Shrinkage Parameter Estimation for Penalized Logistic Regression Analysis of Case-Control Data

by

Ying Yu

B.Sc., The University of British Columbia, 2017

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
Summer 2019

Approval

Name:	**Ying Yu**
Degree:	**Master of Science (Statistics)**
Title:	**Shrinkage Parameter Estimation for Penalized Logistic Regression Analysis of Case-Control Data**
Examining Committee:	**Chair:** Jiguo Cao Associate Professor
	Brad McNeney Senior Supervisor Associate Professor
	Liangliang Wang Supervisor Associate Professor
	Jinko Graham Internal Examiner Professor
Date Defended:	**August 15, 2019**

Abstract

In genetic epidemiology, rare variant case-control studies aim to investigate the association between rare genetic variants and human diseases. Rare genetic variants lead to sparse covariates that are predominately zeros and this sparseness leads to estimators of log-odds-ratio parameters that are biased away from their null value of zero. Different penalized-likelihood methods have been developed to mitigate this sparse-data bias for case-control studies. In this project, we study penalized logistic regression using a class of log-F priors indexed by a shrinkage parameter m to shrink the biased MLE towards zero. We propose a simple method to select the value of m based on a marginal likelihood. The marginal likelihood is maximized by the Monte Carlo EM algorithm. Properties of the proposed method are evaluated in a simulation study, and the method is applied to a real dataset from the ADNI-1 study.

Keywords: Case-control data, Penalized logistic regression, Rare variant association studies, Monte Carlo EM algorithm, Monte Carlo integration, Shrinkage, Alzheimer's Disease

Dedication

In memory of my grandfathers and to my beloved parents Dr. Huimin Yu and Mrs. Ying Xiao who give me love and instill me with the passion for learning.

Acknowledgements

I would first like to express my sincere gratitude to my senior supervisor Dr. Brad McNeney for bringing me to the field of statistical genetics, and for his generous patience, support, and encouragement throughout my M.Sc. studies at Simon Fraser University. I greatly appreciate the effort and insights he has put into this project.

I would also like to express my thanks to my examining committees, Dr. Liangliang Wang, Dr. Jinko Graham and Dr. Jiguo Cao for taking their time to participate in my defence. My deep gratitude goes out to all the staff and faculty in the department of statistics and actuarial science for their great help and support. In addition, I am extremely grateful to have many amazing people inspiring and guiding me throughout my undergraduate studies at The University of British Columbia. Thank you Dr. Lang Wu for sparking my interests in statistics and encouraging me in the pursuit of a Ph.D. degree. Many thanks to Dr. Denise Daley from the Heart and Lung Innovation for having me worked for her research projects and seeing the potential in me.

Furthermore, I would like to thank all my lovely friends and fellow graduate students, Anqi Chen, Haoyao Ruan, Chuyuan Lin, Jingxue Feng, Jiarui Zhang, Yifan Wu, Mengyang Li, Dr. Yuping Yang and Dr. Peijun Sang. The time we spend, the place we go and the weddings we attend make us friends from a young age through whole life. Special thanks to Mr. Wang for his encouragement, good temper and sense of humor, which bring me comfort and happiness all the time.

Perhaps most importantly of all, I would like to express my deepest appreciation to my parents and family for their endless love, support and encouragement over years, even though they may not understand what I am doing for my research 😄.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

In genetic epidemiology, rare variant case-control studies aim to investigate the association between rare genetic variants and human diseases. In the case-control design, the number of cases and controls is fixed and we compare the frequency of the genetic variant between case and control groups. By contrast, in a "prospective" rare variant study, genetic variant status is considered fixed and we compare the frequency of cases between those with and those without the variant allele. The case-control design is sometimes called retrospective, because the exposures, in this case the genetic variant, are measured *after* disease status has been ascertained.

We are interested in disease studies that collect data on many genetic variants, but only analyze genetic associations one variant at a time. To infer single-variant associations we use a logistic regression with a single covariate coded as 0, 1 or 2 copies of the variant allele. The regression parameter is a log-odds-ratio (log-OR) that represents the additive change in the log odds for each additional copy of the genetic variant. These parameters are estimated by maximum likelihood estimation (MLE). Logistic regression applied to retrospective case-control data yields correct inference of odds-ratio parameters, even though it was derived for data from prospective studies [18]. In fact, the prospective logistic regression likelihood can be viewed as a profile likelihood of the retrospective case-control data [19, 6]. The key difference is that, for retrospective data, the intercept term in the regression model is not generally useful and should be ignored.

