Approximate Marginal Likelihoods for Shrinkage Parameter Estimation in Penalized Logistic Regression Analysis of Case-Control Data

by

Siyuan Chen

B.Sc., Xi'an Jiaotong University, 2018

Project Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Spring 2020

Approval

Name:	**Siyuan Chen**
Degree:	**Master of Science (Statistics)**
Title:	**Approximate Marginal Likelihoods for Shrinkage Parameter Estimation in Penalized Logistic Regression Analysis of Case-Control Data**
Examining Committee:	**Chair:** Jiguo Cao Associate Professor
	Brad McNeney Senior Supervisor Professor
	Liangliang Wang Supervisor Assistant Professor
	Jinko Graham Internal Examiner Professor
Date Defended:	**April 17, 2020**

Abstract

Inference of associations between disease status and rare exposures is complicated by the finite-sample bias of the maximum likelihood estimator for logistic regression. Penalised likelihood methods are useful for reducing such bias. In this project, we studied penalisation by a family of log-F priors indexed by a shrinkage parameter m. We propose a method for estimating m based on an approximate marginal likelihood obtained by Laplace approximation. Derivatives of the approximate marginal likelihood for m are challenging to compute, and so we explore several derivative-free optimization approaches to obtaining the maximum marginal likelihood estimate. We conduct a simulation study to evaluate the performance of our method under a variety of data-generating scenarios, and applied the method to real data from a genetic association study of Alzheimer's disease.

Keywords: case-control study; penalised likelihood; log-F priors; Laplace Approximation; Alzheimer's Disease

Dedication

Hörst du: ich rede zu dir, wenn schwül sie das Sterben vermehren. Schweigsam entwerf ich mir Tod, leise begegn ich den Speeren.
Ich steh. Ich bekenne. Ich ruf.

Ein Krieger, Paul Celan

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor Dr.Brad McNeney for the continuous support of my study and research throughout the master program, and for his patience, encouragement and guidance that help me get to know and be enchanted with statistical genetics. I would also greatly appreciate the effort and time he has put into this project. Thank you for always being so supportive, kind-hearted and insightful.

I would also like to express my thanks to my examining committees, Dr. Jiguo Cao, Dr. Jinko Graham and Dr. Liangliang Wang for taking their time to participate in my defense. Many thanks go to all the staff and faculty in the Department of Statistics and Actuarial Science for their wonderful lectures, talks and generous help. Besides, I am grateful to have the opportunity to work with Dr.Celia Greenwood from Lady Davis Institute and have such an unforgettable summer.

Furthermore, I would like to thank all my lovely friends and fellow graduate students in SFU, and talented people in Greenwood's Lab, for the time we spent together, the laugh we had and the help I received. Special thanks go to Lulu Guo, for always being a considerate friend; also Dongmeng Liu, for offering generous help when I just arrived Canada.

Finally yet importantly, I would appreciate the love, support and company from Alexander Vasilenko, Natalia Vasilenko and Isabell (Kitty). I would also express my appreciation to my parents and grandparents for their love and care throughout my life. Thank you all for bringing me endless delights, and courage for pursuing my goals.

Table of Contents

Approval ii

Abstract iii

Dedication iv

Acknowledgements v

Table of Contents vi

List of Tables viii

List of Figures ix

1 Introduction 1

2 Methodology 3

2.1 Marginal likelihood for Shrinkage Parameter m . . . 3

2.2 Laplace Approximation . . . 5

2.3 Derivative-free Optimization Strategies . . . 8

2.3.1 Nelder-Mead . . . 8

2.3.2 Genetic Algorithm . . . 9

2.3.3 Particle Swarm Optimization . . . 10

2.4 Summary of Maximum Marginal Likelihood Estimator of m . . . 11

3 Simulation Study 12

3.1 Simulation Design . . . 12

3.2 Study 1: Optimization Methods Comparison . . . 13

3.3 Study 2: Dimension of Genetic Variants . . . 13

4 Real Data Modelling 16

4.1 Data Description . . . 16

4.2 Estimating Shrinkage Parameter m . . . 16

5 Discussion **18**

Bibliography **20**

Appendix A Implementation of Laplace Approximation **22**

Appendix B LDheatmaps for Genes in Real Data Analysis **24**

Appendix C Code **27**

List of Tables

Table 3.1 Study 1, simulation setting 13
Table 3.2 Mean and variance of estimated m 14
Table 3.3 Study 2, Frequency of acceptable estimation, mean and variance of estimation of m 14

Table 4.1 Genes included from ADNI-1 study 17
Table 4.2 Estimated m for each gene 17

List of Figures

Figure 2.1 The shape of log-F distribution 5

Figure 2.2 The original posterior density for β and corresponding unnormalized Normal density 7

Figure 2.3 $(\hat{L}_{MC} - \hat{L}_{LA})/\hat{L}_{MC}$, dashed lines indicating ∓ 0.3 8

Figure 2.4 $log(\hat{L}_{MC})$ and $log(\hat{L}_{LA})$ for $m = 0.5, 1, ..., 10$ 9

Figure 3.1 Study 1, estimation of m for settings of true $m = 2, 4, 8$ separately . 15

Figure B.1 LDheatmaps for genes included in real data analysis section, using R^2 measure of LD(part 1) 25

Figure B.2 LDheatmaps for genes included in real data analysis section, using R^2 measure of LD(part 2) 26

Chapter 1

Introduction

The case-control design is common for genetic epidemiology studies of the relationship between disease status and genetic variants of interest. Case-control studies are retrospective in the sense that subjects are sampled based on disease status, with covariates such as genetic variants measured retrospectively. Thus, disease status, Y, is fixed and covariate information, X, is random. However, inference of disease-risk parameters is by logistic regression. A prospective model which assumes disease status, Y, is random and covariate information X is fixed. The maximum likelihood estimator of the regression parameters is thus obtained under prospective sampling with fixed covariates and random binary response variables.

Regardless of the sampling design, the logistic regression estimator is known to be biased away from zero [5, 6], and the bias is particularly acute when the covariate data are sparse (e.g., many zeroes) or the sample size is small. In the case of extremely sparse or small samples the likelihood may be monotone, so that the maximum likelihood estimator does not exist.

Firth [5] proposed a bias-reduced modification of logistic regression, obtained by maximizing the logistic regression likelihood penalized by the Jeffreys prior. For regression parameters $\boldsymbol{\beta}$, the Jeffreys prior is proportional to $\sqrt{|I(\boldsymbol{\beta})|}$, where $I(\boldsymbol{\beta})$ is the Fisher information matrix. Thus, Firth's estimator maximizes the penalized log-likelihood

$$l^*(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) + \frac{1}{2} log|I(\beta)| \tag{1.1}$$

where $l(\boldsymbol{\beta})$ is the logistic regression log-likelihood. Firth showed that the maximizer of $l^*(\boldsymbol{\beta})$ always exists and has reduced the first-order bias compared to the maximum likelihood estimator. Though there is no formal justification for using Firth logistic regression on case-control data, it has been shown to perform well in simulation studies [6].

Inspired by Firth, Zhang [28] developed a bias-reduced estimator for case-control data. Zhang's estimator is the maximizer of a penalized *profile* likelihood, where the profile likelihood is obtained by maximizing over the infinite-dimensional parameter semiparametric case-control likelihood, and the penalty is like that in the equation (1.1), but with an estimate, $\hat{I}(\theta)$, of the Fisher information in the profile likelihood. Simulation studies suggest that the Zhang's method and Firth logistic regression have similar statistical properties when applied to case-control data [6]. The similarity of Firth and Zhang logistic regression suggests that we can apply other penalized logistic regression methods to case-contol data.

Alternatives to Firth logistic regression were considered by Greenland and Mansournia.[7]. They recommend penalization by log-F prior distributions over other possible priors such as normal, t- and Cauchy distributions. The family of log-F priors is indexed by a shrinkage parameter m. Larger values of m induce greater shrinkage. Graham et al.[6] found that the log-F priors performed well in limited simulations of case-control data, but did not propose a method for choosing m.

Greenland and Mansournia suggested an empirical Bayes approach to estimating m. The empirical Bayes estimator is the maximizer of the marginal likelihood for m obtained by integrating $\boldsymbol{\beta}$ out of the joint distribution of the data and $\boldsymbol{\beta}$. Yu [27] followed this suggestion and applied the EM algorithm to maximize the marginal likelihood, treating the regression coefficients $\boldsymbol{\beta}$ as missing data. However, the integral required for the E-step is challenging, and Yu implemented a Monte Carlo EM algorithm. The Monte Carlo EM algorithm was found to be computationally slow. In this project we investigate a computationally faster alternative, based on a Laplace approximation to the marginal likelihood.

The project is organized as follows. In Chapter 2 we formulate the marginal likelihood for m. Next we introduce the Laplace approximation of the marginal likelihood function and optimization methods for maximizing this approximate likelihood to obtain empirical Bayes estimates of m. In Chapter 3 we provide simulation studies to evaluate our method. Chapter 4 gives an application to real data from a genetic association study of Alzheimer's disease. Concluding remarks are made in Chapter 5.

