison.

The work of Liu (2012) provides the first method for generating the stationary distribution $\pi\left(r^2\right)$ in the literature, as far as we know. However, it is based on the three-dimensional Maxent solution for $\widetilde{\pi}\left(\boldsymbol{p}\right) = \widetilde{\pi}\left(p_1, p_2, p_3\right)$, which creates a massive computational burden, firstly to find the moments $\mathbb{E}\left(p_1^{i_1} p_2^{i_2} p_3^{i_3}\right)$, and secondly to optimise the equivalent of (4.39) using a three-dimensional quadrature and multivariate Chebyshev polynomials. Liu (2012) performed computations on a high-performance computing cluster, and each setting of (ρ, θ) still required over 24 hours to compute $\widetilde{\pi}\left(p_1, p_2, p_3\right)$ using 40 CPUs. To apply Liu (2012)'s method for parameter estimation using sample data on r^2, we would need to construct this three-dimensional density $\widetilde{\pi}\left(p_1, p_2, p_3\right)$ repeatedly for numerous values of (ρ, θ), and use it to generate the univariate density $\pi\left(r^2\right)$ by integration, so each single likelihood evaluation would take over 24 hours on a parallel cluster. The method is therefore computationally impracticable.

In the following chapter, we will develop a new method to calculate the stationary PDF of r^2 directly using the univariate Maxent approach. Under our new approach, estimating mutation and recombination parameters by maximum likelihood using sample data on r^2 will become computationally feasible, and computation times will be reduced to a few minutes per likelihood evaluation on a single customary laptop.

5

A New Method for Estimating Mutation and Recombination

5.1 Overview

In this chapter, we first develop an extension to the original method of Song and Song (2007) that will enable us to compute higher-order moments of r^2 at stationarity. Then a more general and efficient approach based on a model reparametrization is proposed. Using these methods, we calculate a substantial sequence of moments of r^2, and use it to construct the stationary distribution $\widetilde{\pi}\left(r^2\right)$ under the Maxent principle. We show how this Maxent density of r^2 can be used for maximum likelihood estimation of mutation rate and recombination rate using sample observations of r^2. We also showcase other quantities of interest such as the variance of r^2, which was first computed by Liu (2012). We demonstrate the performance of the new method by simulation studies and real data analysis. Our method differs from that of Liu (2012) because we create a direct computation of $\mathbb{E}\left\{\left(r^2\right)^i\right\}$ for $i = 1, 2, 3, \ldots$, and use these moments to construct the univariate Maxent density $\widetilde{\pi}\left(r^2\right)$. By contrast, Liu (2012) derived moments $\mathbb{E}\left(p_1^{i_1} p_2^{i_2} p_3^{i_3}\right)$ and used these to construct the three-dimensional Maxent density $\widetilde{\pi}\left(p_1, p_2, p_3\right)$, from which $\widetilde{\pi}\left(r^2\right)$ could be obtained by integration.

5.2 Moment Problem: Method I

We take the computation of $\mathbb{E}\left(r^4\right)$ as an example to illustrate how to extend Song and Song (2007)'s method to compute higher-order moments of r^2 at stationarity. Although the derivation process below focuses on $\mathbb{E}\left(r^4\right)$, it can be extended naturally to other moments.

We showed in equation (4.17) that

$$r^2 = D^2 \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \left\{ p^m q^n + (1-p)^m q^n + p^m (1-q)^n + (1-p)^m (1-q)^n \right\}. \tag{5.1}$$

It follows that

$$\begin{aligned} r^4 = D^4 &\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \left\{ p^m q^n + (1-p)^m q^n + p^m (1-q)^n + (1-p)^m (1-q)^n \right\} \\ &\qquad \times \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \left\{ p^k q^l + (1-p)^k q^l + p^k (1-q)^l + (1-p)^k (1-q)^l \right\} \\ = &\sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \Big\{ D^4 p^{m+k} q^{n+l} + D^4 p^m (1-p)^k q^{n+l} + \cdots \\ &\qquad + D^4 p^k (1-p)^m (1-q)^{n+l} + D^4 (1-p)^{m+k} (1-q)^{n+l} \Big\}. \end{aligned} \tag{5.2}$$

We do not write down all 16 monomials in the formula above, but they are all of the same form $D^4 p^w (1-p)^x q^y (1-q)^z$ with w, x, y, z being either zero or expressed in terms of m, n, k, and l. Therefore $\mathbb{E}\left(r^4\right)$ can be expressed as an infinite sum of 16 expectations of the form $\mathbb{E}\left\{D^4 p^w (1-p)^x q^y (1-q)^z\right\}$ for all combinations of m, n, k, and l. We will show that the 16 expectations can be divided into four groups, each of which contains four expectations that are identical to each other.

By the symmetry of the TLD model, we can exchange p and $1-p$, and likewise q and $1-q$. Thus,

$$\begin{aligned}\mathbb{E}\left(D^4 p^{m+k} q^{n+l}\right) &= \mathbb{E}\left\{D^4 (1-p)^{m+k} q^{n+l}\right\} \\ &= \mathbb{E}\left\{D^4 p^{m+k} (1-q)^{n+l}\right\} \\ &= \mathbb{E}\left\{D^4 (1-p)^{m+k} (1-q)^{n+l}\right\}.\end{aligned} \tag{5.3}$$

Similar relationships hold for the other 12 expectations. It follows that

$$\begin{aligned}\mathbb{E}\left(r^4\right) = 4\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\Big[&\mathbb{E}\left(D^4 p^{m+k} q^{n+l}\right) + \mathbb{E}\left\{D^4 p^m (1-p)^k q^{n+l}\right\} \\ &+ \mathbb{E}\left\{D^4 p^{m+k} q^n (1-q)^l\right\} + \mathbb{E}\left\{D^4 p^m (1-p)^k q^n (1-q)^l\right\}\Big].\end{aligned} \tag{5.4}$$

Since m, n, k and l are arbitrary non-negative integers, we can exchange the positions of m and n, and the positions of l and k, which we do in the first line below. Then applying the symmetry of p and q in the second line yields

$$\begin{aligned}\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\mathbb{E}\left\{D^4 p^m (1-p)^k q^{n+l}\right\} &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\mathbb{E}\left\{D^4 p^n (1-p)^l q^{m+k}\right\} \\ &= \sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\mathbb{E}\left\{D^4 p^{m+k} q^n (1-q)^l\right\}.\end{aligned} \tag{5.5}$$

Thus, equation (5.4) can be simplified to

$$\begin{aligned}\mathbb{E}\left(r^4\right) = 4\sum_{m=0}^{\infty}\sum_{n=0}^{\infty}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty}\Big[&\mathbb{E}\left(D^4 p^{m+k} q^{n+l}\right) + 2\mathbb{E}\left\{D^4 p^m (1-p)^k q^{n+l}\right\} \\ &+ \mathbb{E}\left\{D^4 p^m (1-p)^k q^n (1-q)^l\right\}\Big].\end{aligned} \tag{5.6}$$

By the following binomial expansions,

$$(1-p)^k = \sum_{i=0}^{k} (-1)^i \binom{k}{i} p^i \quad \text{and} \quad (1-q)^l = \sum_{j=0}^{l} (-1)^j \binom{l}{j} q^j, \tag{5.7}$$

we have

$$\mathbb{E}\left\{D^4 p^m (1-p)^k q^{n+l}\right\} = \sum_{i=0}^{k} (-1)^i \binom{k}{i} \mathbb{E}\left(D^4 p^{m+i} q^{n+l}\right) \tag{5.8}$$

and

$$\mathbb{E}\left\{D^4 p^m (1-p)^k q^n (1-q)^l\right\} = \sum_{i=0}^{k} \sum_{j=0}^{l} (-1)^{i+j} \binom{k}{i} \binom{l}{j} \mathbb{E}\left(D^4 p^{m+i} q^{n+j}\right). \tag{5.9}$$

Then the final formula for computing $\mathbb{E}\left(r^4\right)$ becomes

$$\begin{aligned}\mathbb{E}\left(r^4\right) = 4 \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \Bigg\{ & \mathbb{E}\left(D^4 p^{m+k} q^{n+l}\right) + 2 \sum_{i=0}^{k} (-1)^i \binom{k}{i} \mathbb{E}\left(D^4 p^{m+i} q^{n+l}\right) \\ & + \sum_{i=0}^{k} \sum_{j=0}^{l} (-1)^{i+j} \binom{k}{i} \binom{l}{j} \mathbb{E}\left(D^4 p^{m+i} q^{n+j}\right) \Bigg\}.\end{aligned} \tag{5.10}$$

It can be seen that all expectations involved in equation (5.10) are of the same form $\mathbb{E}(D^4 p^x q^y)$ with $x, y \in \{0, 1, 2, \ldots\}$. Note that in Song and Song (2007)'s original method, inserting $f(p, q, D) = D^\gamma p^{\alpha+2-\gamma} q^{\beta+2-\gamma}$ with $\gamma = 0, 1, \ldots, \beta + 2$ into the master equation (4.14) of the TLD model leads to a system of $\beta + 3$ linear equations with $\beta + 3$ unknown expectations to be solved, where α, β, and γ are non-negative integers. When $\alpha, \beta \geq 2$, we saw in (4.24) that the $\beta + 3$ unknown expectations of the system include $\mathbb{E}(D^4 p^{\alpha-2} q^{\beta-2})$, which is needed for calculating $\mathbb{E}\left(r^4\right)$ in (5.10).

As for the calculation of $\mathbb{E}\left(r^2\right)$, we specify a truncation level $\ell_{\max} = m + n + k + l$ to obtain an approximation to $\mathbb{E}\left(r^4\right)$, namely

$$\begin{aligned}\mathbb{E}\left(r^4\right)_{\ell_{\max}} = 4 \sum_{m,n,k,l \geq 0}^{\ell_{\max}} \Bigg\{ & \mathbb{E}\left(D^4 p^{m+k} q^{n+l}\right) + 2 \sum_{i=0}^{k} (-1)^i \binom{k}{i} \mathbb{E}\left(D^4 p^{m+i} q^{n+l}\right) \\ & + \sum_{i=0}^{k} \sum_{j=0}^{l} (-1)^{i+j} \binom{k}{i} \binom{l}{j} \mathbb{E}\left(D^4 p^{m+i} q^{n+j}\right) \Bigg\}.\end{aligned} \tag{5.11}$$

We used $\ell_{\max} = 700$ following Song and Song (2007), but found that our calculation was rather slow. This is not surprising, because there is a huge number of possible combinations of (m, n, k, l) that sum to 700 or less. More importantly, for each combination of (m, n, k, l), considerable computation is required for summing up the last two terms on the right-hand side of equation (5.11).

To improve the efficiency of the algorithm, we split the expectations constituting $\mathbb{E}\left(r^4\right)_{\ell_{\max}}$ into two parts, such that

$$\mathbb{E}\left(r^4\right)_{\ell_{\max}} = E_1 + E_2, \tag{5.12}$$

where

$$E_1 = 4 \sum_{m,n,k,l\geq 0}^{\ell_{\max}} \mathbb{E}\left(D^4 p^{m+k} q^{n+l}\right) \tag{5.13}$$

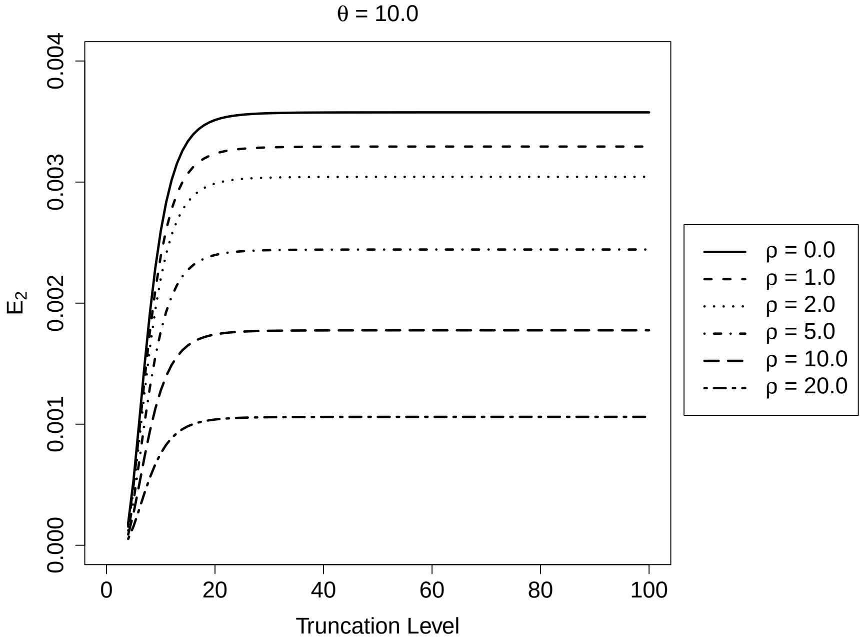
and

$$\begin{aligned} E_2 = 4 \sum_{m,n,k,l\geq 0}^{\ell_{\max}} \Bigg\{ & 2\sum_{i=0}^{k}(-1)^i \binom{k}{i} \mathbb{E}\left(D^4 p^{m+i} q^{n+l}\right) \\ & + \sum_{i=0}^{k}\sum_{j=0}^{l}(-1)^{i+j}\binom{k}{i}\binom{l}{j}\mathbb{E}\left(D^4 p^{m+i} q^{n+j}\right)\Bigg\}. \end{aligned} \tag{5.14}$$

Calculating E_1 with $\ell_{\max} = 700$ is fast. The main difficulty of the algorithm lies in computing E_2 using a large truncation level $\ell_{\max}$. The accuracy of the approximation increases as the truncation level increases, but a higher truncation level demands a longer computation time.

To address this problem, we apply different truncation levels for computing E_1 and E_2. We find empirically that the sequence of E_2 with different truncation levels converges very quickly as the truncation level increases, although the rate of convergence depends on the value of θ. It can be seen from Fig. 5.1 that when θ is large (say $\theta = 10.0$), the value of E_2 remains stable after the truncation level reaches 20. When θ is small (say $\theta = 0.1$), a truncation level of 100 is sufficiently large for an accurate approximation. Therefore, we use 100 as the truncation level to calculate E_2.

Adopting the strategy described above makes the computation faster and somewhat practicable. The computing time for $\mathbb{E}\left(r^4\right)$ for one pair of ρ and θ is approximately 30 minutes on a customary 1.3 GHz laptop. Fig. 5.2 shows the performance of the method for computing $\mathbb{E}\left(r^4\right)$ in a variety of scenarios.