Rare variant association studies are prone to sparse data bias [7]. Sparse data bias for estimation of a log-OR parameter is a bias away from the null value of zero that arises when the corresponding covariate is predominately zero. Such is the case for covariates that count the number of copies (0, 1 or 2) of the minor allele of a single nucleotide polymorphism (SNP). In this project, we consider penalized-likelihood methods that mitigate sparse data

bias.

1.2 Penalized-likelihood Methods for Case-control Studies

Firth [5] proposes a "preventive" approach to reduce the finite-sample bias of the MLE by a suitable modification of the score function. When applied to logistic regression models for prospective data, the approach is known as Firth logistic regression (FLR) [11]. In FLR, the likelihood is penalized by the Jeffreys prior [13]. Firth shows that the penalization has the desirable effect of shrinking the upwardly-biased MLE towards the origin [6]. Zhang [26] extends FLR to case-control data by introducing a two-sample semiparametric model and applying a Firth-type penalty term to the score function of the model's profile likelihood.

Besides Firth-type methods, a variety of alternative penalties have been proposed for logistic regression, offering more or less shrinkage than the Jeffreys prior [6]. Greenland and Mansournia [8] take a Bayesian view of the penalized likelihood and develop penalized logistic regression based on a class of log-F priors. The family of log-F priors is indexed by the shrinkage parameter m, and imposing a log-$F(m, m)$ prior will shrink the biased MLE towards zero. In the penalization of the likelihood, each regression parameter is assumed to follow a log-$F(m, m)$ prior distribution, except the intercept. The penalty term is the product of independent log-$F(m, m)$ priors. The log-$F(m, m)$ penalized approach can be viewed as a simple extension of the Firth's method, because FLR for a single-covariate model is equivalent to imposing a log-$F(1, 1)$ prior for the regression parameter [8]. For a given m, the log-F penalized-likelihood method can be implemented by fitting a standard logistic regression to a dataset augmented with m pseudo-individuals per covariate [8].

Limited simulation studies have shown that the log-$F(m, m)$ penalized methods outperform other approaches for case-control data [6], but there is no clear guidance on how to choose the shrinkage parameter m. Greenland and Mansournia [8] advocate choosing a log-$F(m, m)$ prior that reflects plausible values of the regression coefficient, but also suggest that Empirical Bayes methods could be used to estimate m from data. In this project, we follow this suggestion to develop an Empirical Bayes method to estimate the shrinkage parameter m for the log-$F(m, m)$ prior used to penalize the logistic regression analysis for case-control data.

1.3 Overview of the Project

This project is organized as follows. In Chapter 2, I will begin with an overview of our methodology. I then describe the approach for selecting m based on a marginal likelihood

obtained by integrating the random log-OR parameters out of the joint distribution of the case-control data and random effects. Next, I will illustrate the use of the Monte Carlo and EM approaches to maximize the marginal likelihood. In Chapter 3, simulation studies are conducted to evaluate the proposed method with different number of genetic variants. Next, Chapter 4 discusses the application of our method to a real dataset from the ADNI-1 study. To end this project, I provide some concluding remarks and discussion in Chapter 5.

Chapter 2

Methodology

This chapter describes a method for choosing the shrinkage parameter, m, for the log-$F(m, m)$ prior used to penalize the case-control likelihood. Though motivated by the Empirical-Bayes suggestion of Greenland and Mansournia [8], we emphasize that our approach is frequentist in nature, and that estimation of m is in the spirit of shrinkage parameter selection.

2.1 Overview of Methods

In this chapter we develop an Empirical Bayes (EB) method to estimate the shrinkage parameter m. Here we give an outline of the approach, with details to follow in subsequent sections. In models that include multiple independent and identically distributed (IID) realizations of a random effect, EB methods can be used to estimate the parameters of the random effect distribution. For example, if we had multiple independent case-control studies of the same disease and same genetic variant, we could combine data from these studies to estimate m. However, in this project, we gain replicates by combining data from multiple independent genetic variants from the *same* case-control study. To do so we must assume that the regression effects of each genetic variant are IID realizations from the same log-$F(m, m)$ distribution. Thus, in our context, the random effects are the regression coefficients for the genetic variants, and so information about the hyperparameter m accrues from multiple realizations of the genetic effects which are regression coefficients for individual variants.