Chapter 2

Methodology

2.1 Marginal likelihood for Shrinkage Parameter m

Consider a case-control study of association between a genetic variant and a disease status. Under case-control sampling design, cases and controls are selected from the population, and their covariates are recorded. Denote disease status on individual i as as $Y_i \in \{0, 1\}$, with 1 indicating case and 0 indicating control. Denote the k_{th} genetic variant covariate as X_i^k,$i = 1, ..., n, k = 1, ..., K$. For each genetic variant, assume a logistic regression model with log-OR parameter β_k and an intercept term α_k:

$$\log\left[\frac{P(Y_i = 1|X_i^k)}{P(Y_i = 0|X_i^k)}\right] = \alpha_k + \beta_k X_i^k,$$

or, equivalently,

$$P(Y_i = 1|X_i^k) = \frac{\exp(\alpha_k + \beta_k X_i^k)}{1 + \exp(\alpha_k + \beta_k X_i^k)}, i = 1, ..., n; k = 1, ..., K \tag{2.1}$$

For future reference we note that the intercept terms, α_k, are different for each covariate. Qin and Zhang [17] derived the following two-sample semi-parametric model for the covariate data:

$$\begin{aligned} P(X_i^k|Y_i = 0) &= g(X_i^k) \\ P(X_i^k|Y_i = 1) &= c(\beta_k, g)\exp(X_i^k\beta_k)g(X_i^k), \end{aligned} \tag{2.2}$$

where $g()$ is the distribution of X^k in controls and $c(\boldsymbol{\beta}, g)$ is a normalization constant. The likelihood is then

$$L(\beta, g) = \prod_{Y_i=0} g(X_i^k) \prod_{Y_i=1} c(\beta_k, g)\exp(X_i^k\beta_k)g(X_i^k). \tag{2.3}$$

Finding the MLE of β_k is complicated by the infinite-dimensional nuisance parameter g. Qin and Zhang [17] proved that a profile likelihood function obtained by profiling g out has the same form as the logistic regression (2.1), with a different intercept term α_k^*.

$$L(\alpha_k^*, \beta_k) = \prod_{i=1}^{n} \frac{exp(Y_i(\alpha_k^* + X_i^k \beta_k))}{1 + exp(\alpha_k^* + X_i^k \beta_k)} \tag{2.4}$$

The intercept α_k^* can be shown to depend on the logistic regression intercept, α, the ratio of cases to controls in the study, and the disease prevalence [20]. The dependence on α_k means that the intercepts are different for each covariate.

We penalize this profile likelihood with the log-$F(m, m)$ prior distribution [11]

$$p(\beta_k|m) = \frac{1}{Beta(m/2, m/2)} \frac{exp(-\frac{m}{2}\beta_k)}{(1 + exp(-\beta_k))^m}, k = 1, ..., K \tag{2.5}$$

The distribution has a symmetrical bell shape with mean zero and a variance that decreases as the parameter m increases. When $m > 10$, the distribution is nearly Normal [7].

Now assume K independent covariates. We combine the data across all k to form the joint penalized likelihood

$$L(\alpha^*, \boldsymbol{\beta}, m) = \prod_{i=1}^{K} L(\alpha_k^*, \beta_k) p(\beta_k|m) \tag{2.6}$$

where $\alpha^* = (\alpha_1^*, ..., \alpha_K^*)$. A marginal likelihood for the hyper-parameter m can be estimated by integrating $\boldsymbol{\beta}$ out of this joint likelihood:

$$L(\alpha^*, m) = \int \prod_{i=1}^{K} L(\alpha_k^*, \beta_k) p(\beta_k|m) d\boldsymbol{\beta} = \prod_{k=1}^{K} \int L(\alpha_k^*, \beta_k) p(\beta_k|m) d\beta_k \tag{2.7}$$

The vector of intercepts α^* is considered to be a nuisance parameter in this marginal likelihood. We note that the dimension of this intercept increases with the number of covariates K.

We make two additional remarks. First, although $L(\alpha_k^*, \beta_k)$ is a penalized profile likelihood, rather than a standard likelihood function, it is of the same form as a penalized prospective likelihood of logistic models. Prentice and Pyke showed that the MLE of β_k can be obtained by maximizing $L(\alpha_k^*, \beta_k)$ [16]. A similar approach in the context of partly linear regression models was taken by Shen [22]. Second, we note that m controls the variance of the log-F distribution, with larger m leading to larger penalties on β_k values that are far away from 0. In what follows we label the prior distributions as "flat", "medium" or "pointed" for m $m \leq 3$, $3 \leq m \leq 7$, and $m > 7$ respectively. In the Chapter 3 simulations we choose one m from each category.

Note that evaluating the integrals in $L(\alpha^*, m)$ analytically is challenging. A straight-forward way to solve this problem is to apply numerical approximate integration methods such as Gaussian approximation, Monte Carlo integration and quadrature methods. We considered Laplace Approximation in particular, because it is widely used for approximating marginal likelihoods and its simplicity in computation and minimal computation time are advantages over quadrature and MC integration, respectively.

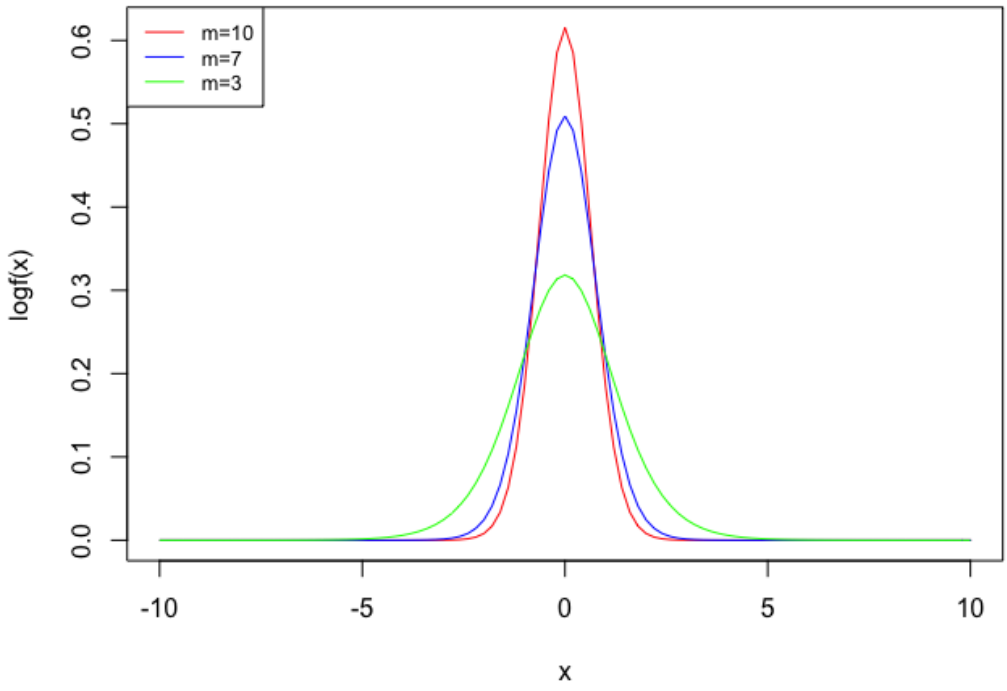


Figure 2.1: The shape of log-F distribution

2.2 Laplace Approximation

Our goal is to approximate the integrals in the marginal likelihood of equation (2.7). Each integral can be viewed as the marginal distribution of the data in a Bayesian problem with likelihood $L(\alpha_k^*, \beta_k)$ and prior $p(\beta_k|m)$. We first discuss theoretical results from the Bayesian inference literature that justify Laplace approximations of marginal distributions when the sample size is large. We then present an empirical investigation of the utility of Laplace approximation in the kinds of small sample problems that we are interested in.

Laplace's method approximates an integral by approximating the integrand with an easy-to-integrate function. In particular, the integrand is approximated by an unnormalized

Guassian density function whose mean coincides with the mode of the integrand. Suppose our integrand is an un-normalized posterior density $P(\theta)$ with $l(\theta) = \log P(\theta)$. If $l(\theta)$ is a smooth function with continuous second derivative and a sharp peak about its maximum at $\hat{\theta}$, we can approximate it with a 2-term Taylor expansion:

$$\begin{aligned} l(\theta) &\approx l(\hat{\theta}) + (\hat{\theta} - \theta) l^{'}(\hat{\theta}) + \frac{1}{2}(\hat{\theta} - \theta)^2 l^{''}(\hat{\theta}) \\ &= l(\hat{\theta}) + 0 + \frac{1}{2}(\hat{\theta} - \theta)^2 l^{''}(\hat{\theta}) \\ &= const - \frac{1}{2}(\hat{\theta} - \theta)^2 (-l^{''}(\hat{\theta})), \end{aligned} \tag{2.8}$$

where $l^{'}$ and $l^{''}$ denote the first and second derivatives of l, respectively. Up to a constant, the approximation matches the log-*pdf* of a Gaussian distribution with mean $\hat{\theta}$ and variance equal to the inverse of $-l^{''}(\hat{\theta})$. Thus we can approximate $P(\theta)$ by an unnormalized Normal distribution $P(\hat{\theta}) exp[-\frac{c_P(\theta-\hat{\theta})^2}{2}]$, where $c_P = -\frac{d^2 log(P(\theta))}{d\theta^2}|_{\theta=\hat{\theta}}$. It follows that the desired normalization constant $\int P(\theta) d\theta$ can be approximated by [14]

$$\begin{aligned} \int P(\hat{\theta}) exp[-\frac{c_P(\theta - \hat{\theta})^2}{2}] d\theta &= P(\hat{\theta}) \int exp[-\frac{c_P(\theta - \hat{\theta})^2}{2}] d\theta \\ &= P(\hat{\theta}) \sqrt{\frac{2\pi}{c_P}} \end{aligned} \tag{2.9}$$

An expression for $P(\hat{\theta})\sqrt{\frac{2\pi}{c_P}}$ in our problem is given in Appendix A.

The quality of the Laplace approximation to the posterior density function has been discussed theoretically for parametric models. For example, Tierney and Kadane [24] proved that for a smooth prior distribution and fixed value of the data the Laplace approximation to the marginal posterior density has an error that is $O(n^{-1})$ in a multi-parameter setting. Another motivation for the approximation is a result called the Bernstein-von Mises Theorem (see the version below taken from [25]) that tells us that posterior distributions tend toward Gaussian distributions as the sample size increases. From this it follows that the unnormalized posterior distribution $P(\theta)$ tends towards an unnormalized Gaussian.