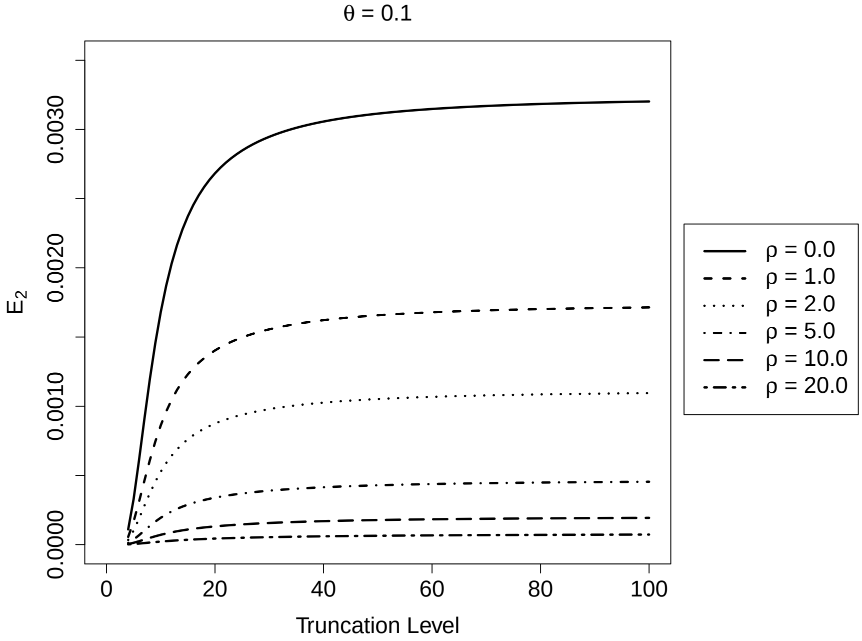
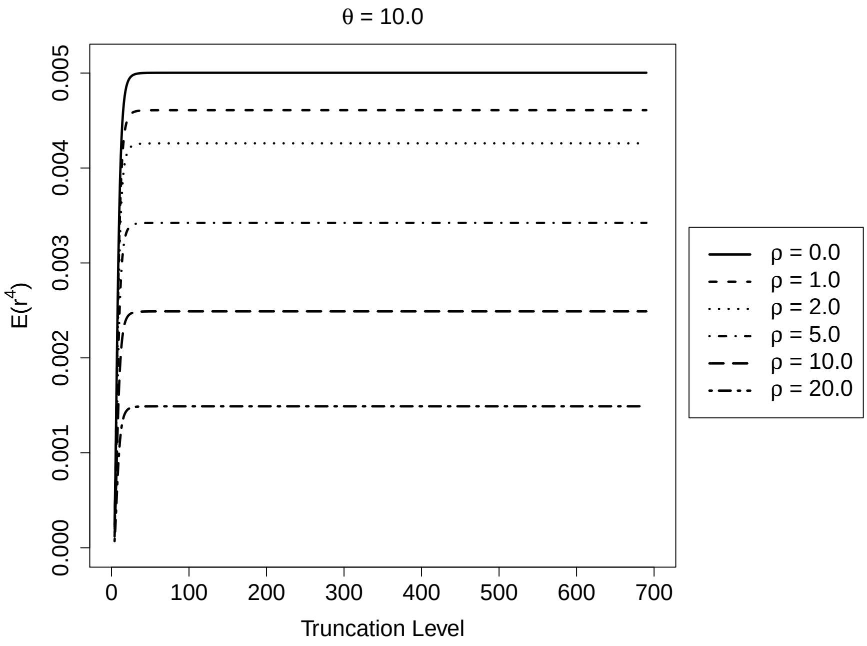


Fig. 5.1 Plots of level-truncated approximations to E_2 given different truncation levels $\ell_{\max} \in [0, 100]$ for various ρ and θ.



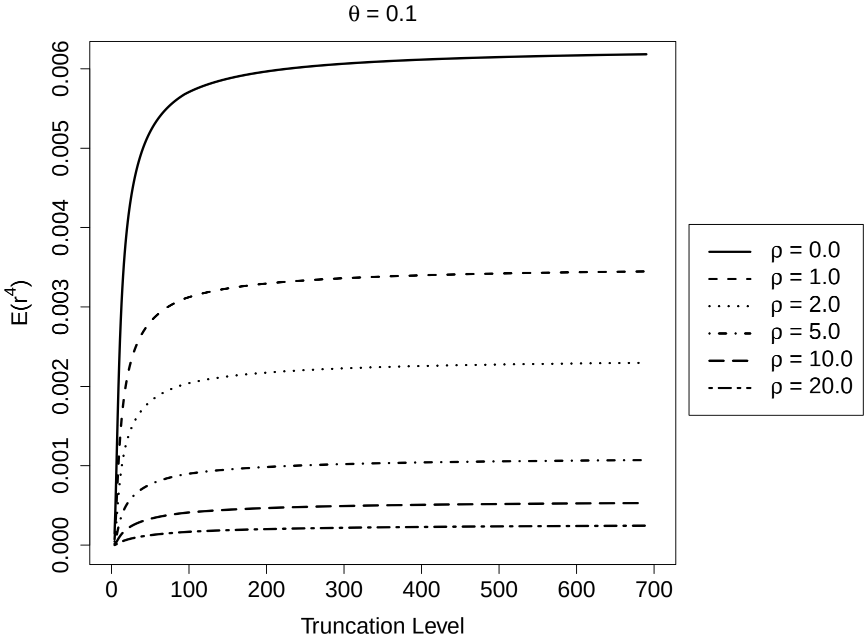


Fig. 5.2 Plots of level-truncated approximations to $\mathbb{E}\left(r^4\right)$ for various ρ and θ. The truncation level for E_2 is set to be 100, contrasting with 700 for E_1.

5.3 Moment Problem: Method II

The method discussed above provides a satisfactory solution to the computation of $\mathbb{E}\left(r^4\right)$. We find that it is also effective for computing low-order moments of r^2, such as $\mathbb{E}\left(r^6\right), \mathbb{E}\left(r^8\right)$, and $\mathbb{E}\left(r^{10}\right)$; however, it is very slow and becomes problematic when the order of the moment to be computed increases. In this section, we describe a more general and efficient approach based on a reparametrization of the model. This idea was proposed by my co-supervisor J. Goodman.

First, we reparametrize the TLD model. Let

$$u = 1 - 2p \quad \text{and} \quad v = 1 - 2q. \tag{5.15}$$

By the fact that $0 < p, q < 1$, we have $-1 < u, v < 1$ and thus $0 \leq u^2, v^2 < 1$. Since $p = (1-u)/2$ and $q = (1-v)/2$, we have

$$p\left(1-p\right) = \frac{1-u^2}{4} \tag{5.16}$$

and

$$q\left(1-q\right) = \frac{1-v^2}{4}. \tag{5.17}$$

Note that $\partial/\partial p = -2\partial/\partial u$ and $\partial/\partial q = -2\partial/\partial v$. Then the diffusion generator (4.12) of the TLD model in terms of the new parameters (u, v, D) can be written as

$$\begin{aligned}\mathcal{L} = {} & \frac{1}{2}\left(1-u^2\right)\frac{\partial^2}{\partial u^2} + \frac{1}{2}\left(1-v^2\right)\frac{\partial^2}{\partial v^2} + \frac{1}{2}\left\{\frac{1}{16}\left(1-u^2\right)\left(1-v^2\right) + Duv - D^2\right\}\frac{\partial^2}{\partial D^2} \\ & + 4D\frac{\partial^2}{\partial u \partial v} - 2Du\frac{\partial^2}{\partial D \partial u} - 2Dv\frac{\partial^2}{\partial D \partial v} - \frac{1}{2}\theta u\frac{\partial}{\partial u} - \frac{1}{2}\theta v\frac{\partial}{\partial v} \\ & - D\left(1 + \frac{1}{2}\rho + \theta\right)\frac{\partial}{\partial D}.\end{aligned} \tag{5.18}$$

The definition of r^2 in terms of the new parameters is

$$r^2 = \frac{D^2}{p\,(1-p)\,q\,(1-q)} = \frac{16D^2}{(1-u^2)\,(1-v^2)}. \tag{5.19}$$

For $0 < u^2, v^2 < 1$, we have

$$\frac{1}{1-u^2} = \sum_{k=0}^{\infty} u^{2k} \quad \text{and} \quad \frac{1}{1-v^2} = \sum_{l=0}^{\infty} v^{2l}. \tag{5.20}$$

Substituting equation (5.20) into (5.19) yields

$$r^2 = 16 \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} D^2 u^{2k} v^{2l}, \tag{5.21}$$

so that

$$\mathbb{E}\left(r^2\right) = 16 \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \mathbb{E}\left(D^2 u^{2k} v^{2l}\right). \tag{5.22}$$

Similar expressions can be derived for computation of other moments of r^2. For the moment of r^2 of order $M = 1, 2, \ldots$, we obtain from (5.21),

$$\begin{aligned} \mathbb{E}\left(r^{2M}\right) &= \mathbb{E}\left\{16^M \left(\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} D^2 u^{2k} v^{2l}\right)^M\right\} \\ &= 16^M \sum_{k_1=0}^{\infty} \cdots \sum_{k_M=0}^{\infty} \sum_{l_1=0}^{\infty} \cdots \sum_{l_M=0}^{\infty} \mathbb{E}\left\{D^{2M} u^{2(k_1+\cdots+k_M)} v^{2(l_1+\cdots+l_M)}\right\}. \end{aligned} \tag{5.23}$$

Note that in the equation above the value of the expectation

$$\mathbb{E}\left\{D^{2M} u^{2(k_1+\cdots+k_M)} v^{2(l_1+\cdots+l_M)}\right\} \tag{5.24}$$

only involves the two sums $K = k_1 + \cdots + k_M$ and $L = l_1 + \cdots + l_M$, regardless of their compositions. Given a positive integer K, the number of all possible non-negative and integer-valued vectors $(k_1, \ldots, k_M)$ satisfying $K = k_1 + \cdots + k_M$ is

$$N_{K,M} = \binom{K+M-1}{M-1}. \tag{5.25}$$

To see this, we consider an example. Suppose $K+M-1$ bubbles are placed in line. The number of ways of choosing $M-1$ of the bubbles is $N_{K,M}$. The $M-1$ selected bubbles partition the remaining K bubbles into M groups, whose counts are $k_1, \ldots, k_M$ respectively. Thus the number of different ways to write a positive integer K as the sum of M non-negative integers is therefore $N_{K,M}$ as shown in equation (5.25). Similarly, the number of ways to write $L = l_1 + \cdots + l_M$ is

$$N_{L,M} = \binom{L+M-1}{M-1}. \tag{5.26}$$

Thus equation (5.23) can be simplified to

$$\mathbb{E}\left(r^{2M}\right) = 16^M \sum_{K=0}^{\infty} \sum_{L=0}^{\infty} \binom{K+M-1}{M-1} \binom{L+M-1}{M-1} \mathbb{E}\left(D^{2M} u^{2K} v^{2L}\right). \tag{5.27}$$

The problem of computing the moment of r^2 of order M therefore reduces to computing expectations of the form $\mathbb{E}\left(D^{2M} u^i v^j\right)$ for all combinations of non-negative even numbers i and j. To compute these expectations, we follow the procedure of Song and Song (2007)'s original algorithm described in Section 4.5, but use parameters u and v to replace parameters p and q respectively, and use the new diffusion generator (5.18) to replace the original generator (4.12). For example, we can see from (4.24) that if m and n are both even numbers that are large enough, the $n+3$ expectations contain expectation $\mathbb{E}\left(D^4 u^{m-2} v^{n-2}\right)$ for computing $\mathbb{E}\left(r^4\right)$, expectation $\mathbb{E}\left(D^6 u^{m-4} v^{n-4}\right)$ for computing $\mathbb{E}\left(r^6\right)$, and other expectations for computing higher-order moments of r^2.

Setting $\ell_{\max} = 2(K+L) = 700$ as the truncation level, we use

$$\mathbb{E}\left(r^{2M}\right)_{\ell_{\max}} = 16^M \sum_{K,L \geq 0}^{\ell_{\max}} \binom{K+M-1}{M-1} \binom{L+M-1}{M-1} \mathbb{E}\left(D^{2M} u^{2K} v^{2L}\right) \tag{5.28}$$

as an approximation to $\mathbb{E}\left(r^{2M}\right)$. With this reparametrized formulation, our method costs roughly 100 seconds to compute M (say $M = 20$) moments of r^2 simultaneously, and the computation time remains almost the same if more moments are computed. Song and Song (2007)'s method requires a similar amount of computation time; however, their method only gives $\mathbb{E}\left(r^2\right)$, whereas ours gives $\mathbb{E}\left(r^{2M}\right)$ for all $M = 1, \ldots, 20$ simultaneously.

5.4 Variance of r^2 in the TLD Model

The method based on the reparametrization (D, u, v) is general and efficient, so it is the primary method we use for computing moments of r^2 in this thesis. To test our computation, we compute a series of moments of r^2 for various values of ρ and θ. Results for $\mathbb{E}\left(r^2\right)$ from our method are identical (to eight decimal places) to those obtained by Song and Song (2007); see Table 1 in their paper or Table 4.2 in Liu (2012) for details. This provides evidence for the validity of our proposed method, since the two methods adopt different parameters and diffusion generators.

In this section, we focus on $\mathbb{V}\left(r^2\right)$, the variance of r^2, which might be of interest in some contexts. This can be computed from our method using the first two moments of r^2, via

$$\mathbb{V}\left(r^2\right) = \mathbb{E}\left(r^4\right) - \left\{\mathbb{E}\left(r^2\right)\right\}^2. \tag{5.29}$$

As far as we know, $\mathbb{V}\left(r^2\right)$ in the TLD model has previously only been obtained by Liu (2012); however, Liu (2012) indicated that his computation was not satisfactory for small θ due to limitations of computing power. Liu (2012)'s method involved finding the Maxent density $\widetilde{\pi}\left(p_1, p_2, p_3\right)$ and integrating with respect to this density to find $\mathbb{V}\left(r^2\right)$; whereas our method for $\mathbb{V}\left(r^2\right)$ is analytic and does not involve any density approximation or numerical optimisation or integration; only solutions of systems of linear equations.

Table 5.1 and Fig. 5.3 provide a comparison of $\mathbb{V}\left(r^2\right)$ between our new analytic method and Liu (2012)'s multivariate Maxent approach. When θ is large, say $\theta > 1$, the difference between the two sets of results is negligible; however, when θ is small, the difference tends to be larger. Note that for the setting of $\rho = 20$ and $\theta = 0.6$, the two methods give noticeably different results, producing the outlier shown in Fig. 5.3. This appears to be a typo or other error in Liu (2012). Based on Table 5.1, it can be seen that the variance of r^2 is a strictly decreasing function of ρ if θ is fixed. The results of Liu (2012) also mirror this pattern, except for the sole setting of $\rho = 20$ and $\theta = 0.6$, so we presume this to be an error. When ρ is fixed, $\mathbb{V}\left(r^2\right)$ typically increases to a peak, and then declines again as θ increases.

Table 5.1 Comparison of $\mathbb{V}\left(r^2\right)$ between the proposed analytic method and Liu (2012)'s Maxent method.