The EB estimate of m is the maximizer of a "marginal likelihood", obtained by integrating the random effects out of the joint distribution of the genetic data and random effects. We assume that the genetic data are independent and that the random effects are independent. The conditional distribution of the data given the random effects is a product over independent genetic variants, with the kth genetic variant contributing a distribution that depends on a fixed intercept α_k^* and a random slope β_k. The random slopes are assumed to be IID according to a log-$F(m, m)$ distribution. These slopes are integrated out

of the joint distribution of the data and the random effects, leaving a marginal likelihood $L(\boldsymbol{\alpha}^*, m)$ that is a function of m and the intercept terms, $\boldsymbol{\alpha}^* = (\alpha_1^*, \ldots, \alpha_K^*)$. The EB estimate of m takes on the value $\hat{m}$ at the maximum of the marginal likelihood over m and $\boldsymbol{\alpha}^*$.

It is not possible to analytically evaluate the integral over the random slope terms, and so we are faced with the problem of maximizing a function that we cannot evaluate. We consider two solutions to this problem. The most straightforward is to estimate the marginal likelihood for m and $\boldsymbol{\alpha}^*$ by Monte Carlo integration. The drawback to this approach is that it is computationally expensive. An alternative is to treat the random slopes as missing data and devise an EM algorithm to maximize the marginal likelihood. This trick involves viewing the marginal likelihood as an "observed-data" likelihood in the EM algorithm. The "complete-data" likelihood would be given by the joint distribution of genetic data and the random effects. Conveniently, the EM algorithm does not require evaluation of the observed-data likelihood. This is good because as noted above the observed-data or marginal likelihood of m and $\boldsymbol{\alpha}^*$ cannot be evaluated analytically. However, it can be difficult to implement the E- or the M-step. We opt for a hybrid of the Monte Carlo and EM approaches. First, for fixed m, we devise an EM algorithm to maximize the marginal likelihood $L(\boldsymbol{\alpha}^*, m)$ over $\boldsymbol{\alpha}^*$; call the maximizer $\hat{\boldsymbol{\alpha}}^*(m) = \left(\hat{\alpha}_1^*(m), ..., \hat{\alpha}_K^*(m)\right)$. Inserting $\hat{\boldsymbol{\alpha}}^*(m)$ into the marginal likelihood gives a profile likelihood $L(\hat{\boldsymbol{\alpha}}^*(m), m)$. Second, we evaluate $L(\hat{\boldsymbol{\alpha}}^*(m), m)$ by Monte Carlo integration over a grid of values for m and find the maximizer $\hat{m}$.

2.2 Marginal Likelihood for Selecting m

In this section we derive the marginal likelihood for selecting the shrinkage parameter m, obtained by integrating the random effects out of the joint distribution of the genetic data and random effects. The joint distribution of the genetic data and random effects will be referred to as the complete-data likelihood, and the marginal likelihood obtained by integrating out the random effects will be referred to as the observed-data likelihood [3]. In our case the distribution of the case-control data includes an infinite-dimensional nuisance parameter (see Section 2.2.1 below) that makes the complete-data likelihood difficult to work with. A more convenient starting point for deriving a marginal likelihood for m is a *profile* complete-data likelihood, obtained by profiling the nuisance parameter out of the complete-data likelihood. In the sub-sections that follow we describe the two components of the profile complete-data likelihood: (i) the profile likelihood for case-control data and (ii) the random effects distribution, log-$F(m, m)$. By starting our derivation with a profile complete-data likelihood, we technically cannot call its integral a marginal likelihood. However, we ignore this technicality to keep the exposition as simple as possible. That is, throughout we refer to the integral of the profile complete-data likelihood over the random

effects as simply the marginal likelihood.

2.2.1 Profile Likelihood for Case-control Data

Consider a case-control study of association between a rare genetic variant (i.e. SNP) and disease status, in which the n_0 controls are indexed by $i = 1, ..., n_0$ and n_1 cases are indexed by $i = n_0 + 1, ..., n$ for $n = n_0 + n_1$. Let Y_i be a binary response indicating the disease status and X_i be the covariate data for subject i, and let β be the log-OR parameter of the model.