Theorem (Bernstein-von Mises Theorem) Consider $X_1, X_2, ..., X_n$ are IIDs from the *pdf* $f(x_i|\theta), \theta \in \Theta$. If $log f(x_i|\theta)$ is twice continuously differentiable w.r.t θ for each x_i, and the sample size n is large enough, for any positive, bounded and twice differentiable over Θ prior density $\xi(\theta)$, we have:

$$sup_z |P(\theta \leq z | X = x) - \Phi(\sqrt{c_P}(z - \hat{\theta}))| \approx 0 \tag{2.10}$$

The Bernstein-von Mises Theorem proves that when the sample size is large the posterior density will be close to a normal density, which assures the quality of the Laplace

approximation. To investigate the quality of Laplace Approximation in our context we performed limited simulations to assess whether the shape of the unnormalized posterior $L(\alpha_k^*, \beta_k)p(\beta_k|m)$ is close to a normal density function and to judge the quality of Laplace Approximation to the marginal likelihood. The simulations were conducted as described in Chapter 3, with the exception that here we used a small sample of 10 cases and 40 controls. The results are as follows.

For a single covariate simulated under $m = 4$ a plot of $L(\alpha_1^*, \beta_1)p(\beta_1|m)$ for $m = 4$ and $\alpha^* = -3$ is shown in Figure 2.2, with the approximating unnormalized Gaussian distribution superposed. We can see that the posterior is unimodal (see Appendix A for a proof of unimodality) with a heavier tail than the approximating Gaussian. Overall the approximation looks reasonable for this simulated covariate.

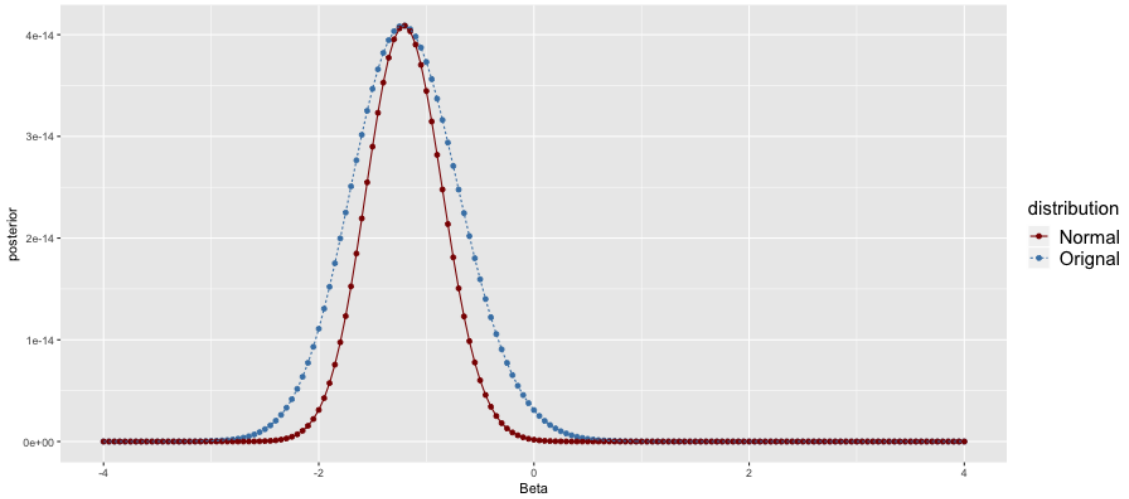


Figure 2.2: The original posterior density for β and corresponding unnormalized Normal density

Next we investigate the quality of Laplace Approximation to the marginal likelihood. For each dataset, the marginal likelihood $L(\alpha, m) = \int L(X|\beta)p(\beta|m)d\beta$ can be regarded as $\boldsymbol{E}_p(L(X)|\Theta)$, in which Θ indicates the parameter space of β. Such an expectation can be estimated by Monte Carlo by sampling β's from the prior distribution and calculating the mean of the likelihood values from each β. The precision of such an estimate depends on the Monte Carlo sample size. In the following results we used a Monte Carlo sample size of 1 million.

We compare the approximated marginal likelihood at $m = 4$ and $\alpha^* = -3$ from Laplace Approximation, $\hat{L}_{LA}$, to the estimation from Monte Carlo, $\hat{L}_{MC}$, for each of 100 simulated single-covariate datasets simulated under $m = 4$, and calculate the relative difference $(\hat{L}_{MC} - \hat{L}_{LA})/\hat{L}_{MC}$. In our study, the absolute relative difference is less than 0.3 in around 75% of the datasets (see Figure 2.3).

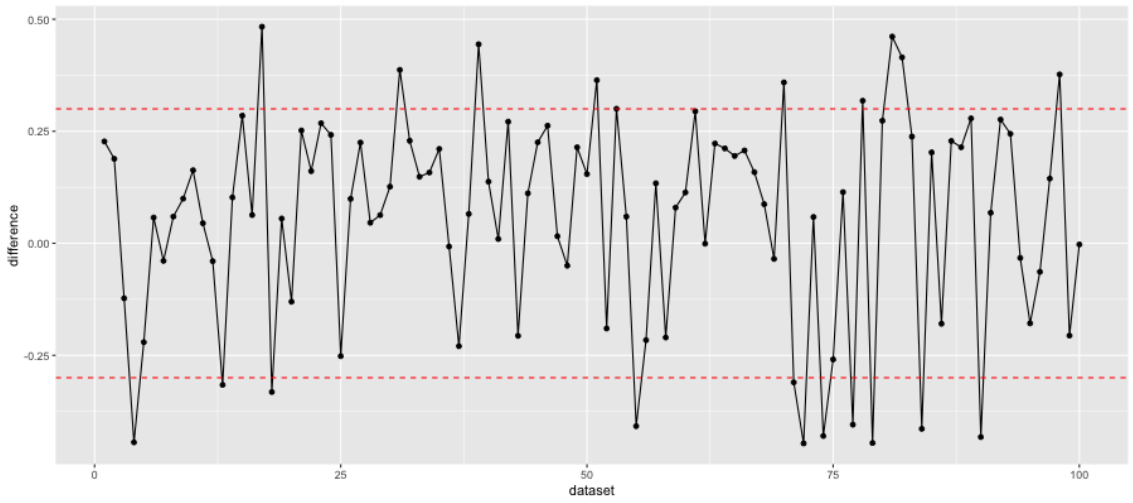


Figure 2.3: $(\hat{L}_{MC} - \hat{L}_{LA})/\hat{L}_{MC}$, dashed lines indicating ∓ 0.3

Finally, for a single dataset we compared the MC and LA estimates of the marginal likelihood of $\alpha^* = -3$ and a grid of $m = (0.5, 1, ..., 10)$. We plot the natural log of LA and MC estimates versus m(see figure 2.4). We see that the argmax of the LA-approximated marginal likelihood is smaller than the *argmax* of the MC-approximated marginal likelihood. Whether such underestimation is typical and leads to biased estimation of m is an area for future work.

2.3 Derivative-free Optimization Strategies

We maximize the approximate marginal likelihood, denoted $\widetilde{L}(\alpha^*, m)$ to estimate (α^*, m). Calculation of derivatives of $\widetilde{L}(\alpha^*, m)$ is challenging and so we opted for derivative-free optimization methods. We consider the Nelder-Mead algorithm, a genetic algorithm, and the particle swarm optimization method. The genetic algorithm and particle swarm are examples of the larger class of evolutionary algorithms. We discuss each method briefly in the following subsections. Our simulations (Chapter 3) suggested that the genetic algorithm and the particle swarm optimization method perform better in general. Throughout we let $f(x)$ denote the objective function to be maximized over x in some subset Ω of $\mathbb{R}^p$.

2.3.1 Nelder-Mead

The Nelder-Mead extended simplex method is most easily described for the case $p = 2$. Starting from an initial triangle over Ω and with f evaluated at each of the vertices, we pivot the triangle by reflecting the lowest-value vertex across the edge opposite, in a way that maintains the area of the triangle. We continue this pivoting process, tending towards higher values of the objective function, until the objective function value is nearly the same at all vertices of the triangle. Refinements of the algorithm allow for the triangle to be

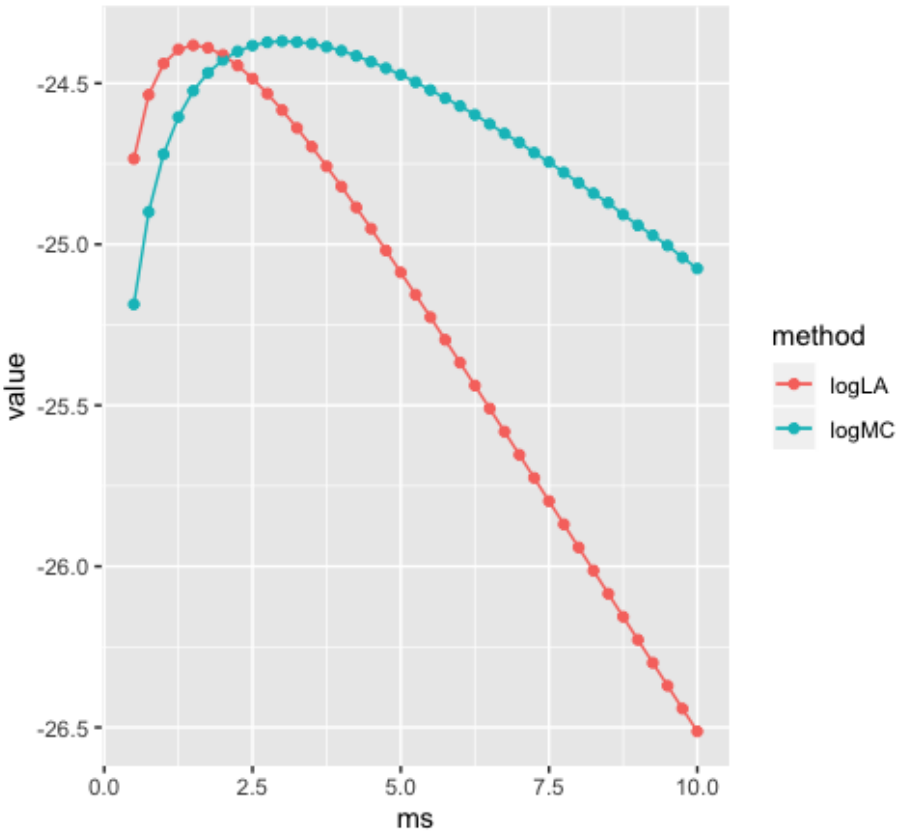


Figure 2.4: $log(\hat{L}_{MC})$ and $log(\hat{L}_{LA})$ for $m = 0.5, 1, ..., 10$

expanded or contracted at different iterations [13], to change the speed at which we move through Ω. For p-dimensional Ω we replace triangles with simplexes. Though simple, the Nelder-Mead method is not guaranteed to converge, and there are multiple examples of its failure, even in two dimensions [26, 15]. The Nelder-Mead algorithm is implemented in the R function *optim()* included in the base-R *stats* package [18].