ρ	θ										
	0.1	0.2	0.4	0.6	0.8	1.0	2.0	4.0	6.0	8.0	10.0
(a) Results of the analytic method II using truncation level $\ell_{\max} = 700$											
0.0	0.006202	0.016378	0.03133	0.03739	0.03826	0.03673	0.02426	0.01166	0.00685	0.00452	0.003200
1.0	0.003478	0.009620	0.01972	0.02473	0.02629	0.02603	0.01893	0.00992	0.00608	0.00410	0.002952
2.0	0.002329	0.006563	0.01388	0.01786	0.01940	0.01957	0.01521	0.00855	0.00542	0.00374	0.002732
5.0	0.001101	0.003137	0.00680	0.00896	0.00998	0.01032	0.00895	0.00581	0.00400	0.00291	0.002203
10.0	0.000555	0.001567	0.00335	0.00440	0.00491	0.00511	0.00474	0.00349	0.00263	0.00203	0.001610
20.0	0.000264	0.000727	0.00150	0.00191	0.00210	0.00216	0.00204	0.00168	0.00139	0.00115	0.000969
(b) Results of Liu (2012) using a three-dimensional Maxent approach											
0.0	0.005934	0.015332	0.02908	0.03516	0.03711	0.03569	0.02436	0.01175	0.00687	0.00454	0.003212
1.0	0.003226	0.008784	0.01805	0.02248	0.02500	0.02526	0.01902	0.00999	0.00612	0.00413	0.002963
2.0	0.002119	0.005896	0.01350	0.01653	0.01837	0.01897	0.01531	0.00862	0.00544	0.00377	0.002743
5.0	0.000972	0.002723	0.00609	0.00812	0.00943	0.00993	0.00905	0.00587	0.00404	0.00293	0.002213
10.0	0.000487	0.001292	0.00295	0.00388	0.00459	0.00476	0.00482	0.00356	0.00266	0.00206	0.001619
20.0	0.000241	0.000626	0.00136	0.00272	0.00198	0.00209	0.00211	0.00174	0.00142	0.00117	0.000976

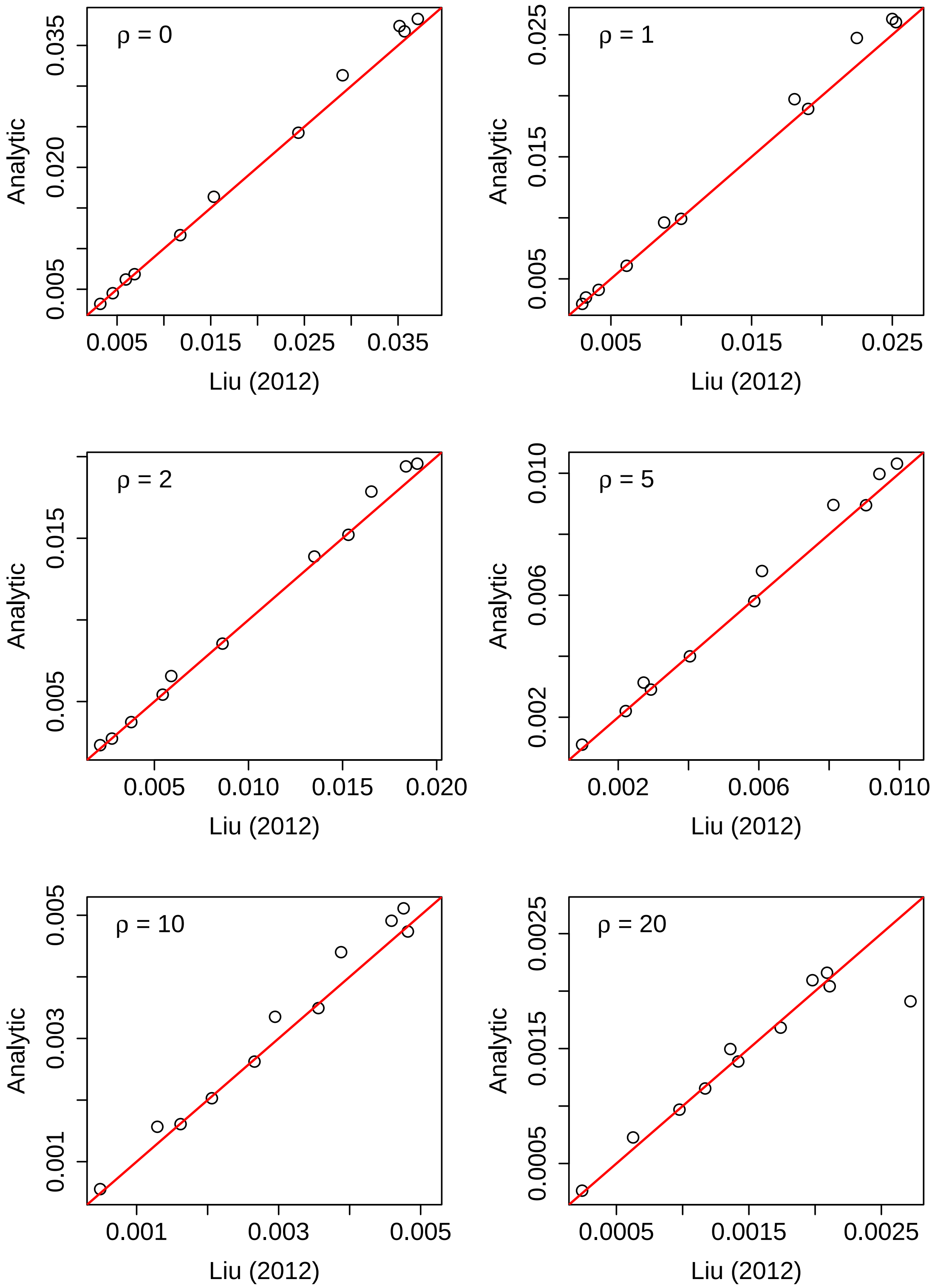


Fig. 5.3 Variances of r^2 computed by the proposed analytic method under various settings shown against the results obtained by Liu (2012). Points on the red lines indicate that results from the two approaches are identical. The outlier in the panel for $\rho = 20$ probably results from a typo in Liu (2012).

5.5 Maxent Density of r^2

We have shown that it is feasible in both efficiency and accuracy to compute a finite sequence of moments of r^2 without knowing its underlying probability distribution. The moments obtained can be used to construct the density function of r^2 via the univariate Maxent approach. In general, the larger the number of moments used, the closer is the Maxent density to the true density function of r^2. However, incorporating more moments incurs a larger computation time, since the complexity of solving the unconstrained optimisation problem (4.39) increases. One important purpose of this project is to apply the Maxent density of r^2 for parameter estimation, so we need consider both efficiency and accuracy of the proposed method.

5.5.1 Sequential Updating Method

To decide the most appropriate number of moments that should be used for constructing the Maxent density of r^2, we propose a criterion based on the sequential updating method of Wu (2003). Instead of taking into account all the moments available simultaneously, this criterion proposes that we incorporate the moments one by one from lower to higher order, and update the Maxent density of r^2 sequentially until some threshold of accuracy is reached.

Define the moments of r^2 of order up to n that are computable using our method to be

$$\left\{m_i \mid m_i = \mathbb{E}\left(r^{2i}\right), i = 1, 2, \ldots, n\right\}. \tag{5.30}$$

If the first k moments, m_i for $i = 1, 2, \ldots, k$, are used, the solution of (4.39) is a $(k+1)$-dimensional vector $\boldsymbol{\lambda}_k^c$, from which $\boldsymbol{\lambda}_k = (\lambda_0, \lambda_1, \ldots, \lambda_k)$ can be obtained by the linear transformation (4.40). The Maxent density of r^2 in this case is given by (4.29) as

$$\widetilde{\pi}_k\left(r^2\right) = \exp\left(\lambda_0 + \lambda_1 r^2 + \cdots + \lambda_k r^{2k}\right). \tag{5.31}$$

Using this Maxent density of order k, the moment of r^2 of order $k+1$ can be predicted by

$$\widetilde{m}_{k+1} = \int_0^1 x^{k+1} \widetilde{\pi}_k (x)\, dx. \tag{5.32}$$

The difference between the predicted moment $\widetilde{m}_{k+1}$, based on the kth-order approximation $\widetilde{\pi}_k$, and the true moment m_{k+1}, from the analytic computation, serves as an indicator to decide whether more moments are needed. If $\widetilde{m}_{k+1}$ is very close to m_{k+1}, this suggests that almost all information contained in m_{k+1} is already provided by the first k moments. Thus there is no need to incorporate the moment m_{k+1}. We use the percentage bias b_k defined below to measure the difference between m_{k+1} and $\widetilde{m}_{k+1}$:

$$b_k = \frac{\widetilde{m}_{k+1} - m_{k+1}}{m_{k+1}} \times 100\%. \tag{5.33}$$

Note that b_k can be regarded as an indicator of the performance of using moments of order up to k to construct the density of r^2. In general, b_k tends towards zero as k increases. In our computation, we first set a threshold for b_k at a small percentage, say 1%. Starting with $k = 1$ moment, we increase k by one at each step until b_k is found to be under the threshold. We then use the Maxent density $\widetilde{\pi}_k$ as our final estimate.

The optimisation problem (4.39) becomes very sensitive to the starting value of $\boldsymbol{\lambda}^c$ when the number of moments incorporated is large. To deal with this problem, we adopt the strategy proposed by Wu (2003). When $k = 1$, we use $\boldsymbol{\lambda}_1^{c*} = (0, 0)$ as the starting value for finding $\boldsymbol{\lambda}_1^c$. When $k \geq 2$, we use $\boldsymbol{\lambda}_k^{c*} = \left(\boldsymbol{\lambda}_{k-1}^c, 0\right)$ as the starting value for finding $\boldsymbol{\lambda}_k^c$. In this case, the solution of the optimisation problem can generally be obtained within a small number of iterations.

5.5.2 Results

The performance of the sequential updating method is shown in Fig. 5.4. In the two scenarios, the indicator b_k approaches zero as the number of moments k increases.

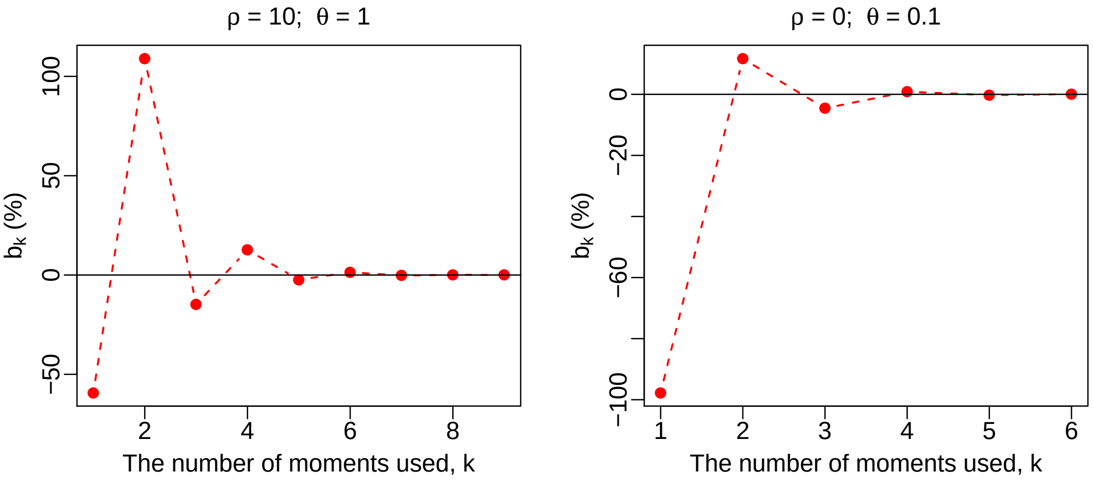


Fig. 5.4 Relationship between the percentage bias b_k and the number of moments used, k, for $(\rho, \theta) = (10, 1)$ (left) and $(\rho, \theta) = (0, 0.1)$ (right).

The rate of convergence depends on the values of ρ and θ. For example, b_4 is almost zero when $(\rho, \theta) = (0, 0.1)$, while for $(\rho, \theta) = (10, 1)$, it is clearly larger than zero.

By the sequential updating method, we can construct the Maxent density of r^2 for any setting of (ρ, θ). For the problems we consider in this chapter, the number of Gaussian-Legendre quadrature nodes used for numerical integration is set to be 1,000. Fig. 5.5 presents Maxent densities of r^2 for four different scenarios. Here, we use 1% as the threshold to decide when to terminate the sequential updating algorithm. A smaller threshold means that more moments will be incorporated. Computation time for constructing the Maxent density of r^2 for one pair of (ρ, θ) is roughly two minutes on a customary laptop.

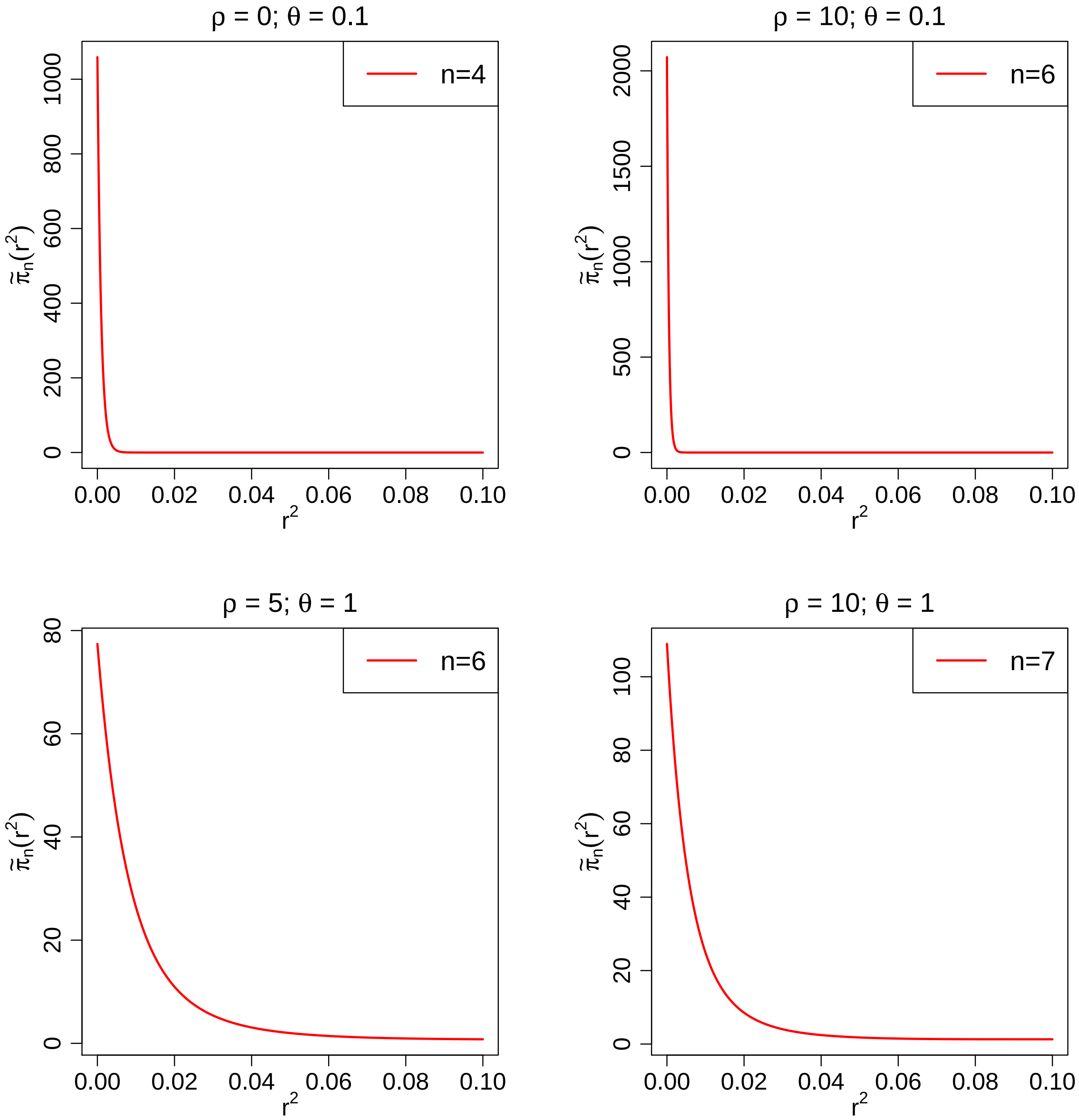


Fig. 5.5 Maxent density functions of r^2 for four different settings of ρ and θ, constructed using moments of order up to n. The horizontal axis is confined to the interval $[0, 0.1]$ for clarity, although the support of r^2 is $[0, 1]$.

5.6 Maximum Likelihood Estimation

Since the stationary distribution of r^2 can now be constructed for any combination of ρ and θ, the likelihood function for any ρ and θ can be approximated by the Maxent density $\widetilde{\pi}_k\left(r^2;\rho,\theta\right)$, where k, the number of moments used, depends on the values of ρ and θ. In this section, we demonstrate how to use the Maxent density of r^2 to estimate ρ and θ from simulated data using maximum likelihood.