Under the case-control sampling design, in which we assume the binary outcome variable $\boldsymbol{Y} = \{Y_i\}_{i=1}^n$ is fixed and the covariate $\boldsymbol{X} = \{X_i\}_{i=1}^n$ is random, we select cases and controls from the population and the values of covariates are then measured conditionally for the subjects which are selected. Several parametrizations of case-control likelihood are available in the literature [2, 18, 19]. For example, Qin and Zhang [19] expressed the likelihood in terms of a two-sample semi-parametric model in which $\{X_1, ..., X_{n_0}\}$ are independent with density $P(X_i|Y_i = 0) = g(X_i)$ and $\{X_{n_0+1}, ..., X_{n_0+n_1}\}$ are independent with density $P(X_i|Y_i = 1) = h(X_i) = c(\beta, g)\exp(X_i\beta)g(X_i)$, where $c(\beta, g)$ is a normalizing constant. Then the likelihood is given by

$$\begin{aligned} L(\beta, g) &= \prod_{i=1}^{n_0} P(X_i|Y_i = 0) \prod_{i=n_0+1}^{n_0+n_1} P(X_i|Y_i = 1) \\ &= \prod_{i=1}^{n_0} g(X_i) \prod_{i=n_0+1}^{n_0+n_1} c(\beta, g)\exp(X_i\beta)g(X_i). \end{aligned} \tag{2.1}$$

The infinite-dimensional nuisance parameter g makes the case-control likelihood $L(\beta, g)$ difficult to derive, and maximize to find the MLE of β. It is more convenient to obtain the MLE of β by maximizing the profile likelihood for case-control data [18]:

$$\begin{aligned} L(\alpha^*, \beta) &= f(\boldsymbol{X}|\alpha^*, \beta) \\ &= \prod_{i=1}^{n_0} \frac{1}{1 + \exp(\alpha^* + X_i\beta)} \prod_{i=n_0+1}^{n_0+n_1} \frac{\exp(\alpha^* + X_i\beta)}{1 + \exp(\alpha^* + X_i\beta)} \\ &= \prod_{i=1}^{n} \frac{\exp(Y_i(\alpha^* + X_i\beta))}{1 + \exp(\alpha^* + X_i\beta)}, \end{aligned} \tag{2.2}$$

where $\alpha^* = \alpha + \log\left(\frac{n_1}{n_0}\right) - \log\left(\frac{P(D=1)}{P(D=0)}\right)$, α is the intercept term in the logistic regression model for $P(Y = 1|X)$, and $P(D = 1)$ and $P(D = 0)$ are the population probabilities of having and not having the disease, respectively.

Qin and zhang [19] have shown how to obtain the profile case-control likelihood $L(\alpha^*, \beta)$ as follows. First, over-parametrize $L(\beta, g)$ from equation (2.1) by including α^* as a parameter to get the case-control likelihood $L(\alpha^*, \beta, g)$; second, for fixed values of (α^*, β), search over the values of g to find $\hat{g}(\alpha^*, \beta)$ that maximizes the likelihood with constraints on the parameters (α^*, β, g) [6]; finally, insert $\hat{g}(\alpha^*, \beta)$ into $L(\alpha^*, \beta, g)$ to obtain $L(\alpha^*, \beta) \equiv L(\alpha^*, \beta, \hat{g}(\alpha^*, \beta))$. An alternative derivation of the profile likelihood due to Hosmer et.al [12] is given in Appendix A.

Note that the profile likelihood $L(\alpha^*, \beta)$ for case-control data is in the same form as a prospective likelihood. It can be viewed as the likelihood obtained when we ignore the case-control sampling design and pretend the data were collected in a prospective study [24]. Prentice and Pyke [18] have shown that the MLE of β can be obtained by maximizing the likelihood $L(\alpha^*, \beta)$, as well as the asymptotic variance matrices. In other words, the MLE of the odds ratio parameters under the case-control sampling design can be obtained by applying the standard logistic regression model as if the data were collected in a prospective cohort study [18, 19, 24].

2.2.2 Log-$F(m, m)$ Random Effects

The log-$F(m, m)$ distribution is obtained by taking the logarithm of a random variable having F-distribution with m numerator and m denominator degrees of freedom. The log-$F(m, m)$ prior density of β is [14]

$$f(\beta|m) = \frac{1}{\text{Beta}(\frac{m}{2}, \frac{m}{2})} \frac{\exp(-\frac{m}{2}\beta)}{(1 + \exp(-\beta))^m}. \tag{2.3}$$

The log-$F(m, m)$ distribution has a symmetrical bell shape centered at zero. The variance of a log-$F(m, m)$ distribution decreases with an increase of m. When m is sufficiently large, the distribution becomes a point mass at zero.