2.3.2 Genetic Algorithm

Genetic algorithms are stochastic search algorithms that equate values of the objective function to genetic fitness and mimic the process of natural selection of organisms to maximize this fitness measure. Starting from an initial "generation" of candidate solution vectors $x_1, \ldots, x_N \in \Omega$, the x_i's are allowed to "recombine" with others or "mutate" to produce the next generation, with higher-fitness x_i's being more likely to reproduce. In our context, recombination is the exchange of coordinates between two vectors and mutation is the change of a specific coordinate value to a new value. The solution is the highest value of the objective function found before the algorithm terminates. Termination can be because there is no improvement in the fitness value for specific number of iterations, or the algorithm

reaches a specified maximum generations. The approach was pioneered by Holland [10] and later generalized; see Corez [4] for a review. A pseudo-code implementation of a genetic algorithm is shown in Algorithm 1 [19] below. In our study we use the *ga()* function from the R package *GA* [21].

Algorithm 1: Genetic Algorithm

Result: final population P
Input: evaluation function f, control parameters set C;
Initialization: random initial population P;
while *termination condition not satisfied* **do**
  evaluate current population and get the best individuals E from P;
  select parents set $Parents$ from P;
  if *crossover condition satisfied* **then**
    $Children \leftarrow$ crossover($Parents$);
  end
  if *mutation condition satisfied* **then**
    $Children \leftarrow$ mutation($Parents$);
  end
  $P \leftarrow E \bigcup Children$
end

2.3.3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a stochastic optimization technique proposed by Eberhart and Kennedy [12]. An initial set of vectors $x_1, \ldots, x_N$ is viewed as "particles" that can move about Ω. The velocity (direction and speed) of the movements are partly random and partly influenced by values of the objective function seen previously by the particle itself and others in its neighbourhood. Various modifications of PSO, and hybrids of PSO and other modern evolutionary optimization algorithms are reviewed by Cortez [4] . Pseudo-code for the SPSO 2007 algorithm of Clerc [3] is given in Algorithm 2 [4]. In our study we use the

psoptim() function from the R package *pso* [2], which implements both the SPSO 2007 and SPSO 2011 algorithms of [3].

Algorithm 2: Particle Swarm Optimization

Result: Best solution B

Input: evaluation function f, control parameters set C;

Initialization: initial swarm P, including random position and velocity for each particle;

get the best particle B;

while *termination condition not satisfied* **do**

  for *each particle* $x = (s, v, p, l) \in P$ **do**

    update the velocity $v \leftarrow velocity(s, v, p, l)$;

    move the particle $s \leftarrow s + v$;

    check if x is on the boundary, if so, adjust;

    if s *is better fitted than* p **then**

      $p \leftarrow s$

    end

    if s *is better fitted than* B **then**

      $B \leftarrow s$

    end

  end

  update l for all particles following given topology in C;

end

2.4 Summary of Maximum Marginal Likelihood Estimator of m

The approximate marginal likelihood is $\widetilde{L}(\alpha^*, m) = \prod_{k=1}^{K} \widetilde{E}[L|\alpha^*, m]$, where $\tilde{E}[L|\alpha^*, m]$ is the Laplace approximation to $\int L(\alpha^*, \beta)p(\beta|m)d\beta$. Each approximate integral $\widetilde{E}[L|\alpha^*, m]$ can be evaluated as discussed in the Appendix. To avoid underflow we take logarithms and maximize $\widetilde{l}(\alpha^*, m) = \sum_{k=1}^{K} \log \widetilde{E}[L|\alpha^*, m]$. This objective function can be passed to any of the three derivative-free optimization methods discussed in Section 2.3 to obtain the estimates $(\hat{\alpha^*}, \hat{m})$.

Chapter 3

Simulation Study

Our simulation study addressed two questions:

1. Which of Nelder-Mead, the genetic algorithm (GA) or particle swarm optimization (PSO) is the best optimization method for our problem?
2. How does the number of genetic markers affect the bias and variance of our estimator of m?

In this chapter we describe the design of our study and the results.

3.1 Simulation Design

Under the covariate data model (2.2) and prior distribution (2.5) the parameters of the data-generating process are (i) the numbers of cases, n_1, and controls, n_0, (ii) the distribution of covariates in controls, g, and (iii) the precision parameter of the prior, m. In addition we can control the number of genetic markers K. In all simulations we set $n_1 = 200$, $n_0 = 800$ and g to be the standard normal distribution. We chose a relatively large sample size so that the quality of the Laplace approximations would not be an issue, and used a fairly standard 1:4 ratio of cases to controls. The choice of a standard normal g was for convenience. One can show that with this distribution of covariates in controls, the distribution of covariates in cases is also normal with variance 1, but with mean equal to the log-OR parameter β_k [27]. Thus, we can generate covariates directly from their distributions, rather than the alternative of generating a large cohort and sub-sampling cases and controls. For each simulation configuration we generated 20 data sets.

For the comparison of optimization methods (question 1) we set $K = 20$ and considered $m = 2$, 4 or 6. For the investigation of the bias and variance of our estimator as a function of K (question 2) we set $m = 4$ and considered $K = 10$, 20, 30, 40 or 70. In an empirical Bayes procedure, information about the hyperparameter of the prior accrues as the number

of samples from the prior increases, which in our study is as K increases. We therefore expect bias and variance of the estimator of m to decrease with K. For each simulation configuration we generated 20 data sets.

3.2 Study 1: Optimization Methods Comparison

The methods of Nelder-Mead, GA and the PSO algorithm SPSO 2011 were run with their default settings, and the same initial values of the parameters. An initial value of $m = 4$ was selected when the true m was 2 or 8, and an initial $m = 6$ was chosen when the true m was 4. The GA and PSO methods also allow the user to limit the range of m values to search; the search limits we chose are shown in Table 3.1.

True m	Initial m	Method	Setting
2	4	Nelder-Mead	NA
		GA	$m \in \{0, 10\}$
		PSO	$m \in \{0, 10\}$
4	6	Nelder-Mead	NA
		GA	$m \in \{0, 10\}$
		PSO	$m \in \{0, 10\}$
8	4	Nelder-Mead	NA
		GA	$m \in \{0, 15\}$
		PSO	$m \in \{0, 15\}$

Table 3.1: Study 1, simulation setting

The results are shown in Figure 3.1 and Table 3.2. As an indication of performance, the shaded area in each panel is the range $(0.5m, 1.5m)$. For $m = 2$, there is no obvious difference in performance between Nelder-Mead and PSO, while GA tends to overestimate. For $m = 4$, all three methods provide reasonable estimates of m, though Nelder-Mead always underestimates the true value. Under the largest value $m = 8$ the estimates from Nelder-Mead appear to be substantially downwardly biased, while the estimates from PSO are highly variable. Overall, GA outperforms the other two methods in terms of accuracy.

Note that there are datasets for which all three methods give similar estimates that are far below the true value. We speculate that for these datasets the simulated β_k values are highly dispersed and are more compatible with a small value of m.

3.3 Study 2: Dimension of Genetic Variants

Though GA was the best method in Study 1, we used PSO in this study because of its reduced computation time. Following Clerc [3] we chose the number of particles in the swarm to be 25, 35, 45 and 45 for $K = 10, 20, 30$ and 50, respectively. For $K = 70$ there is no recommendation and we chose a swarm of 80 particles. The initial value of m and

True m	Method	Mean of $\hat{m}$	Variance of $\hat{m}$
2	Nelder-Mead	1.634081	0.033003
	PSO	1.594066	0.07042947
	GA	3.105852	1.494641
4	Nelder-Mead	2.298701	0.2384101
	PSO	2.607855	0.8402178
	GA	3.729839	1.277228
8	Nelder-Mead	3.258928	0.5827398
	PSO	5.087871	4.29019
	GA	5.273265	2.650115

Table 3.2: Mean and variance of estimated m

ranges of m values to search were 6 and (0,10), as in Study 1 (see Table 3.1). The frequency of estimates being between 2 and 6, which is the range of $(0.5m, 1.5m)$, along with the mean and variance of estimates of m are shown in Table 3.3. Though there is a slight trend toward lower bias and variance as K increases, these trends are weaker than expected. Larger numbers of simulation replicates may be needed to see the expected trends.

p	Frequency	Mean	Variance
10	10	6.007	1.267608
20	18	3.576	0.6640005
30	18	4.794	0.581193
50	20	4.707	0.4509085
70	20	4.328	0.3870284

Table 3.3: Study 2, Frequency of acceptable estimation, mean and variance of estimation of m