5.6.1 Rejection Sampling

The Maxent density of r^2 we construct is based on the diffusion approximation, so the population size N needs to be sufficiently large for this approximation to be reasonable. For this reason, it is infeasible to simulate the discrete TLD model directly due to the enormous state space $\mathcal{S}\subset\mathbb{N}^3$ as shown in equation (4.7). Adequate sampling of such a large state space at equilibrium is computationally impracticable. As far as we know, it is also difficult to sample from the diffusion process corresponding to the TLD model. Since the Maxent density of r^2 for any pair of (ρ,θ) is obtained, we instead generate samples of r^2 directly from this distribution by rejection sampling.

Rejection sampling is a general and flexible method for generating independent observations from an arbitrary probability distribution in $\mathbb{R}^n$ for $n=1,2,\ldots$. We briefly describe the method following Gilks and Wild (1992).

Algorithm 3 Rejection sampling method

```
for $i = 1$ to $m$ do
    Sample $y$ from the distribution of $Y$
    Sample $u$ from the Uniform(0,1) distribution
    Calculate the ratio $r = f(y) / \{cg(y)\}$
    Accept $y$ as a sample if $u < r$; otherwise reject $y$
end for
```

Suppose we aim to draw a sample of observations from a random variable X with density $f(x)$. Sampling from X is difficult, but it is straightforward to sample from another variable Y with density $g(y)$. It is required that there exists a constant

c such that $f(x) \leq cg(x)$ holds for all values of x taken from the support of X. The basic rejection sampling method is shown in Algorithm 3. In the algorithm, m denotes the number of iterations, which should be a sufficiently large integer set by the user to ensure that the sample generated is large enough. For our problem, we use the Uniform(0, 1) distribution for Y and use the maximum value of $\widetilde{\pi}_k$ for c. The performance of this approach will be demonstrated in simulation studies shown below.

5.6.2 General Procedure

Here, we describe a general procedure for simulating data and estimating parameters using the Maxent density of r^2. Suppose we conduct simulations using the setting (ρ_0, θ_0). The procedure consists of four parts.

1. *Specifying the Maxent density of* r^2

We first compute the first $n = 20$ moments of r^2 for the setting (ρ_0, θ_0). Then we implement the sequential updating method of Section 5.5.1 to construct the Maxent density of r^2. Suppose k_0 moments are selected, and the corresponding Maxent density is $\widetilde{\pi}_{k_0}\left(r^2; \rho_0, \theta_0\right)$.

2. *Generating observations of* r^2

We generate a sample of independent observations of r^2 from the density $\widetilde{\pi}_{k_0}\left(r^2; \rho_0, \theta_0\right)$ using the rejection sampling method. Let $\left\{r_1^2, r_2^2, \ldots, r_K^2\right\}$ denote the sample, where K is the sample size.

3. *Specifying the log-likelihood function*

The true log-likelihood function is

$$l(\rho, \theta) = \sum_{i=1}^{K} \log \pi\left(r_i^2; \rho, \theta\right). \tag{5.34}$$

During the process of maximum likelihood estimation, this log-likelihood function needs to be evaluated multiple times at a range of different pairs $\left(\rho_j, \theta_j\right)$ with $j = 1, 2, \ldots$. For each $\left(\rho_j, \theta_j\right)$, we follow the same procedure as in step 1 to find the Maxent density $\widetilde{\pi}_{k_j}\left(r^2; \rho_j, \theta_j\right)$, which is an approximation to the exact density $\pi\left(r^2; \rho_j, \theta_j\right)$. Then the Maxent log-likelihood function evaluated at $\left(\rho_j, \theta_j\right)$ is

$$l^*\left(\rho_j, \theta_j\right) = \sum_{i=1}^{K} \log \widetilde{\pi}_{k_j}\left(r_i^2; \rho_j, \theta_j\right). \tag{5.35}$$

4. *Calculating maximum likelihood estimates*

We use the `R` optimiser `nlm` to find the estimates of (ρ, θ). Variance estimates and 95% confidence intervals are obtained using the inverse Hessian matrix as in Section 3.3.1.

5.6.3 Simulation Study

To show the performance of the maximum likelihood approach described above, we conducted simulation studies in a number of settings.

As an example, we used $(\rho_0, \theta_0) = (10, 1)$. To make the algorithm faster, we used 5% as the threshold for terminating the sequential updating algorithm (see Section 5.5.1). We found that the percentage bias b_5 for the prediction of the sixth moment was under 5%, so we used moments of order up to five to construct the Maxent density of r^2, i.e. $\widetilde{\pi}_5\left(r^2; \rho_0, \theta_0\right)$. Then we generated a sample of 10,000 observations of r^2 from the density function $\widetilde{\pi}_5\left(r^2; \rho_0, \theta_0\right)$ using the rejection sampling method. The raw data are shown in Fig. 5.6, from which we can also see the good performance of the rejection sampling method.

From the sample data, the estimates of the parameters were $\left(\widehat{\rho}, \widehat{\theta}\right) = (9.86, 1.03)$ with 95% confidence intervals $(9.32, 10.40)$ for $\widehat{\rho}$ and $(0.99, 1.06)$ for $\widehat{\theta}$. The results have reasonably good precision for both parameters. On a customary 1.3 GHz laptop, the computation took approximately 25 minutes to generate the data and obtain the maximum likelihood estimates, so the approach is also feasible in terms of efficiency.

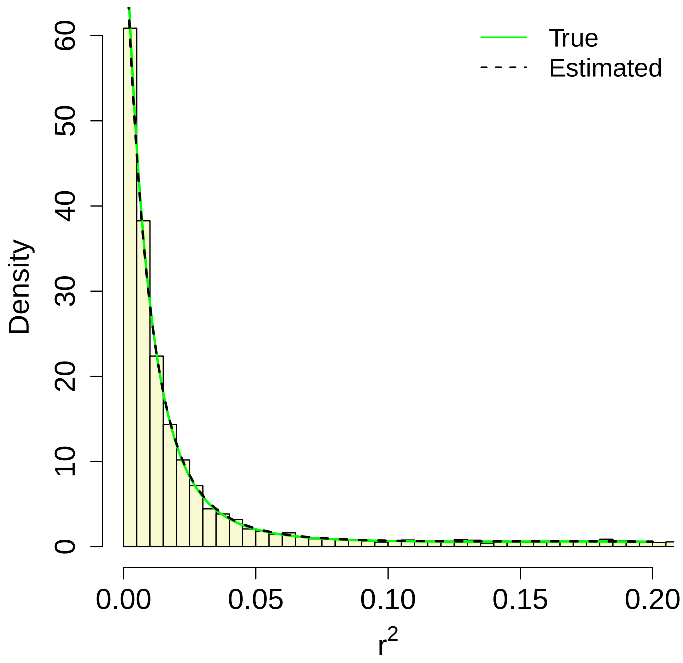


Fig. 5.6 Histogram of sample data of r^2 generated by the rejection sampling method from the Maxent density $\widetilde{\pi}_5\left(r^2;\rho_0,\theta_0\right)$ when $\rho_0 = 10$ and $\theta_0 = 1$. The green solid curve on the plot represents the true Maxent density $\widetilde{\pi}_5\left(r^2;\rho_0,\theta_0\right)$. The dashed curve represents the Maxent density constructed using the estimates $\left(\hat{\rho},\hat{\theta}\right) = (9.86, 1.03)$.

We repeated the procedure above 100 times. Inference results are shown in Fig. 5.7. It can be seen that our method yields approximately unbiased estimation for both parameters with satisfactory confidence interval coverages for 95% confidence intervals.

We also conducted simulation studies using another setting of $(\rho_0, \theta_0) = (5, 1)$. Results shown in Fig. 5.8 indicate that the method has a similar performance in this case. However, both parameters are estimated with lower precision, indicating that the influence of ρ and θ on the distribution of r^2 becomes more subtle as ρ decreases, corresponding to more distant loci or smaller population sizes.

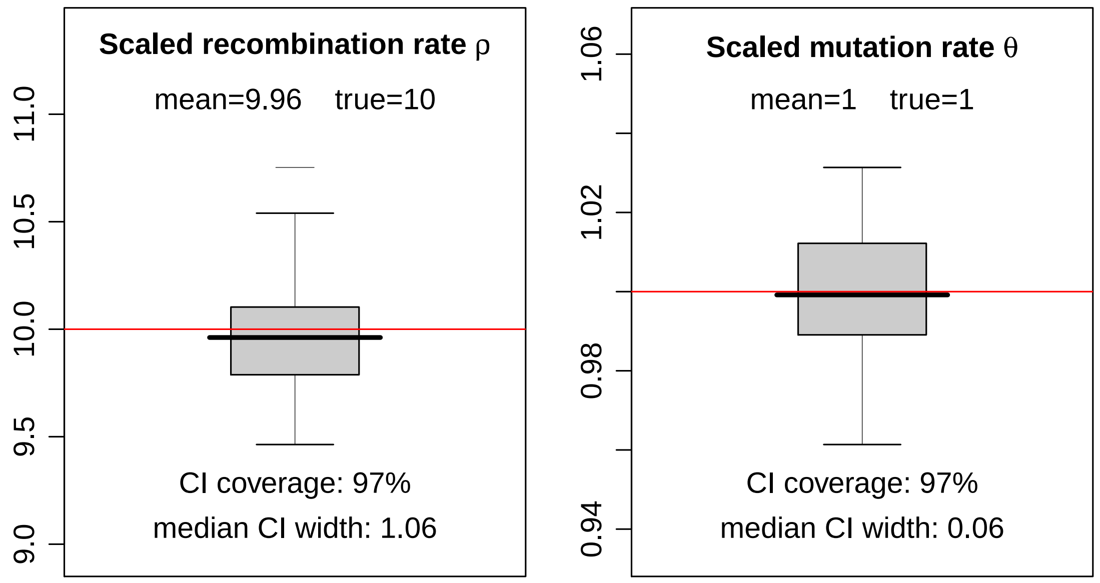


Fig. 5.7 Distributions of parameter estimates from 100 simulations when $\rho = 10$ and $\theta = 1$. The red horizontal lines across the plots show the true values of the parameters. The black horizontal lines across the boxes show the means of the estimates.

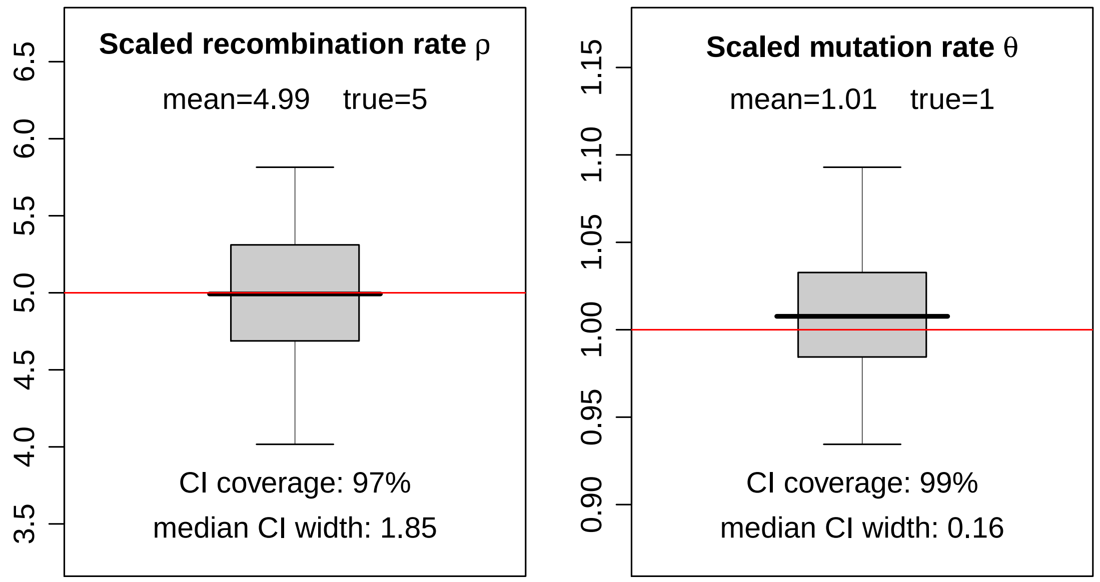


Fig. 5.8 Distributions of parameter estimates from 100 simulations when $\rho = 5$ and $\theta = 1$. Other details of the plots are the same as those for Fig. 5.7.

5.7 Application to 1000 Genomes Data

Simulation studies have shown that the proposed maximum likelihood approach using the Maxent density of r^2 performs well for estimating mutation rate and recombination rate from sample data of r^2. In this section, we apply the method to analysis of real data taken from the 1000 Genomes Project (1000 Genomes Project Consortium, 2015, 2012, 2010).

The 1000 Genomes Project is an international research consortium that aims to generate a detailed description of genetic variation for humans, by applying DNA sequencing to a sample of individuals from a variety of populations in the world. The project consisted of three phases and has now been completed. See 1000 Genomes Project Consortium (2012, 2010) for outputs of the first two phases of the project. Here, we focus on data from the last phase, in which DNA sequences were obtained from 2,504 individuals from five super-populations, namely Africa (AFR), East Asia (EAS), Europe (EUR), South Asia (SAS), and the Americas (AMR) (1000 Genomes Project Consortium, 2015). The original data can be found and downloaded from `ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/`.

The data are presented in the form of variant call format (VCF) files. It is problematic to use `R` to process the data due to the large size of the data files. Instead, we use the software package `VCFtools` developed by Danecek et al. (2011). For instructions on implementing the software, see `https://vcftools.github.io/index.html`.

5.7.1 Data Generation

We use data from the 1000 Genomes Project to generate samples of r^2. These data supply raw gamete-types, not just genotypes, so the sample value of r^2 can be calculated directly from the sample equivalent of equations (4.1) and (4.2). To make our analysis reliable and useful, we need to follow a few rules when generating samples.

Firstly, we use data from a single chromosome and a single population. This is easy to achieve because the data are saved into different VCF files according to which chromosome they are taken from. Extracting data of individuals from the same population is straightforward using `VCFtools`; see Appendix C for the code to complete this operation.

Secondly, we focus on pairs of biallelic loci, each with minor allele frequencies of at least 5%. This is common practice in genetic studies and can also be achieved easily via `VCFtools`.

Thirdly, our method gives one estimate of (ρ, θ) from one sample of r^2, so we require that the underlying parameters ρ and θ for all observations in the sample should be the same, in other words that the sample is generated from a distribution of r^2 with the same value of (ρ, θ). Biologically, mutation rate might or might not remain the same for alleles on the same chromosome, while recombination rate between two genetic loci certainly depends on the distance between them. Our analysis assumes that mutation rate is constant and that distance is the only factor that influences recombination. Therefore we choose numerous pairs of loci each with spacing d within the pair, where d is measured in base pairs (bps). Each pair of loci at spacing d generates one sample observation of r^2, where this is the square of the empirical correlation between the allele in locus 1 and the allele in locus 2 of the pair, and the correlation is taken across all people in the sample from the target population.

Finally, the loci we choose should be close to each other on the chromosome, because recombination rate varies along the chromosome even for a fixed d, and there might exist recombination hotspots. A recombination hotspot is a region on a chromosome within which the recombination rate is much higher than that of its surrounding regions (Auton et al., 2014; Jeffreys et al., 2001). To satisfy this condition, we first select a value of d and then search for pairs of loci each with spacing d that are located within a single "zone", where we define a zone to be a region of the

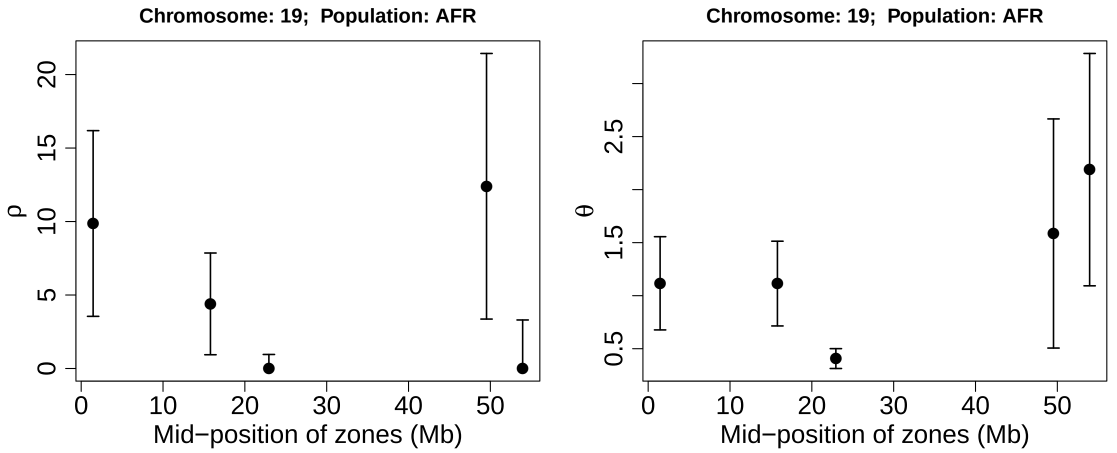


Fig. 5.9 Plots of maximum likelihood estimates and 95% confidence intervals for parameters ρ and θ in five different zones on chromosome 19 for the AFR population. In the plots, one Mb represents one million base pairs. The left and right panels show estimates of ρ and θ respectively.

chromosome of length $l \geq d$. We select zones arbitrarily, with a view to comparing estimated recombination rate along the length of the chromosome.

In practice, the SNP data available are from loci with irregular spacing, so to generate sufficiently large samples for analysis we need to use an interval for the within-pair distance between loci instead of a single value, such as $(d - \Delta d, d + \Delta d)$. Here, Δd is a small distance compared with d, so that all pairs of loci with spacing in this interval can be regarded as having approximately the same within-pair distance. Thus we extract data for analysis by searching for all pairs of loci in the raw data using parameters $(d, \Delta d, l)$, which are all measured in bps: see Appendix C for details.

5.7.2 Results

For an example, we use data from chromosome 19, and focus on population AFR to show the performance of the proposed method. We search for pairs of loci using the settings $(d, \Delta d, l) = (5000, 100, 5000)$ and require each zone to contain at least 40 pairs of loci for useful inference. We hoped to find more locus pairs for each zone, but unfortunately it is hard to find pairs that meet all of our requirements. With 40 pairs of loci, we obtain sample data $\left(r_1^2, \ldots, r_{40}^2\right)$ for each zone.

Fig. 5.9 shows our estimation results from the five zones that fulfilled our requirements. The parameter estimates are very imprecise, and mostly have wide confidence intervals. This can be explained by the small sample sizes of only 40 observations per zone.

To make more precise inference, we relax the requirement that all pairs of loci need be located within a focused zone, and select pairs of loci that have approximately the same within-pair distance from throughout a whole chromosome. Figures 5.10 and 5.11 present eight scenarios with $d = 50,000$ and $d = 100,000$ in which Maxent maximum likelihood densities of r^2 have a moderate, although not wholly satisfactory, fit to the sample data. 95% confidence intervals for these cases are much narrower than those obtained in the example above. The sample data consistently exhibit a slightly steeper decline than the fitted curves, calling into question whether the TLD model under the diffusion approximation with a single (ρ, θ) per sample is wholly adequate for these data.

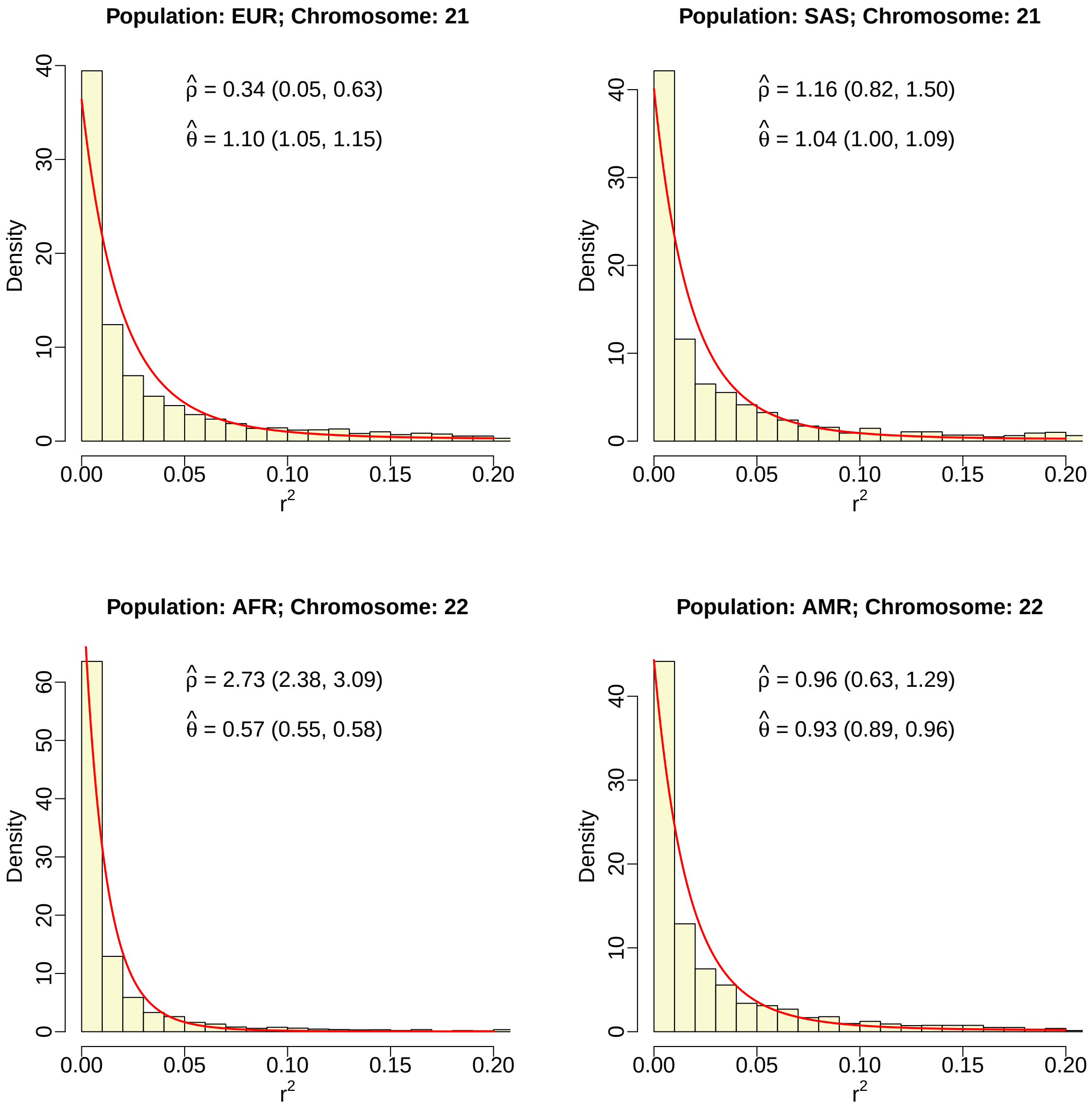


Fig. 5.10 Histograms of sample observations of r^2 shown against fitted Maxent densities. In each of the four scenarios, we use pairs of loci with spacing $d = 50,000$ bps from the whole chromosome, and apply a leeway of $\Delta d = 5$ bps. Sample sizes are 3329 (top left), 3507 (top right), 6255 (bottom left), and 3567 (bottom right) respectively for the four scenarios.

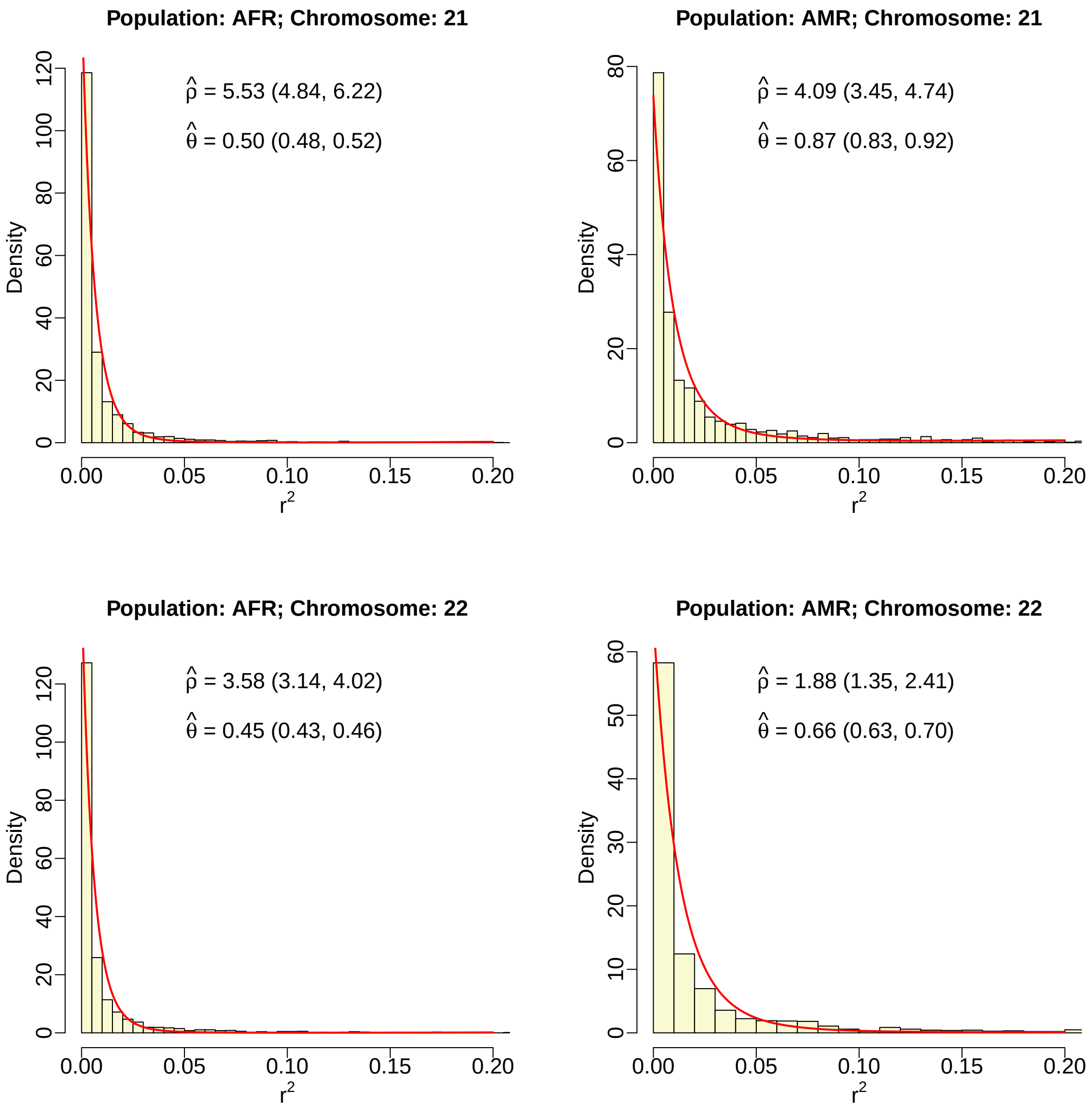


Fig. 5.11 Histograms of sample observations of r^2 shown against fitted Maxent densities. In each of the four scenarios, we use pairs of loci with spacing $d = 100,000$ bps from the whole chromosome, and apply a leeway of $\Delta d = 5$ bps. Sample sizes are 3441 (top left), 1838 (top right), 3407 (bottom left), and 1883 (bottom right) respectively for the four scenarios.

5.8 A Different Inference Problem

The application of the proposed maximum likelihood approach in estimating the scaled mutation rate θ and recombination rate ρ from sample data of r^2 in the previous section is somewhat compromised by the difficulty of finding sufficiently many locus pairs with common spacing d which are located in a focused region of the chromosome and satisfy our requirements for minor allele frequency. In this section, we describe how we might use the method to draw inference from data in a more achievable setting.

We reformulate the recombination rate C using Haldane's mapping function (Haldane, 1919) such that

$$C = \frac{1}{2}\left\{1 - \exp\left(-2m\right)\right\}, \tag{5.36}$$

where m is measured in centimorgans. In genetics, one centimorgan (cM) is a unit for measuring genetic distance, defined such that recombination occurs at a rate of 0.01 per cM on average in a single generation. It is different from the base pair unit we used in the last section, which measures physical distance. The number of base pairs that one cM corresponds to is not constant, but depends on the positions of the target loci on their chromosomes and on other factors such as sex. It has been found that one cM corresponds to roughly one million bps on average for humans.

We define $m = \alpha d$ for this inference problem. Our interest lies in how the value of α varies along a chromosome. By equation (5.36), we have

$$\rho = 4NC = 2N\left\{1 - \exp\left(-2\alpha d\right)\right\}, \tag{5.37}$$

where N denotes the genetic effective population size (N_e), which is assumed to be known in this context. The inference problem now corresponds to estimating parameters α and θ using the proposed method.

For a single pair of loci that are distance d_0 apart, the true density of r^2 in terms of α and θ can be written as $\pi\left(r^2; d_0, \alpha, \theta\right)$, which corresponds to $\pi\left(r^2; \rho_0, \theta\right)$, where $\rho_0 = 2N\left\{1 - \exp\left(-2\alpha d_0\right)\right\}$. The procedure for finding maximum likelihood estimates of α and θ is almost the same as that for estimating ρ and θ described in the last section. Note that the requisite data are now of the form $\left(d, r^2\right)$, not just r^2. Suppose a sample of data is $\left\{\left(d_1, r_1^2\right), \left(d_2, r_2^2\right), \cdots, \left(d_K, r_K^2\right)\right\}$. The log-likelihood function is

$$l\left(\alpha, \theta\right) = \sum_{i=1}^{K} \log \pi\left(r_i^2; d_i, \alpha, \theta\right), \tag{5.38}$$

where for each d_i we may use a different number of moments to reconstruct the Maxent density of r^2, namely $\widetilde{\pi}_{k_i}\left(r^2; d_i, \alpha, \theta\right)$.

Computing time for this inference problem is highly dependent on the number of different distances among $(d_1, \ldots, d_K)$, because for a single evaluation of the log-likelihood function $l\left(\alpha, \theta\right)$, we need to construct a different Maxent density $\widetilde{\pi}_{k_i}\left(r^2; d_i, \alpha, \theta\right)$ for each distinct value of d_i. Note that the size of the sample within each value of d_i has negligible impact on computing time.

5.9 Conclusions and Closing Remarks

In this chapter, we first proposed a fast and general method to compute a finite sequence of moments of r^2 at stationarity by generalising the method of Song and Song (2007). Then we constructed the density function of r^2 from the moments obtained using the univariate Maxent approach. As far as we know, we are the first to achieve this. We demonstrated that the Maxent density of r^2 can be used to draw inference on mutation and recombination parameters from simulated or real data.

The Maxent approach performs well in both efficiency and accuracy for constructing the density of r^2 for specific valus of (ρ, θ). Applying the Maxent density to real data for estimating ρ and θ by maximum likelihood has been proved feasible, although the performance relies heavily on the size of samples. Simulation studies indicated that if sufficiently large samples are provided, the Maxent maximum likelihood approach can yield precise estimation results.