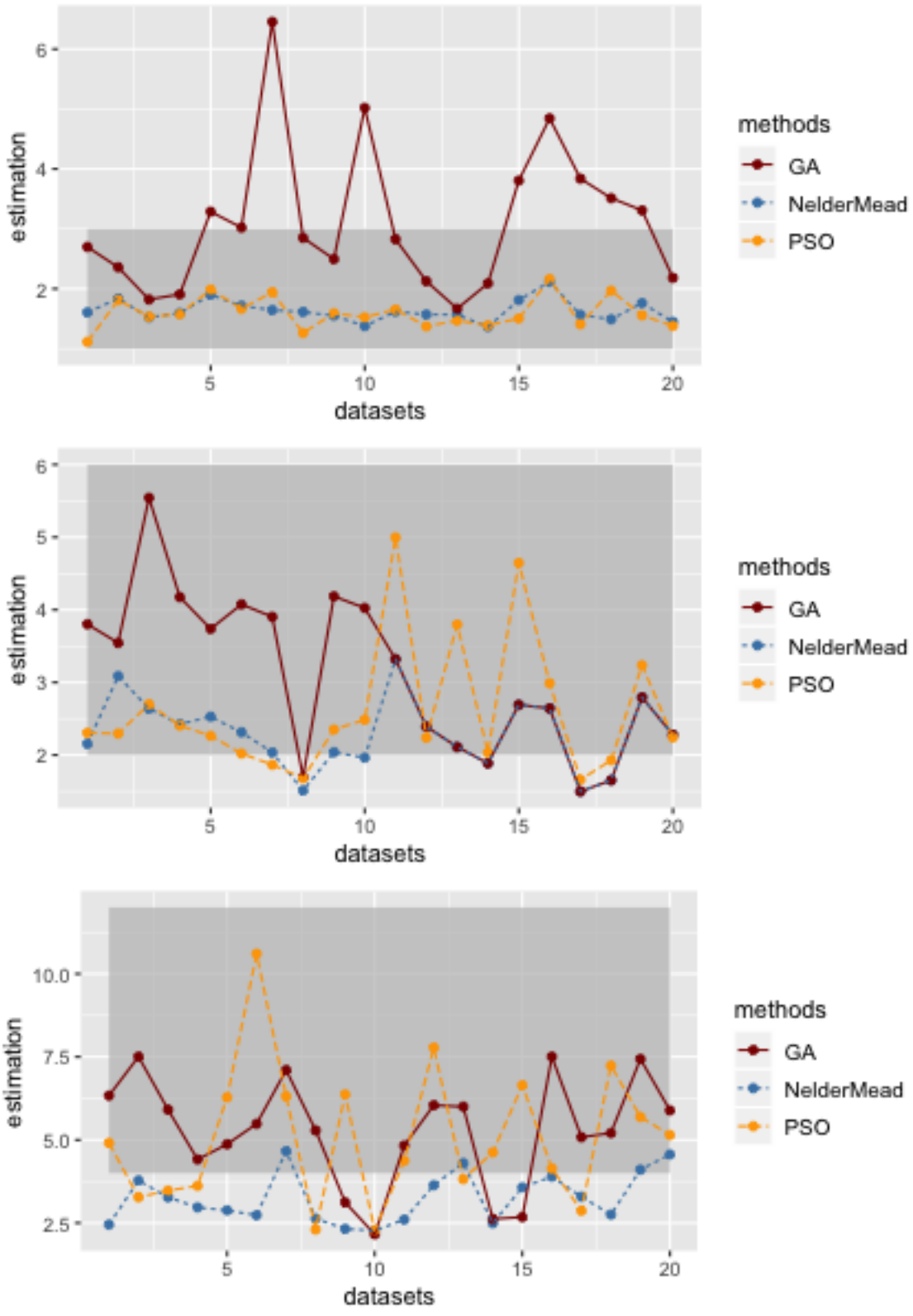


Figure 3.1: Study 1, estimation of m for settings of true $m = 2, 4, 8$ separately

Chapter 4

Real Data Modelling

4.1 Data Description

The Alzheimer's Disease Neuroimaging Initiative (ADNI) is a longitudinal multicenter study, aiming to identify significant genetic variants for the early detection and tracking of AD [1]. We apply our methodology to a dataset obtained from the first phase of the ADNI study (abbreviated ADNI-1). ADNI-1 is a five-year study with an initial participant pool of 800 subjects, among which 200 are cognitively normal individuals (CN), 200 are diagnosed to have Alzheimer's Disease (AD) and 400 had mild cognitive impairment (MCI). More information about the ADNI-1 study design are available on the ADNI website.

Our analysis is of the AD (case) and CN (control) subjects from ADNI-1. In particular, we use a subset of 179 CN and 144 AD subjects from a dataset prepared for Greenlaw *et al.* [8]. The SNPs for this study are from the top 40 candidate genes for AD in the AlzGene database on June 10, 2010 [8]. After imputation and quality control there were 490 SNPs in 33 genes. Of these we focus on the 426 SNPs in the 12 genes with 10 or more SNPs (Table 4.1). Covariates for each SNP genotype are coded as 0, 1 or 2 copies of the minor allele.

4.2 Estimating Shrinkage Parameter m

An assumption of our method is that covariates are independent. To assess the plausibility of this assumption we generated LDheatmaps [23] for the genes in Table 4.1 (see Appendix B). LDheatmaps display measures of linkage disequilibrium (LD) between pairs of SNPs. These graphs suggest relatively low correlation between SNPs. Under the assumption of independent covariates, and using PSO as the optimization method we estimated m for each gene. To mitigate the possibility of the optimizer finding local maxima we re-ran the optimization procedure 10 times with different initial values, and took the estimate $\hat{m}$ to be the argument of the maximum. The results are shown in Table 4.2. We included some notes

Gene	Chromesome	Basepair Start	Basepair End	Number of SNPs
CR1	1	205737551	205881074	15
ECE1	1	21417188	21546592	37
MTHFR	1	11768839	11789631	10
BIN1	2	127534016	127575911	12
NEDD9	6	11300561	11491962	69
DAPK1	9	89300645	89511843	82
IL33	9	6203387	6241507	14
SORCS1	10	108334549	1089911743	94
GAB2	11	77608440	77801479	18
PICALM	11	85349350	85458970	23
SORL1	11	120826964	121006965	33
ADAM10	15	56690061	56809304	19

Table 4.1: Genes included from ADNI-1 study

indicating that results reached the upper boundary and are regarded as unstable so that we cannot make appropriate inference about m, or that the estimated m is larger than 10, which makes the log-F prior practically Normal [7]. The results suggest m-values of about 4 for ECE1, 5 for NEDD9, 8.5 for SORCS1 and values of 10 or more for the other genes.

Gene	$\hat{m}$
CR1	>10
ECE1	4.17
MTHFR	Upper bound reached
BIN1	7.76
NEDD9	4.80
DAPK1	>10
IL33	Upper bound reached
SORCS1	8.54
GAB2	Upper bound reached
PICALM	>10
SORL1	>10
ADAM10	>10

Table 4.2: Estimated m for each gene

Chapter 5

Discussion

Our method of estimating the shrinkage parameter m is based on an approximated marginal likelihood. We started from the penalised profile likelihood of the log-OR parameters β with a log-F prior, which can be regarded as a modification of Zhang's approach. By approximating the marginal likelihood obtained by integrating out the β's, we can estimate the shrinkage parameter. We argued that the Laplace approximation used to approximate the marginal likelihood is reasonably good for our unimodal penalized likelihood. The advantage of our approach over the Monte Carlo EM approach of Yu [27] is the reduced computation time. We explored some of the properties of the estimator by simulation and applied it to data from a genetic association study of Alzheimer's disease.

The possible methods that could be used to maximize the approximated marginal likelihood are not limited to the three methods mentioned in this project. The factors that make optimization problems more difficult are the dimension of the parameter space and flatness or irregularity of the criterion function. In our problem the dimension is controlled by the number of covariates K. We expect that the search for m will actually become easier as K increases and we gain more replicates from the prior. We therefore focused our investigation of the optimization methods on the impact of the true m on the criterion function. Our tentative conclusions from our simulation study is that the optimization problem becomes more challenging as m increases. For small m the Nelder-Mead and PSO approaches worked well, but for large m we benefit from the stability of the GA approach. The drawback of GA is its increased computation time. One suggestion is to fit regressions under a trial value of m to get a sense of the variability in the β coefficients. If these appear concentrated about 0, suggesting a large m, then one might adopt GA to estimate m.

We applied our method to data from the ADNI-1 study. Assuming independence of SNPs within a gene we obtained estimates of m in the ECE1, BIN1, NEDD9 and SORCS1 genes. These values could be used in subsequent SNP-specific logistic regression modeling. For the

other genes we studied we were not able to obtain trustworthy estimates of m and would recommend a large value of m=10, or a Gaussian prior. The genetic covariates in the ADNI-1 study were 0, 1, 2 counts, which differed from the data generation in our simulations. An area of future work is to conduct simulations with sparse count covariates.

A shortcoming of this project is that our method is not appropriate for low-dimensional datasets. Information about the shrinkage parameter m comes from multiple realizations from the prior distribution and we therefore need multiple covariates. By contrast, very high-dimensional datasets pose computation problems and may lead to poor performance of the optimization methods. For example, Helwig and Wanka [9] showed that the initialization and bound handling mechanism of particle swarm optimization can cause particles to become trapped at local maxima in high-dimensional search spaces.