In practice, it is difficult to acquire sufficent data for inference on (ρ, θ). We proposed a different form of inference in Section 5.8 that imposes fewer restrictions on the data required. However, estimating (α, θ) becomes very slow when the sample data has a large number of different values of d, because each evaluation of the likelihood function involves constructing a density of r^2 for each distinct value of d. This problem could be addressed by using more computing power (say computer clusters) and parallel computing. Another issue for this model is that the genetic effective population size N needs to be known or estimated independently.

To summarise, we have provided a novel method for estimating mutation and recombination parameters in population genetics. We made assumptions that recombination is only influenced by the distance between a pair of loci, while mutation remains the same for all loci in the sample. These assumptions might be questionable in practice, but we have shown that the method itself has a promising performance if sufficiently many samples can be obtained that meet these requirements. Note that we are the first to offer this link from the distribution of r^2 to inference on real data. Our aim is to show how this can be done, rather than to provide a definitive data analysis.

Song and Song (2007) mentioned that although mutation is assumed to be symmetric and recurrent for the TLD model, their method of calculating $\mathbb{E}\left(r^2\right)$ can be generalised to models with different mutation structures. Song and Song (2007) also commented that their method might be generalised to genetic models with natural selection. Our method is based on that of Song and Song (2007), so it is likely that it too has wider applicability than the TLD model alone.

6

A Numerical Approach to the Kolmogorov Equation

6.1 Kolmogorov Forward Equation of the TLD Model

As mentioned in Chapter 4, each diffusion process has associated with it a Kolmogorov forward equation, which is typically a partial differential equation (PDE). The solution of the equation is the stationary distribution of the diffusion process. However, solving PDEs analytically is intractable in most cases. Alternative numerical methods for solving PDEs are widespread; see Boitard and Loisel (2007) for an example in population genetics. In this chapter, we consider a numerical approach to solving the Kolmogorov equation of the diffusion process corresponding to the TLD model.

Instead of the parameters (p, q, D) used in the last two chapters, in this chapter we use gametic frequencies $\boldsymbol{p} = (p_1, p_2, p_3)$ following Liu (2012). For convenience, we use π to replace $\pi(\boldsymbol{p})$, the stationary distribution of the TLD diffusion model. In addition, any function $g(\boldsymbol{p})$ with argument $\boldsymbol{p}$ will be replaced by g in this chapter.

The Kolmogorov forward equation of the TLD model was derived by Ewens (2004) and Liu (2012) in detail. We follow the work of Liu (2012), in which the Kolmogorov equation is written as

$$-\sum_{i=1}^{3} \frac{\partial}{\partial p_i} (M_i \pi) + \frac{1}{2} \sum_{i=1}^{3} \frac{\partial^2}{\partial p_i^2} (V_i \pi) + \sum_{i=1}^{2} \sum_{j>i}^{3} \frac{\partial^2}{\partial p_i \partial p_j} \left(W_{ij} \pi \right) = 0, \tag{6.1}$$

where

$$\begin{aligned} M_1 &= -2\theta p_1 + \theta p_2 + \theta p_3 - \rho D \\ M_2 &= \theta - 3\theta p_2 - \theta p_3 + \rho D \\ M_3 &= \theta - \theta p_2 - 3\theta p_3 + \rho D \\ D &= p_1 - p_1^2 - p_1 p_2 - p_1 p_3 - p_2 p_3 \\ V_i &= p_i (1 - p_i) \\ W_{ij} &= -p_i p_j. \end{aligned} \tag{6.2}$$

Note that the parameters ρ and θ used here differ from those used in Song and Song (2007) by factors of 1/4 and 1/8 respectively because of the different scales adopted in deriving the diffusion processes. For example, $(\rho, \theta) = (5, 1.25)$ in this chapter is equivalent to $(20, 10)$ in the last two chapters.

The Kolmogorov equation (6.1) can be expanded as

$$\begin{aligned} 0 \;=\; & \frac{1}{2}\sum_{i=1}^{3} V_i \frac{\partial^2 \pi}{\partial p_i^2} - \sum_{i=1}^{2}\sum_{j>i}^{3} p_i p_j \frac{\partial^2 \pi}{\partial p_i \partial p_j} + \sum_{i=1}^{3} \left(1 - 4p_i - M_i\right) \frac{\partial \pi}{\partial p_i} \\ & + (8\theta + \rho - 6)\pi, \end{aligned} \tag{6.3}$$

which is a three-dimensional PDE with variable coefficients. More details regarding this calculation can be found in Appendix D.

6.2 A Finite Difference Method

Finite difference methods (FDMs) are numerical methods for solving differential equations (ordinary or partial) in computational mathematics. The central idea of FDMs is to discretize the domain of objective functions and then approximate derivatives using finite differences. A system of difference equations can then be generated. Solving the system of equations gives approximations to the true values of the objective function evaluated at a series of discrete points of its domain. See Iserles (2009) and Smith (1985) for a comprehensive review of FDMs.

To illustrate the procedure of applying FDMs, we consider a simple example. Suppose $f(x)$ is a differentiable function defined on $[0, 1]$ and the value of $f(0)$ is known, say $f(0) = 1$. We consider a FDM to solve the ordinary differential equation

$$f'(x) = f(x) - \frac{2x}{f(x)}. \tag{6.4}$$

First, we select a set of discrete points $x_i = x_0 + ih, i = 1, 2, \ldots, n$ from $[0, 1]$ where $x_0 = 0$. Note that the step size $h = x_{i+1} - x_i$ should be sufficiently small to ensure the accuracy of the method. The first-order derivative $f'(x_i)$ is approximated by the backward difference:

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{h}. \tag{6.5}$$

Substituting this into (6.4) yields

$$f\left(x_i\right)=f\left(x_{i-1}\right)+hf\left(x_i\right)-\frac{2hx_i}{f\left(x_i\right)},\quad i=1,2,\ldots,n. \tag{6.6}$$

Solving the n difference equations in (6.6) gives approximate values of $f\left(x_i\right)$ for $i=1,\ldots,n$.

Note that to apply the FDM, it is necessary to know the value of $f\left(x_0\right)$, which is called the boundary condition; otherwise the system of difference equations does not have a unique solution. In some contexts, finding boundary conditions is a considerable challenge.

For our problem, the domain of the density function π is

$$\Omega=\left\{\left(p_1,p_2,p_3\right)\in\mathbb{R}^3 \mid p_1+p_2+p_3\leq 1;\ p_1,p_2,p_3\geq 0\right\}. \tag{6.7}$$

We use the same step size $h=1/n$ for all three dimensions to discretize the domain Ω, where $n\in\mathbb{N}$ is a sufficiently large integer. Then our FDM considers the values of π evaluated at points $\left(p_{1i},p_{2j},p_{3k}\right)=(ih,jh,kh)$, where i,j,k are non-negative integers satisfying $i+j+k\leq n$. For clarity, we use π_{ijk} to replace $\pi\left(p_{1i},p_{2j},p_{3k}\right)$, in other words, the value of the function π evaluated at point $\left(p_{1i},p_{2j},p_{3k}\right)$.

An explicit Euler-centered difference scheme (Iserles, 2009) is used to approximate both the first-order and second-order derivatives of π. Suppose each derivative is evaluated at the point $\Big(p_{1i}, p_{2j}, p_{3k}\Big)$. Then we have the results below:

$$\begin{aligned}
\frac{\partial \pi}{\partial p_1} &\approx \frac{\pi_{i+1jk} - \pi_{i-1jk}}{2h} \\
\frac{\partial \pi}{\partial p_2} &\approx \frac{\pi_{ij+1k} - \pi_{ij-1k}}{2h} \\
\frac{\partial \pi}{\partial p_3} &\approx \frac{\pi_{ijk+1} - \pi_{ijk-1}}{2h} \\
\frac{\partial^2 \pi}{\partial p_1^2} &\approx \frac{\pi_{i+1jk} - 2\pi_{ijk} + \pi_{i-1jk}}{h^2} \\
\frac{\partial^2 \pi}{\partial p_2^2} &\approx \frac{\pi_{ij+1k} - 2\pi_{ijk} + \pi_{ij-1k}}{h^2} \\
\frac{\partial^2 \pi}{\partial p_3^2} &\approx \frac{\pi_{ijk+1} - 2\pi_{ijk} + \pi_{ijk-1}}{h^2} \\
\frac{\partial^2 \pi}{\partial p_1 \partial p_2} &\approx \frac{\pi_{i+1j+1k} - \pi_{i+1j-1k} - \pi_{i-1j+1k} + \pi_{i-1j-1k}}{4h^2} \\
\frac{\partial^2 \pi}{\partial p_1 \partial p_3} &\approx \frac{\pi_{i+1jk+1} - \pi_{i+1jk-1} - \pi_{i-1jk+1} + \pi_{i-1jk-1}}{4h^2} \\
\frac{\partial^2 \pi}{\partial p_2 \partial p_3} &\approx \frac{\pi_{ij+1k+1} - \pi_{ij+1k-1} - \pi_{ij-1k+1} + \pi_{ij-1k-1}}{4h^2}.
\end{aligned} \tag{6.8}$$

Substituting these finite differences into equation (6.3) yields a system of difference equations:

$$\begin{aligned}
0 = &\left\{2V_1 + 2h\left(1 - 4p_1 - M_1\right)\right\}\pi_{i+1jk} - \left\{2V_1 - 2h\left(1 - 4p_1 - M_1\right)\right\}\pi_{i-1jk} \\
&+ \left\{2V_2 + 2h\left(1 - 4p_2 - M_2\right)\right\}\pi_{ij+1k} - \left\{2V_2 - 2h\left(1 - 4p_2 - M_2\right)\right\}\pi_{ij-1k} \\
&+ \left\{2V_3 + 2h\left(1 - 4p_3 - M_3\right)\right\}\pi_{ijk+1} - \left\{2V_3 - 2h\left(1 - 4p_3 - M_3\right)\right\}\pi_{ijk-1} \\
&- p_1p_2\Big(\pi_{i+1j+1k} - \pi_{i+1j-1k} - \pi_{i-1j+1k} + \pi_{i-1j-1k}\Big) \\
&- p_1p_3\Big(\pi_{i+1jk+1} - \pi_{i+1jk-1} - \pi_{i-1jk+1} + \pi_{i-1jk-1}\Big) \\
&- p_2p_3\Big(\pi_{ij+1k+1} - \pi_{ij+1k-1} - \pi_{ij-1k+1} + \pi_{ij-1k-1}\Big) \\
&- 4\Big\{V_1 + V_2 + V_3 - h^2\left(8\theta + \rho - 6\right)\Big\}\pi_{ijk},
\end{aligned} \tag{6.9}$$

where all coefficients in these equations are evaluated at point $\Big(p_{1i}, p_{2j}, p_{3k}\Big)$.

Table 6.1 Comparison of π evaluated at various points of Δ by the FDM and the Maxent approach of Liu (2012) for $(\rho, \theta) = (5, 1.25)$. The mean relative difference given by $|\text{FDM} - \text{Maxent}|/\text{Maxent} \times 100\%$ for all points in Δ is 1.03%.

Point ($\times 10^{-5}$)	FDM	Maxent	Relative Difference (%)
(9.5, 9.5, 2.0)	8.0056	8.0110	0.067
(9.5, 9.5, 2.5)	8.0013	8.0067	0.068
(9.5, 9.5, 3.0)	7.9970	8.0025	0.069
(9.5, 9.5, 3.5)	7.9927	7.9983	0.070
(9.5, 9.5, 4.0)	7.9885	7.9941	0.070
(9.5, 9.5, 4.5)	7.9843	7.9889	0.070
(9.5, 9.5, 5.0)	7.9801	7.9857	0.070
(9.5, 9.5, 5.5)	7.9760	7.9815	0.069
(9.5, 9.5, 6.0)	7.9719	7.9773	0.068
(9.5, 9.5, 6.5)	7.9678	7.9731	0.066

6.3 Preliminary Results

To apply the FDM described above, we need to find boundary conditions for the TLD problem, namely the values of π evaluated at the boundary points of the domain Ω. Unfortunately, it is not clear how to do this without using an existing solution for π, which defeats the objective of using this method to gain a new, independent approximation to π. The only existing formulation of $\pi(p_1, p_2, p_3)$ is due to Liu (2012). Thus, we use Liu (2012)'s approximation to $\pi(p_1, p_2, p_3)$ to calculate the boundary conditions required to enable us to solve the finite difference equation (6.9).

The value of n must be sufficiently large to ensure the accuracy of the FDM, so considerable computing time and power is needed, especially because π can have a very spiked shape. Here, we only test the method on a subset of the domain Ω to check its performance. Specifically, we consider a cube

$$\Delta = [0, 0.0001] \times [0, 0.0001] \times [0, 0.0001], \tag{6.10}$$

which is a subset of the domain Ω. We use $n = 20$ in this example to evenly discretize each dimension of Δ. Values of π on the boundary of Δ are evaluated using Liu

(2012)'s Maxent approach, while values of π evaluated at the interior points of Δ are computed using both the FDM and Liu (2012)'s approach.

A comparison of the results from the two approaches is shown in Table 6.1 for selected interior points. The two methods give very similar interior values for π, which presents some evidence of the validity of each calculation. However, the FDM does not offer a satisfactory standalone approach for constructing π, due to the unknown boundary conditions and enormous computational demands.

7

Conclusions and Future Work

7.1 Conclusions

In this thesis, we have demonstrated the power of probability density approximation for parameter estimation using maximum likelihood, by investigating two specific problems of real interest arising in very different contexts. Two distinct approximation techniques, the saddlepoint approximation and the maximum entropy principle, as well as the diffusion approximation, have been applied to solve the two problems.

In the first project, we derived a general approximate likelihood function for latent multinomial models using the saddlepoint approximation method. The accuracy of the proposed method for parameter estimation was demonstrated by applying it to model $M_{t,\alpha}$, whose exact likelihood function is available for comparison. We further compared the method with the hybrid approximation method for the two-source model. Point estimates and 95% confidence intervals of model parameters obtained by the two methods were consistently close to one another. We then applied the method to another two latent multinomial problems in the contexts of multi-list studies and multi-way contingency tables. The performance of the method was illustrated by simulation studies and real data examples.

In the second project, we constructed the density function of r^2 using the univariate Maxent approach from a series of its moments, which can be obtained under the diffusion approximation without knowing the distribution of r^2. As a byproduct, we created a fast, analytic computation of the variance of r^2, which was previously only available via an extremely lengthy computation derived by Liu (2012). We then used this Maxent density of r^2 to estimate the evolutionary parameters ρ and θ from sample observations of r^2. The performance of this method was illustrated by simulation studies. We also applied the method to analysis of real data from the 1000 Genomes Project. The precision of the maximum likelihood estimates was highly dependent on the size of the sample; a larger sample generated narrower confidence intervals for both parameters.

7.2 Future Work

Based on the work of this thesis, there are various projects that could be undertaken in the future. We briefly describe some ideas here. For convenience, we use $\boldsymbol{\theta}$ here to denote the parameter vector of a model, noting that we previously used $(N, \boldsymbol{\theta})$ for this purpose when studying latent multinomial models (LMMs).