The major limitation of this work is that it does not provide confidence intervals for our estimates of the shrinkage parameter m. One possible approach is to obtain confidence intervals by inverting a profile likelihood ratio test. The profile likelihood is obtained as follows. For fixed m we consider the marginal log-likelihood to be a function of the α^*'s. We can use a derivative-free optimization method to maximize this function over the α^*'s to obtain an approximate profile likelihood value at m. Repeating this procedure for a grid of m values gives an approximate profile log-likelihood for m. A profile likelihood ratio test of a specific value m_0 would retain the null hypothesis when -2 times the log-likelihood ratio of m_0 *versus* $\hat{m}$ is less than about 4. This reasoning leads to the so-called "drop-down-two" confidence interval comprised of all m_0 such that the estimated profile log-likelihood at m_0 is within about 2 of the estimated profile log-likelihood at $\hat{m}$. Investigation of the properties of such an approach can be included in the future work.

Ultimately, the purpose of estimating m is to use it as the smoothing parameter in single-SNP logistic regression analyses. It is therefore of interest to explore the statistical properties of the log-OR estimator from the two-step process of first estimating m and then estimating log-ORs under a log-F(m,m) penalty. In addition to considering the approximate maximum likelihood estimator $\hat{m}$, we might also use the m value at, say, the upper or lower limits of the confidence interval for m. These explorations are also future work.

Bibliography

[1] *ADNI General Procedures Manual*, 2006. https://adni.loni.usc.edu/wp-content/uploads/2010/09/ADNI_GeneralProceduresManual.pdf.

[2] Claus Bendtsen. *pso: Particle Swarm Optimization*, 2012. R package version 1.0.3.

[3] Maurice Clerc. Standard Particle Swarm Optimisation. https://hal.archives-ouvertes.fr/hal-00764996, September 2012.

[4] Paulo Cortez. *Modern Optimization with R*. Springer International Publishing, 2014.

[5] David Firth. Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1):27–38, 1993.

[6] Jinko Graham, Brad McNeney, and Robert W. Platt. Small sample methods. In Nilanjan Chatterjee Mitchell H. Gail Alastair Scott Norman Breslow, Oernulf Borgan and Christopher John Wild, editors, *Handbook of Statistical Methods for Case-Control Studies*, page 134–162. Chapman & Hall/CRC Handbooks of Modern Statistical Methods, 2018.

[7] Sander Greenland and Mohammad Ali Mansournia. Penalization, bias reduction, and default priors in logistic and related categorical and survival regressions. *Statistics in medicine*, 34(23):3133–3143, 2015.

[8] Keelin Greenlaw, Elena Szefer, Jinko Graham, Mary Lesperance, Farouk S Nathoo, and Alzheimer's Disease Neuroimaging Initiative. A Bayesian group sparse multi-task regression model for imaging genetics. *Bioinformatics*, 33(16):2513–2522, 2017.

[9] Sabine Helwig and Rolf Wanka. Particle Swarm Optimization in high-dimensional bounded search spaces. In *2007 IEEE Swarm Intelligence Symposium*, pages 198–205, 2007.

[10] John Henry Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.

[11] M. C. Jones. Families of distributions arising from distributions of order statistics. *Test*, 13(1):1–13, 2004.

[12] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, volume 4, pages 1942–1948 vol.4, 1995.

[13] Jeffrey Larson, Matt Menickelly, and Stefan M. Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019.

[14] David J.C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[15] K. I. M. McKinnon. Convergence of the Nelder-Mead simplex method to a nonstationary point. *SIAM Journal on Optimization,*, 9(1):148–158, 1998.

[16] Ross L Prentice and Ronald Pyke. Logistic disease incidence models and case-control studies. *Biometrika*, 66(3):403–411, 1979.

[17] Jing Qin and Biao Zhang. A goodness-of-fit test for logistic regression models based on case-control data. *Biometrika*, 84(3):609–618, 1997.

[18] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

[19] Colin Reeves and Jonathan Rowe. *Genetic Algorithms—Principles and Perspectives: A Guide to GA Theory*. Springer US, 2002.

[20] AJ Scott and CJ Wild. Maximum likelihood for generalised case-control studies. *Journal of Statistical Planning and Inference*, 96(1):3–27, 2001.

[21] Luca Scrucca. GA: A package for genetic algorithms in R. *Journal of Statistical Software*, 53, 2013.

[22] Xiaotong Shen. Asymptotic normality of semiparametric and nonparametric posterior distribution. *Journal of the American Statistical Association*, 97:222–235, 2002.

[23] Ji-Hyung Shin, Sigal Blay, Brad McNeney, and Jinko Graham. LDheatmap: An R function for graphical display of pairwise linkage disequilibria between single nucleotide polymorphisms. *Journal of Statistical Software, Code Snippets*, 16(3):1–9, 2006.

[24] Luke Tierney and Joseph B. Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81:82–86, 1986.

[25] A. W. van der Vaart. *Bayes Procedures*, page 138–152. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

[26] Daniel John Woods. *An interactive approach for solving multi-objective optimization problems*. PhD thesis, Rice University, 1985.

[27] Ying Yu. Shrinkage parameter estimation for penalized logistic regression analysis of case-control data. Master's thesis, Simon Fraser University, 2019.

[28] Biao Zhang. Bias-corrected maximum semiparametric likelihood estimation under logistic regression models based on case–control data. *Journal of statistical planning and inference*, 136(1):108–124, 2006.

Appendix A

Implementation of Laplace Approximation

Recall the marginal likelihood for (α^*, m) and denote the product $L(\alpha_k^*, \beta_k)p(\beta_k|m)$ by L_p, which can be regarded as an unnormalized posterior density. Note that L_p is differentiable, and denote the maxima of L_p with α_k^*, m given, as β_k^{max}. $\int L_p d\beta_k$ can be approximated with

$$L_p|_{\beta_k^{max}} \sqrt{\frac{2\pi}{c_P}}, c_P = -\frac{\partial^2}{\partial \beta_k} log(L_p)|_{\beta_k^{max}} \tag{A.1}$$

In Practice, the value of L_p can be too small to compute in R, instead we computed $log(L_p)$ to access the value of β_k^{max} by simply taking the derivatives. Plug-in (2.4) and (2.5) we have

$$\begin{aligned} log(L_p) = \sum_{i=1}^{n} (Y_i(\alpha_k^* + X_i^k \beta_k) - log(1 + exp(\alpha_k^* + X_i^k \beta_k))) \\ -log(Beta(m/2, m/2)) - \frac{m}{2}\beta_k - m * log(1 + exp(-\beta_k)) \end{aligned} \tag{A.2}$$

$$\frac{\partial log(L_p)}{\partial \beta_k} = \sum_{i=1}^{n} (Y_i X_i^k - \frac{exp(\alpha_k^* + X_i^k \beta_k) X_i^k}{1 + exp(\alpha_k^* + X_i^k \beta_k)}) - \frac{m}{2} + m \frac{exp(-\beta_k)}{1 + exp(-\beta_k)} \tag{A.3}$$

To show that L_p is well-peaked enough for Laplace approximation, we prove the following result to ensure its unimodality:

Result The root of $\partial log(L_p)/\partial \beta_k = 0$, denoted by β_k^{max}, is the global maxima of L_p.

Proof Rewrite (A.3) with notations of $e_\alpha = exp(\alpha_k^*), e_k = exp(\beta_k)$, we have

$$\sum_{i=1}^{n} (Y_i X_i^k - \frac{e_\alpha e_k^{X_i^k} X_i^k}{1 + e_\alpha e_k^{X_i^k}}) - \frac{m}{2} + m \frac{1/e_k}{1 + 1/e_k} = \sum_{i=1}^{n} (Y_i X_i^k - X_i^k + \frac{X_i^k}{1 + e_\alpha e_k^{X_i^k}}) - \frac{m}{2} + \frac{m}{1 + e_k} \tag{A.4}$$

Consider when $\beta_k \to -\infty, e_k \to 0$, $\frac{X_i^k}{1+e_\alpha e_k^{X_i^k}} \to X_i^k$ when $X_i^k > 0$; $\to 0$ when $X_i^k < 0$. Then

$$\begin{aligned}
&\sum_{i=1}^{n}(Y_i X_i^k - X_i^k + \frac{X_i^k}{1+e_\alpha e_k^{X_i^k}}) - \frac{m}{2} + \frac{m}{1+e_k} \\
&\to \sum_{i=1}^{n}(Y_i X_i^k - X_i^k) + \sum_{i=1}^{n} X_i^k I(X_i^k > 0) + \frac{m}{2} \\
&= \sum_{i=1}^{n}(Y_i X_i^k) - \sum_{i=1}^{n} X_i^k I(X_i^k < 0) + \frac{m}{2} \\
&= \sum_{i=1}^{n}(X_i^k I(Y_i = 1) I(X_i^k > 0) - X_i^k (Y_i = 0) I(X_i^k < 0)) + \frac{m}{2} > 0
\end{aligned} \tag{A.5}$$

Similarly, when $\beta_k \to \infty, e_k \to \infty$, $\frac{X_i^k}{1+e_\alpha e_k^{X_i^k}} \to 0$ when $X_i^k > 0$; $\to X_i^k$ when $X_i^k < 0$. We have

$$\begin{aligned}
&\sum_{i=1}^{n}(Y_i X_i^k - X_i^k + \frac{X_i^k}{1+e_\alpha e_k^{X_i^k}}) - \frac{m}{2} + \frac{m}{1+e_k} \\
&\to \sum_{i=1}^{n}(X_i^k I(Y_i = 1) I(X_i^k < 0) - X_i^k (Y_i = 0) I(X_i^k > 0)) - \frac{m}{2} < 0
\end{aligned} \tag{A.6}$$