In the first project, we showed that the saddlepoint approximation to the PMF of model $M_{t,\alpha}$ does not have a good match to the true mass function in most cases (see Fig. 3.3). This might be due to a missing normalisation constant for the saddlepoint mass function $\tilde{f}_{\boldsymbol{X}}(\boldsymbol{x})$ as shown in equation (3.14), in which case $\tilde{f}_{\boldsymbol{X}}(\boldsymbol{x})$ is not strictly a valid probability density function because

$$c = \sum_{\boldsymbol{x} \in \mathcal{X}} \tilde{f}_{\boldsymbol{X}}(\boldsymbol{x}) \neq 1, \tag{7.1}$$

where $\mathcal{X}$ denotes the support of $\boldsymbol{X}$. The normalised density $\tilde{f}_{\boldsymbol{X}}(\boldsymbol{x})/c$ might provide a better match with the true PMF; however, calculating (7.1) is not straightforward as far as we know. We showed that using the unnormalised density $\tilde{f}_{\boldsymbol{X}}(\boldsymbol{x})$ performs extremely well for parameter estimation in LMMs, but we do not know if the excellent performance of the saddlepoint method can be assumed to apply in other modelling contexts. While the saddlepoint is known to be a powerful technique for density approximation, its performance for maximum likelihood estimation depends upon maintaining uniformly good performance throughout a subspace of the parameter space close to the parameter estimates $\hat{\boldsymbol{\theta}}$, or at least for any approximation errors to be negligible with respect to the location and curvature of the minimum negative log-likelihood. This appeared to be the case for model $M_{t,\alpha}$ based on Fig. 3.3: although the height of $-\log \widetilde{\mathcal{L}}(\boldsymbol{\theta})$ was not accurate, the minimum and curvature were accurate so $\hat{\boldsymbol{\theta}}$ and $\widehat{\text{var}}\left(\hat{\boldsymbol{\theta}}\right)$ were very accurate.

It is possible that problems might occur for other models, rendering the saddlepoint approximation unsuitable for estimation purposes, especially if the error term depends on the parameters $\boldsymbol{\theta}$ over which the likelihood is to be maximised. A possible solution is to derive a second-order saddlepoint mass function for LMMs or other models of interest. See Sections 3.4.4 and 3.4.5 of Butler (2007) for an introduction to high-dimensional saddlepoint approximations. It is shown in Butler (2007) that the normal-based saddlepoint approximation that we used can be significantly biased when approximating a multinomial mass function, but that a second-order saddlepoint approximation greatly improves the performance of the method. Deriving the exact form of the second-order saddlepoint approximation to the PMF of LMMs is not straightforward, but the second-order approach serves as a possibility that is worth exploring if needed.

In the genetics project, we assumed symmetric and reversible mutation for the TLD model, but our method of calculating moments of r^2 can be generalised naturally to models with different mutation patterns. The method should also be applicable to models incorporating selection as mentioned by Song and Song (2007). Once the moments of r^2 (or other random variables of interest) are obtained, the process of constructing the PDF using the Maxent principle is generally applicable to any model.

In Chapter 6, we presented some preliminary work for finding a numerical solution to the Kolmogorov equation of the TLD model. This is the most direct way to investigate the stationary distribution of the model, and was indeed the starting point of this PhD project. However, we did not explore it further because our aim was to use the distribution for data analysis, which is computationally impracticable using this method. If the stationary distribution itself is of primary interest, the finite difference method is certainly a plausible option. Finding boundary conditions is the greatest challenge in applying the finite difference method to this problem. Boitard and Loisel (2007) proposed some approximation strategies to address the boundary problem for another two-locus model. It might be possible to develop similar methods for the TLD model.

Appendices

A

An Example of Likelihood Factorization for LMMs

We consider model $M_{t,\alpha}$ with $K = 2$ capture occasions as an example to illustrate the procedure of likelihood factorization described in Section 3.2.

We have shown in Chapter 2 the relationship $\boldsymbol{y} = T\boldsymbol{z}$ in this case is

$$\begin{bmatrix} y_{01} \\ y_{10} \\ y_{11} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_{00} \\ z_{01} \\ z_{02} \\ z_{10} \\ z_{11} \\ z_{12} \\ z_{20} \\ z_{21} \\ z_{22} \end{bmatrix} . \tag{A.1}$$

Suppose $\boldsymbol{y} = (y_{01}, y_{10}, y_{11}) = (0, 7, 1)$. From $y_{01} = 0$, we find that $z_{01} = z_{02} = z_{12} = z_{21} = z_{22} = 0$. In addition, $z_{11} = y_{11}$ because history 11 is fully-observed. In this case, the vector of verified elements is $\boldsymbol{v} = (z_{01}, z_{02}, z_{11}, z_{12}, z_{21}, z_{22}) = (0, 0, 1, 0, 0, 0)$, and the unverified vector is $\boldsymbol{u} = (z_{00}, z_{10}, z_{20})$.

Now, we reorder the elements of $\boldsymbol{z}$ and T as follows:

$$\begin{bmatrix} y_{01} \\ y_{10} \\ y_{11} \end{bmatrix} = \left[\begin{array}{cccccc:ccc} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} z_{01} \\ z_{02} \\ z_{11} \\ z_{12} \\ z_{21} \\ z_{22} \\ \hdashline z_{00} \\ z_{10} \\ z_{20} \end{array}\right] . \tag{A.2}$$

It follows that

$$B = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}, \tag{A.3}$$

and

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \tag{A.4}$$

Then we have

$$\begin{aligned} \boldsymbol{x} = \boldsymbol{y} - B\boldsymbol{v} &= \begin{bmatrix} 0 \\ 7 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 7 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 7 \\ 0 \end{bmatrix}. \end{aligned} \tag{A.5}$$

Here, $\boldsymbol{x}$ contains two zero elements, and matrix A contains two rows composed completely of zero entries. In practice, we need to remove the two zero components of $\boldsymbol{x}$ and the two zero rows of A to make the saddlepoint approximation work.

B

R Code for Implementing TMB to Fit LMMs

Here, we attach the key code for fitting LMMs via the `TMB` package. We first copy the source code of `TMB`. The required function for modification is function `h` inside the function `MakeADFun`. We only need to modify two lines of the original `h` function to obtain the correct formulation for our problem. Let `my.MakeADFun` be the name of the new `MakeADFun` function. We use `my.MakeADFun` to deliver the saddlepoint PMF for LMMs.

Below is the code of the new `h` function. The two lines that have been modified (with comments) can be found easily.

```
h <- function(theta=par, order=0, hessian, L, ...)
{
  if(order==0){
    ## When order==0, function h represents the negative
    ## log-likelihood function (3.29), which is the objective for minimisation.
    logdetH <- 2*determinant(L)$mod
    ## We modify the line below by changing the first sign to -
    ## and the third to +.
    ans <- -f(theta,order=0)+.5*logdetH+length(random)/2*log(2*pi)
    if(LaplaceNonZeroGradient){
      grad <- f(theta,order=1)[random]
      ans-.5*sum(grad*as.numeric(solve(L,grad)))
    } else
      ans
  }
  else if(order==1){
    ## When order==1, function h represents the gradient of
    ## the negative log-likelihood function (3.29).
    if(LaplaceNonZeroGradient)
      stop("Not correct for LaplaceNonZeroGradient=TRUE")
    e <- environment(spHess)
    solveSubset <- function(L).Call("tmb_invQ",L,PACKAGE="TMB")
    solveSubset2 <- function(L).Call("tmb_invQ_tril_halfdiag",L,PACKAGE="TMB")
    ihessian <- solveSubset2(L)
    ## Profile case correction (1st order case only)
    if(!is.null(profile)){
      perm <- L@perm+1L
      ihessian <- .Call("tmb_sparse_izamd", ihessian, profile[perm],
                        0.0, PACKAGE="TMB")
    }
    lookup <- function(A,B,r=NULL){
      A <- tril(A); B <- tril(B)
      B@x[] <- seq.int(length.out=length(B@x))
      if(!is.null(r)){
        B <- .Call("tmb_half_diag", B, PACKAGE="TMB")
        B <- tril(B[r,r,drop=FALSE])+tril(t(B)[r,r,drop=FALSE])
      }
      m <- .Call("match_pattern", A, B, PACKAGE="TMB")
      B@x[m]
    }
    if(is.null(e$ind1)){
      if (!silent) cat("Matching hessian patterns... ")
      iperm <- invPerm(L@perm+1L)
      e$ind1 <- lookup(hessian,ihessian,iperm)
      e$ind2 <- lookup(hessian,e$Hfull,random)
      if (!silent) cat("Done\n")
    }
    w <- rep(0,length=length(e$Hfull@x))
    w[e$ind2] <- ihessian@x[e$ind1]

    ## We modify the line below by changing the sign of
    ## as.vector(f(theta,order=1)) to -.
    (-1) * as.vector(f(theta,order=1))+
    .Call("EvalADFunObject", e$ADHess$ptr, theta,
            control=list(
              order=as.integer(1),
              hessiancols=as.integer(0),
              hessianrows=as.integer(0),
              sparsitypattern=as.integer(0),
              rangecomponent=as.integer(1),
              rangeweight=as.double(w),
              dumpstack=as.integer(0),
              doforward=as.integer(1)
            ),
            PACKAGE=DLL)
  }## order==1
  else stop(sprintf("'order'=%d not yet implemented", order))
} ## end{h}
```

C

Introduction to VCFtools

Implementing VCFtools to extract sample data of r^2 from original genotype data from the 1000 Genomes Project is straightforward. We use the one-line command below.

```
./vcftools --gzvcf chr21.vcf.gz --hap-r2  --maf 0.05 --keep AMR
                                   --ld-window-bp 100000 --out outfile
```

Here, chr21.vcf.gz is the name of the VCF file containing data for human chromosome 21. This can be downloaded from ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/. The option --hap-r2 indicates that data on r^2 will be computed on the phased genotype data. Note that for these data we know

Table C.1 Sample output data from `VCFtools`. Here, POS1 and POS2 represent the positions of two loci on chromosome 21 (CHR). N_CHR represents the total number of chromosomes from the sample. It follows that there are 347 (694/2) individuals for the AMR sample. R.2 represents the observation of r^2. D and Dprime are not needed for our analysis.

CHR	POS1	POS2	N_CHR	R.2	D	Dprime
21	9411410	9411500	694	0.16455	0.10129	0.40799
21	9411410	9411602	694	0.00848	−0.02114	−0.14430
21	9411410	9411645	694	0.17882	0.10506	0.45596
⋮	⋮	⋮	⋮	⋮	⋮	⋮

how genotypes are organised into haplotypes for each person in the sample: in other words, we know the gamete-type frequencies. The option `--maf 0.05` sets the minor allele frequency to be at least 0.05. Option `--keep AMR` indicates that the computation is focused on individuals from population AMR (Americans). `AMR` is the name of a text file we create that contains IDs of individuals from the AMR population. See the website `ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/20130502/integrated_call_samples_v3.20130502.ALL.panel` for information on the population each individual belongs to. Option `--ld-window-bp 100000` indicates that we consider pairs of loci with spacing less than or equal to 100,000 base pairs. `--out outfile` specifies the output data file.

Importing the output data file into `R`, we obtain a large table, as shown in Table C.1. We extract data of r^2 from the table after setting values for parameters $(d, \Delta d, l)$. Consider $(d, \Delta d, l) = (5000, 100, 10000)$ for an example. The first row of Table C.1 gives an observed value 0.16455 of r^2 for two loci that are 90 (POS2 − POS1) base pairs apart. Thus, we do not use this observation since 90 is not in the interval $(4900, 5100)$. Let $\left\{r_1^2, \ldots, r_n^2\right\}$ denote all observations of r^2 for pairs of loci that have within-pair distances between 4900 and 5100 base pairs. Each of the observations corresponds to two loci. We write simple code to find a set of observations for which all loci are located in a zone with a length of $l = 10,000$ base pairs on the chromosome.

D

Simplification of the Kolmogorov Equation of the TLD Model

Here, we present the calculation for simplifying the Kolmogorov forward equation (6.1) of the TLD model.

First, we have

$$\begin{aligned} -\sum_{i=1}^{3} \frac{\partial}{\partial p_i} \left(M_i \, \pi \right) &= -\sum_{i=1}^{3} \left(\frac{\partial M_i}{\partial p_i} \pi + \frac{\partial \pi}{\partial p_i} M_i \right) \\ &= \left(8\theta + \rho \right) \pi - \sum_{i=1}^{3} \frac{\partial \pi}{\partial p_i} M_i . \end{aligned} \tag{D.1}$$

In addition, we have

$$\begin{aligned}\frac{1}{2}\sum_{i=1}^{3}\frac{\partial^2}{\partial p_i^2}\left(V_i\,\pi\right) &= \frac{1}{2}\sum_{i=1}^{3}\left(\frac{\partial^2 V_i}{\partial p_i^2}\pi + 2\frac{\partial V_i}{\partial p_i}\frac{\partial \pi}{\partial p_i} + \frac{\partial^2 \pi}{\partial p_i^2}V_i\right)\\ &= -3\pi + \sum_{i=1}^{3}\frac{\partial \pi}{\partial p_i}\left(1-2p_i\right) + \frac{1}{2}\sum_{i=1}^{3}\frac{\partial^2 \pi}{\partial p_i^2}V_i\end{aligned} \tag{D.2}$$

and

$$\begin{aligned}&\sum_{i=1}^{2}\sum_{j>i}^{3}\frac{\partial^2}{\partial p_i \partial p_j}\left(W_{ij}\,\pi\right)\\ =&\sum_{i=1}^{2}\sum_{j>i}^{3}\left(\pi\frac{\partial^2 W_{ij}}{\partial p_i \partial p_j} + \frac{\partial W_{ij}}{\partial p_i}\frac{\partial \pi}{\partial p_j} + \frac{\partial W_{ij}}{\partial p_j}\frac{\partial \pi}{\partial p_i} + W_{ij}\frac{\partial^2 \pi}{\partial p_i \partial p_j}\right)\\ =&-3\pi - 2\sum_{i=1}^{3}p_i\frac{\partial \pi}{\partial p_i} - \sum_{i=1}^{2}\sum_{j>i}^{3}p_i p_j\frac{\partial^2 \pi}{\partial p_i \partial p_j}.\end{aligned} \tag{D.3}$$

Substituting equations (D.1), (D.2), and (D.3) into (6.1) yields equation (6.3).

Bibliography

1000 Genomes Project Consortium (2010). A map of human genome variation from population scale sequencing. *Nature* **467,** 1061–1073.

1000 Genomes Project Consortium (2012). An integrated map of genetic variation from 1,092 human genomes. *Nature* **491,** 56–65.

1000 Genomes Project Consortium (2015). A global reference for human genetic variation. *Nature* **526,** 68–74.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control* **19,** 716–723.

Auton, A., Myers, S., and McVean, G. (2014). Identifying recombination hotspots using population genetic data. *arXiv preprint arXiv:1403.4264* .

Banavar, J. R., Maritan, A., and Volkov, I. (2010). Applications of the principle of maximum entropy: from physics to ecology. *Journal of Physics: Condensed Matter* **22,** 063101.

Bandyopadhyay, K., Bhattacharya, A., Biswas, P., and Drabold, D. (2005). Maximum entropy and the problem of moments: A stable algorithm. *Physical Review E* **71,** 057701.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational Linguistics* **22,** 39–71.

Biswas, P. and Bhattacharya, A. (2010). Function reconstruction as a classical moment problem: a maximum entropy approach. *Journal of Physics A: Mathematical and Theoretical* **43,** 405003.

Boitard, S. and Loisel, P. (2007). Probability distribution of haplotype frequencies under the two-locus Wright-Fisher model by diffusion approximation. *Theoretical Population Biology* **71,** 380–391.

Bonner, S. J. and Holmberg, J. (2013). Mark-recapture with multiple, non-invasive marks. *Biometrics* **69,** 766–775.

Bonner, S. J., Schofield, M. R., Noren, P., and Price, S. J. (2016). Extending the latent multinomial model with complex error processes and dynamic Markov bases. *The Annals of Applied Statistics* **10,** 246–263.

Burnham, K. P. and Anderson, D. R. (2003). *Model selection and multimodel inference: a practical information-theoretic approach.* Springer Science & Business Media.