Since (A.3) is continuous on $\mathbb{R}$, a root of $\partial log(L_p)/\partial \beta_k = 0$ must exist according to intermediate value theorem. Next we prove this root, denoted by β_k^{max} is the only root. The Hessian

$$\begin{aligned}
\frac{\partial^2}{\partial \beta_k^2} log(L_p) &= \sum_{i=1}^{n}(-\frac{exp(\alpha_k^* + X_i^k \beta_k)(X_i^k)^2}{1+exp(\alpha_k^* + X_i^k \beta_k)} + \frac{(exp(\alpha_k^* + X_i^k \beta_k))^2 (X_i^k)^2}{(1+exp(\alpha_k^* + X_i^k \beta_k))^2}) \\
&\quad + m(-\frac{exp(-\beta_k)}{1+exp(-\beta_k)} + \frac{(exp(-\beta_k))^2}{(1+exp(-\beta_k))^2}) \\
&= \sum_{i=1}^{n}((-1 + \frac{exp(\alpha_k^* + X_i^k \beta_k)}{1+exp(\alpha_k^* + X_i^k \beta_k)}) \frac{exp(\alpha_k^* + X_i^k \beta_k)(X_i^k)^2}{1+exp(\alpha_k^* + X_i^k \beta_k)}) \\
&\quad + m((-1 + \frac{exp(-\beta_k)}{1+exp(-\beta_k)}) \frac{exp(-\beta_k)}{1+exp(-\beta_k)})
\end{aligned} \tag{A.7}$$

is always < 0 for all $\beta_k \in \mathbb{R}$, since the function $\frac{exp(u)}{1+exp(u)}$ is always < 1 for all $u \in \mathbb{R}$. This gives that (A.2) is a monotonically decreasing function on $\mathbb{R}$, and β_k^{max} is the only root of $\partial log(L_p)/\partial \beta_k = 0$. Finally, $\partial log(L_p)/\partial \beta_k > 0$ when $-\infty < \beta_k < \beta_k^{max}$; < 0 when $\beta_k^{max} < \beta_k < \infty$, which indicates that β_k^{max} is the global maxima of $log(L_p)$, and L_p. ■

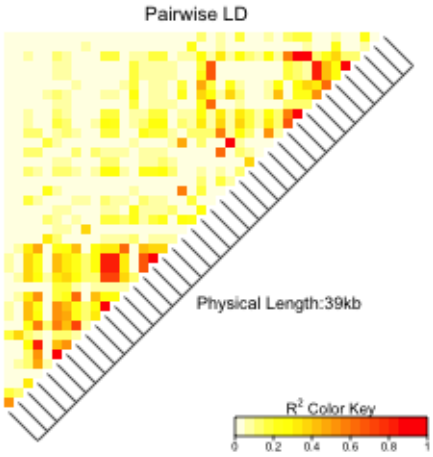
Appendix B

LDheatmaps for Genes in Real Data Analysis

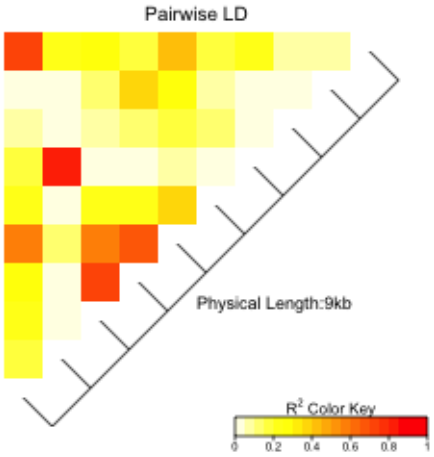
Here we present the LDheatmaps for genes included in our real data analysis of ADNI-1 study. There were indeed SNPs with high pairwise correlation, especially for genes with fewer SNPs. We could have removed part of the SNPs to approach the independence of covariates on one hand, and on the other hand, this would reduce the dimension of covariates and information of m.

The estimated m for ECE1 and NEDD9 in our results from the full set of available SNPs seem to be more plausible based on their LDheatmaps - the pairwise correlation is generally low except for SNPs that are close to each other.

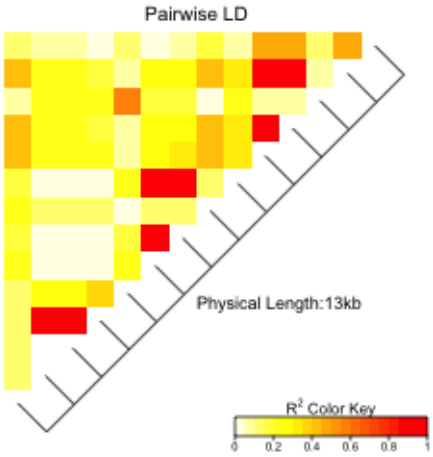
(a) CR1



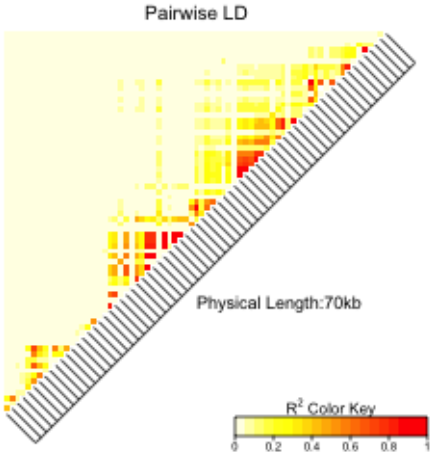
(b) ECE1



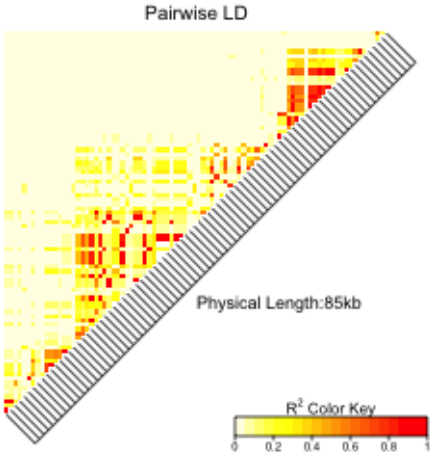
(c) MTHFR



(d) BIN1

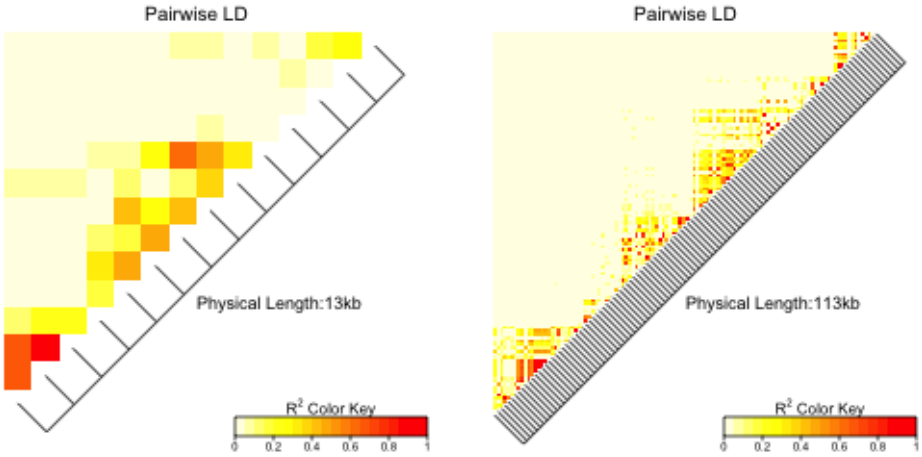


(e) NEDD9



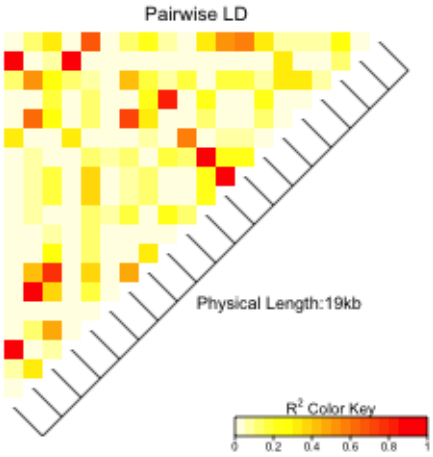
(f) DAPK1

Figure B.1: LDheatmaps for genes included in real data analysis section, using R^2 measure of LD(part 1)

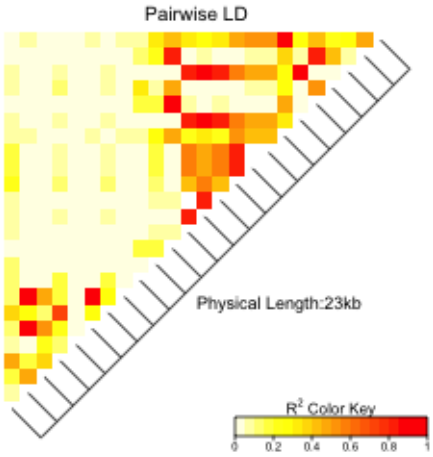


(a) IL33

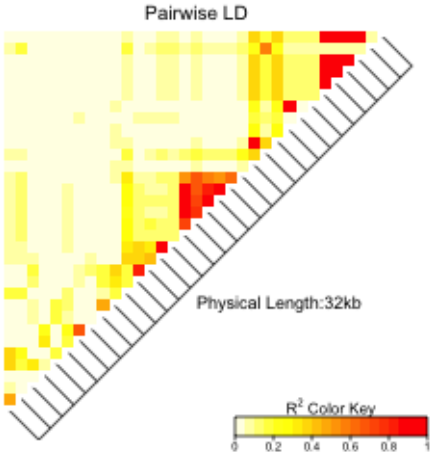
(b) SORCS11



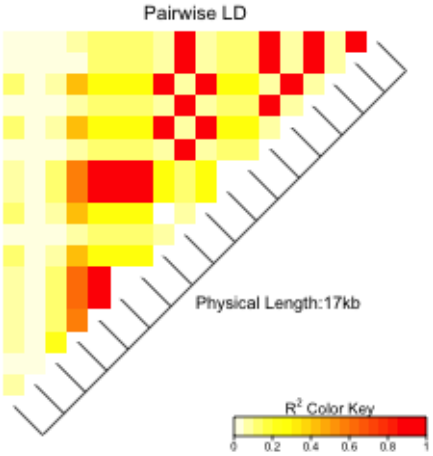
(c) GAB2



(d) PICALM



(e) SORL1



(f) ADAM10

Figure B.2: LDheatmaps for genes included in real data analysis section, using R^2 measure of LD(part 2)

Appendix C

Code

```
library(pso)
library(GA)
n<-1000
mo<-2 #the prior density of beta is log-F
p<-19
simUnmatched = function(n,p,scale=FALSE){
  # n is total sample size, beta1 is value of parameter of interest,
  # p is number of nuisance covariates.wqs
  ConCaseRatio = 4 # assuming 4:1 con:case ratio
  ncase = n/(ConCaseRatio+1); ncon=ncase*ConCaseRatio
  beta = log(rf(p+1,mo/2,mo/2)) # p nuisance params of value 1
  ncov = p+1
  # Simulate cases and controls
  conX = caseX = NULL
  for(i in 1:ncov) {
    conX = cbind(conX,rnorm(ncon,mean=0,sd=1))
    caseX = cbind(caseX,rnorm(ncase,mean=beta[i],sd=1))
  }
  X = rbind(caseX,conX)
  colnames(X) = paste0("x",1:ncov); rownames(X) = NULL
  case = c(rep(1,ncase),rep(0,ncon))
  return(data.frame(case,X))
}
nrep<-20
irep<-1
est.m<-numeric(nrep)
est.m.NM<-numeric(nrep)
est.m.GA<-numeric(nrep)

psoLA<-function(alpha_star.V,m,n.rounds){
  tracer<-matrix(0,nrow=1,ncol=p+4)
  ftracer=0
  i=1
  for (i in 1:n_rounds){
    beta_max<-numeric(p+1)
    for (di in 1:(p+1)){
      X<-XM[,di]
      alpha_star<-alpha_star.V[di]
      dlogPenalisedL<-function(beta){
        sum(X*y-(X*exp(alpha_star+beta*as.numeric(X))
        /(1+exp(alpha_star+beta*as.numeric(X)))))
        -m/2+m*exp(-beta)/(1+exp(-beta))
      }
      beta_max[di]<-uniroot(dlogPenalisedL, c(-20,20))$root
    }
    multi.dimen.logLP_betamax<-function(alpha_star0m0){
      m0=alpha_star0m0[p+2]
      ll<-0
      for (di in 1:(p+1)){
```

```
      alpha_star0<-alpha_star0m0[di]
      X<-XM[,di]
      temp1<-sum(X^2*exp(alpha_star0+beta_max[di]*X)/(1+exp(alpha_star0+beta_max[di]*X)))
      -sum(X^2*(exp(alpha_star0+beta_max[di]*X)/(1+exp(alpha_star0+beta_max[di]*X)))^2)
      temp2<-exp(-beta_max[di])/(1+exp(-beta_max[di]))
      -(exp(-beta_max[di])/(1+exp(-beta_max[di])))^2
      c=temp1+m0*temp2
      LP_di<-sum(y*(alpha_star0+beta_max[di]*as.numeric(X))
      -log(1+exp(alpha_star0+beta_max[di]*as.numeric(X))))
      -log(beta(m0/2,m0/2))-m0/2*beta_max[di]-m0*log(1+exp(-beta_max[di]))-0.5*log(c)
      ll<-ll+LP_di
    }
    ll
  }
  pso.result<-psoptim(par=c(alpha_star.V,m),fn=multi.dimen.logLP_betamax,
  lower=c(rep(-20,p+1), 0), upper = c(rep(10,p+1),15),control=list(trace=100,fnscale=-1,
  maxit=3000,maxit.stagnate=50,s=50,type="SPSO2011"))
  if(abs(pso.result$value-ftracer)>=0.001*abs(ftracer)){
    alpha_star.V=pso.result$par[1:(p+1)]
    m=pso.result$par[p+2]
    tracer<-rbind(tracer,c(as.integer(i),alpha_star.V,m,pso.result$value))
    ftracer<-pso.result$value
    print(tracer[i+1,])
  }else{
    break
  }
 }
 tracer
}
NMLA<-function(alpha_star.V,m,n.rounds){
 tracer<-matrix(0,nrow=1,ncol=p+4)
 ftracer<-0
 i=1
 for (i in 1:n_rounds){
  beta_max<-numeric(p+1)
  for (di in 1:(p+1)){
    X<-XM[,di]
    alpha_star<-alpha_star.V[di]
    dlogPenalisedL<-function(beta){
      sum(X*y-(X*exp(alpha_star+beta*as.numeric(X))
      /(1+exp(alpha_star+beta*as.numeric(X)))))
      -m/2+m*exp(-beta)/(1+exp(-beta))
    }
    beta_max[di]<-uniroot(dlogPenalisedL, c(-20,20))$root
  }
  multi.dimen.logLP_betamax<-function(alpha_star0m0){
    m0=alpha_star0m0[p+2]
    ll<-0
    for (di in 1:(p+1)){
      alpha_star0<-alpha_star0m0[di]
      X<-XM[,di]
      temp1<-sum(X^2*exp(alpha_star0+beta_max[di]*X)/(1+exp(alpha_star0+beta_max[di]*X)))
      -sum(X^2*(exp(alpha_star0+beta_max[di]*X)/(1+exp(alpha_star0+beta_max[di]*X)))^2)
      temp2<-exp(-beta_max[di])/(1+exp(-beta_max[di]))
      -(exp(-beta_max[di])/(1+exp(-beta_max[di])))^2
      c=temp1+m0*temp2
      LP_di<-sum(y*(alpha_star0+beta_max[di]*as.numeric(X))-log(1+exp(alpha_star0
      +beta_max[di]*as.numeric(X))))
      -log(beta(m0/2,m0/2))-m0/2*beta_max[di]-m0*log(1+exp(-beta_max[di]))-0.5*log(c)
      ll<-ll+LP_di
    }
    ll
  }
  opt.result<-optim(par=c(alpha_star.V,m),fn=multi.dimen.logLP_betamax,
  method="Nelder-Mead",control = list(fnscale=-1))
  if(abs(opt.result$value-ftracer)>=0.001*abs(ftracer)){
    alpha_star.V=opt.result$par[1:(p+1)]
    m=opt.result$par[p+2]
    tracer<-rbind(tracer,c(as.integer(i),alpha_star.V,m,opt.result$value))
    ftracer<-opt.result$value
    print(tracer[i+1,])
  }else{
    break
```

```
    }
  }
    tracer
}
GALA<-function(alpha_star.V,m,n.rounds){
  tracer<-matrix(0,nrow=1,ncol=p+4)
  ftracer<-0
  i=1
  for (i in 1:n_rounds){
    beta_max<-numeric(p+1)
    for (di in 1:(p+1)){
      X<-XM[,di]
      alpha_star<-alpha_star.V[di]
      dlogPenalisedL<-function(beta){
        sum(X*y-(X*exp(alpha_star+beta*as.numeric(X))/(1+exp(alpha_star+beta*as.numeric(X)))))
        -m/2+m*exp(-beta)/(1+exp(-beta))
      }
      beta_max[di]<-uniroot(dlogPenalisedL, c(-20,20))$root
    }
    multi.dimen.logLP_betamax<-function(alpha_star0m0){
      m0=alpha_star0m0[p+2]
      ll<-0
      for (di in 1:(p+1)){
        alpha_star0<-alpha_star0m0[di]
        X<-XM[,di]
        temp1<-sum(X^2*exp(alpha_star0+beta_max[di]*X)/(1+exp(alpha_star0+beta_max[di]*X)))
        -sum(X^2*(exp(alpha_star0+beta_max[di]*X)/(1+exp(alpha_star0+beta_max[di]*X)))^2)
        temp2<-exp(-beta_max[di])/(1+exp(-beta_max[di]))
        -(exp(-beta_max[di])/(1+exp(-beta_max[di])))^2
        c=temp1+m0*temp2
        LP_di<-sum(y*(alpha_star0+beta_max[di]*as.numeric(X))-log(1+exp(alpha_star0
        +beta_max[di]*as.numeric(X))))
        -log(beta(m0/2,m0/2))-m0/2*beta_max[di]-m0*log(1+exp(-beta_max[di]))-0.5*log(c)
        ll<-ll+LP_di
      }
      ll
    }
GA <- ga(type = "real-valued", fitness =  function(x) -(-multi.dimen.logLP_betamax(x)),
lower = c(rep(-20,p+1), 0),upper = c(rep(10,p+1), 15), popSize = 50, maxiter = 2000,
run = 50,parallel=4)
if(abs(GA@fitnessValue-ftracer)>=0.001*abs(ftracer)){
  alpha_star.V=GA@solution[1:(p+1)]
  m=GA@solution[p+2]
  tracer<-rbind(tracer,c(as.integer(i),alpha_star.V,m,GA@fitnessValue))
  ftracer<-GA@fitnessValue
  print(tracer[i+1,])
}else{
  break
}
}
  tracer
}
for (irep in 1:nrep){
  simdata<-simUnmatched(n,p)
  y<-simdata$case
  XM<-as.matrix(simdata[,-1])
  #glm1<-glm(y~XM, family=binomial)
  ##Initialization
  ini.alpha<-rep(unname(-5),p+1)
  ini.m<-4
  n_rounds<-400
  tracer1<-psoLA(ini.alpha,ini.m)
  tracer2<-NMLA(ini.alpha,ini.m)
  tracer3<-GALA(ini.alpha,ini.m)
  est.m[irep]<-tracer1[nrow(tracer1),p+3]
  est.m.NM[irep]<-tracer2[nrow(tracer2),p+3]
  est.m.GA[irep]<-tracer3[nrow(tracer3),p+3]
  irep<-irep+1
}
```