Butler, R. W. (2007). *Saddlepoint approximations with applications.* Cambridge University Press.

Carroll, E. L., Patenaude, N. J., Childerhouse, S. J., Kraus, S. D., Fewster, R. M., and Baker, C. S. (2011). Abundance of the New Zealand subantarctic southern right whale population estimated from photo-identification and genotype mark-recapture. *Marine Biology* **158,** 2565.

Chao, A. (2001). An overview of closed capture-recapture models. *Journal of Agricultural, Biological, and Environmental Statistics* **6,** 158–175.

Chen, Y., Diaconis, P., Holmes, S. P., and Liu, J. S. (2005). Sequential Monte Carlo methods for statistical analysis of tables. *Journal of the American Statistical Association* **100,** 109–120.

Chen, Y., Dinwoodie, I. H., and Sullivant, S. (2006). Sequential importance sampling for multiway tables. *The Annals of Statistics* **34,** 523–545.

Clason, C. and von Winckel, G. (2012). A general spectral method for the numerical simulation of one-dimensional interacting fermions. *Computer Physics Communications* **183,** 405–417.

Cormack, R. M. (1979). Models for capture-recapture. In *Sampling Biological Populations*, volume 5, pages 217–255. International Co-operative Publishing House, Montpellier, France.

Cormack, R. M. (1989). Log-linear models for capture-recapture. *Biometrics* **45,** 395–413.

Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory.* John Wiley & Sons.

Danecek, P., Auton, A., Abecasis, G., Albers, C. A., Banks, E., DePristo, M. A., Handsaker, R. E., Lunter, G., Marth, G. T., Sherry, S. T., et al. (2011). The variant call format and VCFtools. *Bioinformatics* **27,** 2156–2158.

Daniels, H. E. (1954). Saddlepoint approximations in statistics. *The Annals of Mathematical Statistics* **25,** 631–650.

Darroch, J. N. (1958). The multiple-recapture census: I. Estimation of a closed population. *Biometrika* **45,** 343–359.

Darroch, J. N. (1959). The multiple-recapture census: II. Estimation when there is immigration or death. *Biometrika* **46,** 336–351.

Davison, A. C. and Hinkley, D. V. (1988). Saddlepoint approximations in resampling methods. *Biometrika* **75,** 417–431.

Diaconis, P. and Sturmfels, B. (1998). Algebraic algorithms for sampling from conditional distributions. *The Annals of Statistics* **26,** 363–397.

DiCiccio, T. J., Martin, M. A., and Young, G. A. (1992). Fast and accurate approximate double bootstrap confidence intervals. *Biometrika* **79,** 285–295.

Dobra, A. (2012). Dynamic Markov bases. *Journal of Computational and Graphical Statistics* **21,** 496–517.

Dobra, A., Tebaldi, C., and West, M. (2006). Data augmentation in multi-way contingency tables with fixed marginal totals. *Journal of Statistical Planning and Inference* **136,** 355–372.

Doob, J. L. (1942). The Brownian movement and stochastic equations. *Annals of Mathematics* **2,** 351–369.

Edwards, D. and Toma, H. (1985). A fast procedure for model search in multidimensional contingency tables. *Biometrika* **72,** 339–351.

Ewens, W. (2004). *Mathematical population genetics: theoretical introduction*, volume 1. Springer Verlag.

Fewster, R. M., Zhang, W., Jupp, P. E., and Madon, B. (in prep.). Likelihood approximations for two-source capture-recapture models for open and closed populations.

Fisher, R. et al. (1922). On the dominance ratio. *Proceedings of the Royal Society of Edinburgh* **42,** 321–341.

Fisher, R. A. (1999). *The genetical theory of natural selection: a complete variorum edition.* Oxford University Press.

Fletcher, R. (1987). *Practical methods of optimization, Volume 1.* Wiley.

Fournier, D. A., Skaug, H. J., Ancheta, J., Ianelli, J., Magnusson, A., Maunder, M. N., Nielsen, A., and Sibert, J. (2012). AD Model Builder: using automatic differentiation for statistical inference of highly parameterized complex nonlinear models. *Optimization Methods and Software* **27,** 233–249.

Geye, C. J. (2015). *trust: Trust Region Optimization.* R package version 0.1-7.

Gilks, W. R. and Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* **41,** 337–348.

Glynn, P. W. (1990). Diffusion approximations. *Handbooks in Operations Research and Management Science* **2,** 145–198.

Golding, G. (1984). The sampling distribution of linkage disequilibrium. *Genetics* **108,** 257–274.

Goutis, C. and Casella, G. (1999). Explaining the saddlepoint approximation. *The American Statistician* **53,** 216–224.

Griewank, A. and Walther, A. (2008). *Evaluating derivatives: principles and techniques of algorithmic differentiation.* Society for Industrial and Applied Mathematics (SIAM).

Gupta, P. K., Rustgi, S., and Kulwal, P. L. (2005). Linkage disequilibrium and association studies in higher plants: present status and future prospects. *Plant Molecular Biology* **57,** 461–485.

Haldane, J. B. S. (1919). The combination of linkage values and the calculation of distances between the loci of linked factors. *Journal of Genetics* **8,** 299–309.

Higgs, M. D., Link, W. A., White, G. C., Haroldson, M. A., and Bjornlie, D. D. (2013). Insights into the latent multinomial model through mark-resight data on female grizzly bears with cubs-of-the-year. *Journal of Agricultural, Biological, and Environmental Statistics* **18,** 556–577.

Hildebrand, F. B. (1987). *Introduction to numerical analysis.* Courier Corporation.

Hill, W. G. and Robertson, A. (1968). Linkage disequilibrium in finite populations. *Theoretical and Applied Genetics* **38,** 226–231.

Horikawa, Y., Oda, N., Cox, N. J., Li, X., Orho-Melander, M., Hara, M., Hinokio, Y., Lindner, T. H., Mashima, H., Schwarz, P. E., et al. (2000). Genetic variation in the gene encoding calpain-10 is associated with type 2 diabetes mellitus. *Nature Genetics* **26,** 163.

Huakau, J. T. (2002). *New methods for analysis of epidemiological data using capture-recapture methods.* PhD thesis, The University of Auckland.

Hudson, R. R. (1985). The sampling distribution of linkage disequilibrium under an infinite allele model without selection. *Genetics* **109,** 611–631.

Iserles, A. (2009). *A first course in the numerical analysis of differential equations.* Cambridge University Press.

Jaynes, E. T. (1957a). Information theory and statistical mechanics. *Physical Review. Series II* **106,** 620–630.

Jaynes, E. T. (1957b). Information theory and statistical mechanics II. *Physical Review. Series II* **108,** 171–190.

Jaynes, E. T. (1982). On the rationale of maximum-entropy methods. *Proceedings of the IEEE* **70,** 939–952.

Jeffreys, A. J., Kauppi, L., and Neumann, R. (2001). Intensely punctate meiotic recombination in the class II region of the major histocompatibility complex. *Nature Genetics* **29,** 217–222.

Jorde, L. B. (2000). Linkage disequilibrium and the search for complex disease genes. *Genome Research* **10,** 1435–1444.

Karanth, K. U., Nichols, J. D., Kumar, N., and Hines, J. E. (2006). Assessing tiger population dynamics using photographic capture-recapture sampling. *Ecology* **87,** 2925–2937.

Kimura, M. (1964). Diffusion models in population genetics. *Journal of Applied Probability* **1,** 177–232.

King, R., Bird, S. M., Hay, G., and Hutchinson, S. J. (2009). Estimating current injectors in Scotland and their drug-related death rate by sex, region and age-group via Bayesian capture-recapture methods. *Statistical Methods in Medical Research* **18,** 341–359.

Kristensen, K., Nielsen, A., Berg, C. W., Skaug, H. J., and Bell, B. M. (2016). TMB: Automatic differentiation and Laplace approximation. *Journal of Statistical Software* **70,** 1–21.

Lee, A. (2002). Effect of list errors on the estimation of population size. *Biometrics* **58,** 185–191.

Lewontin, R. C. (1964). The interaction of selection and linkage. I. General considerations; heterotic models. *Genetics* **49,** 49.

Lewontin, R. C. and Kojima, K. (1960). The evolutionary dynamics of complex polymorphisms. *Evolution* **14,** 458–472.

Li, N. and Stephens, M. (2003). Modeling linkage disequilibrium and identifying recombination hotspots using single-nucleotide polymorphism data. *Genetics* **165,** 2213–2233.

Link, W. A. and Barker, R. J. (2005). Modeling association among demographic parameters in analysis of open population capture-recapture data. *Biometrics* **61,** 46–54.

Link, W. A., Yoshizaki, J., Bailey, L. L., and Pollock, K. H. (2010). Uncovering a latent multinomial: Analysis of mark-recapture data with misidentification. *Biometrics* **66,** 178–185.

Liu, J. (2012). *Reconstruction of probability distributions in population genetics.* PhD thesis, The University of Auckland.

Lugannani, R. and Rice, S. (1980). Saddlepoint approximation for the distribution of the sum of independent random variables. *Advances in Applied Probability* **12,** 475–490.

Lukacs, P. M. and Burnham, K. P. (2005). Estimating population size from DNA-based closed capture-recapture data incorporating genotyping error. *Journal of Wildlife Management* **69,** 396–403.

McClintock, B. T. (2015). multimark: an R package for analysis of capture-recapture data consisting of multiple "noninvasive" marks. *Ecology and Evolution* **5,** 4920–4931.

McClintock, B. T., Conn, P. B., Alonso, R. S., and Crooks, K. R. (2013). Integrated modeling of bilateral photo-identification data in mark-recapture analyses. *Ecology* **94,** 1464–1471.

McCrea, R. S. and Morgan, B. J. (2014). *Analysis of capture-recapture data.* CRC Press.

Morrison, T. A., Yoshizaki, J., Nichols, J. D., and Bolger, D. T. (2011). Estimating survival in photographic capture-recapture studies: Overcoming misidentification error. *Methods in Ecology and Evolution* **2,** 454–463.

Mueller, J. C. (2004). Linkage disequilibrium for different scales and applications. *Briefings in Bioinformatics* **5,** 355–364.

Ohta, T. and Kimura, M. (1969a). Linkage disequilibrium due to random genetic drift. *Genetics Research* **13,** 47–55.

Ohta, T. and Kimura, M. (1969b). Linkage disequilibrium at steady state determined by random genetic drift and recurrent mutation. *Genetics* **63,** 229–238.

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.

Pan, W. (2001). Akaike's information criterion in generalized estimating equations. *Biometrics* **57,** 120–125.

Phillips, S. J., Anderson, R. P., and Schapire, R. E. (2006). Maximum entropy modeling of species geographic distributions. *Ecological Modelling* **190,** 231–259.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). CODA: convergence diagnosis and output analysis for MCMC. *R News* **6,** 7–11.

Pollock, K. H. (2000). Capture-recapture models. *Journal of the American Statistical Association* **95,** 293–296.

Pritchard, J. K. and Przeworski, M. (2001). Linkage disequilibrium in humans: models and data. *The American Journal of Human Genetics* **69,** 1–14.

Pritchard, J. K. and Rosenberg, N. A. (1999). Use of unlinked genetic markers to detect population stratification in association studies. *The American Journal of Human Genetics* **65,** 220–228.

Reid, N. (1988). Saddlepoint methods and statistical inference. *Statistical Science* **3,** 213–227.

Robert, R. (1991). A maximum-entropy principle for two-dimensional perfect fluid dynamics. *Journal of Statistical Physics* **65,** 531–553.

Schofield, M. R. and Bonner, S. J. (2015). Connecting the latent multinomial. *Biometrics* **71,** 1070–1080.

Shannon, C. E. (1948a). A mathematical theory of communication I. *The Bell System Technical Journal* **27,** 379–423.

Shannon, C. E. (1948b). A mathematical theory of communication II. *The Bell System Technical Journal* **27,** 623–656.

Silver, R. and Röder, H. (1997). Calculation of densities of states and spectral functions by Chebyshev recursion and maximum entropy. *Physical Review E* **56,** 4822–4829.

Skaug, H. J. and Fournier, D. A. (2006). Automatic approximation of the marginal likelihood in non-Gaussian hierarchical models. *Computational Statistics & Data Analysis* **51,** 699–709.

Slatkin, M. (2008). Linkage disequilibrium–understanding the evolutionary past and mapping the medical future. *Nature Reviews Genetics* **9,** 477–485.

Smith, G. D. (1985). *Numerical solution of partial differential equations: finite difference methods.* Oxford University Press.

Song, Y. S. and Song, J. S. (2007). Analytic computation of the expectation of the linkage disequilibrium coefficient r^2. *Theoretical Population Biology* **71,** 49–60.

Strawderman, R. L., Casella, G., and Wells, M. T. (1996). Practical small-sample asymptotics for regression problems. *Journal of the American Statistical Association* **91,** 643–654.

Sutherland, J. (2003). *Multi-list methods in closed populations with stratified or incomplete information.* PhD thesis, Simon Fraser University.

Sutherland, J. and Schwarz, C. J. (2005). Multi-list methods using incomplete lists in closed populations. *Biometrics* **61,** 134–140.

Tishkoff, S. A., Dietzsch, E., Speed, W., Pakstis, A. J., Kidd, J. R., Cheung, K., Bonne-Tamir, B., Santachiara-Benerecetti, A. S., Moral, P., Krings, M., et al. (1996). Global patterns of linkage disequilibrium at the CD4 locus and modern human origins. *Science* **271,** 1380–1387.

Vale, R. T. R., Fewster, R. M., Carroll, E. L., and Patenaude, N. J. (2014). Maximum likelihood estimation for model $M_{t,\alpha}$ for capture-recapture data with misidentification. *Biometrics* **70,** 962–971.

Wang, S. (1993). Saddlepoint expansions in finite population problems. *Biometrika* **80,** 583–590.

Watterson, G. (1996). Motoo Kimura's use of diffusion theory in population genetics. *Theoretical Population Biology* **49,** 154–188.

Wheeler, J., Prais, M., and Blumstein, C. (1974). Analysis of spectral densities using modified moments. *Physical Review B* **10,** 2429–2447.

Wright, J. A., Barker, R. J., Schofield, M. R., Frantz, A. C., Byrom, A. E., and Gleeson, D. M. (2009). Incorporating genotype uncertainty into mark-recapture-type models for estimating abundance using DNA samples. *Biometrics* **65,** 833–840.

Wright, S. (1931). Evolution in Mendelian populations. *Genetics* **16,** 97–159.

Wright, S. (1937). The distribution of gene frequencies in populations. *Proceedings of the National Academy of Sciences of the United States of America* **23,** 307–320.

Wright, S. (1949). Adaptation and selection. In *Genetics, Paleontology and Evolution*, pages 365–389. Princeton University Press.

Wright, S. (1968). *Evolution and the genetics of populations: a treatise in four volumes.* University of Chicago Press.

Wright, S. J. and Nocedal, J. (1999). *Numerical optimization.* Springer.

Wu, X. (2003). Calculation of maximum entropy densities with application to income distribution. *Journal of Econometrics* **115,** 347–354.

Yoshizaki, J., Brownie, C., Pollock, K. H., and Link, W. A. (2011). Modeling misidentification errors that result from use of genetic tags in capture-recapture studies. *Environmental and Ecological Statistics* **18,** 27–55.

Yoshizaki, J., Pollock, K. H., Brownie, C., and Webster, R. A. (2009). Modeling misidentification errors in capture-recapture studies using photographic identification of evolving marks. *Ecology* **90,** 3–9.

Zellner, A. and Highfield, R. (1988). Calculation of maximum entropy distributions and approximation of marginal posterior distributions. *Journal of Econometrics* **37,** 195–209.