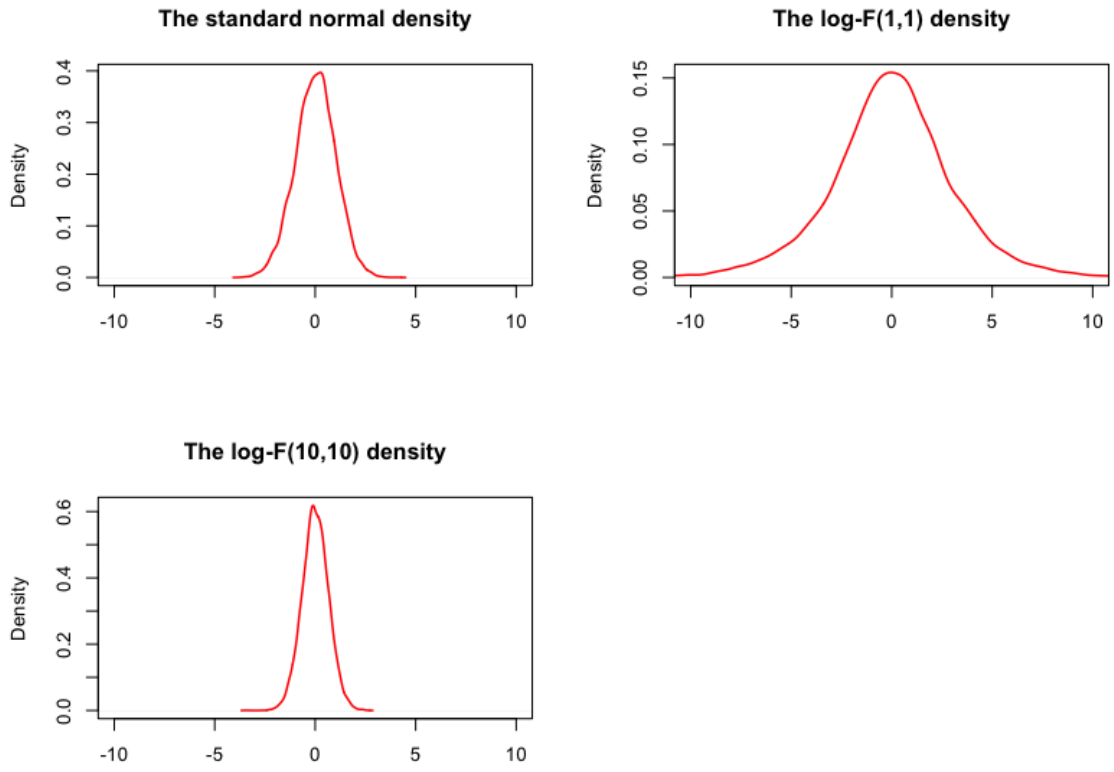


Figure 2.1: Upper-left panel: the standard normal density; Upper-right panel: the log-$F(1,1)$ density; Lower-left panel: the log-$F(10,10)$ density.

Figure 2.1 compares the density curves of a standard normal distribution, a log-$F(1,1)$ distribution and a log-$F(10,10)$ distribution. As shown in Figure 2.1, the variance of the log-$F(1,1)$ distribution is larger than the variance of the standard normal distribution whereas the the log-$F(10,10)$ distribution has a smaller variance than the standard normal distribution.

2.2.3 Marginal Likelihood for m

Suppose we have K genetic variants, encoded in covariates $\boldsymbol{X}_i = (X_{i1}, ..., X_{iK})$ for subject i. Let $\boldsymbol{X}_{.k}$ denote the k^{th} column of the design matrix $\boldsymbol{X}$ such that $\boldsymbol{X}_{.k} = (X_{1k}, ..., X_{nk})^T$, then the profile likelihood of the k^{th} variant is $L(\alpha_k^*, \beta_k) = f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_k)$, where β_k is the log-OR parameter and α_k^* is the intercept term for the k^{th} profile likelihood. The product of the profile case-control likelihood and the random effects distribution is referred to as the

profile complete-data likelihood:

$$L(\boldsymbol{\alpha}^*, \boldsymbol{\beta}) f(\boldsymbol{\beta}|m) = \prod_{k=1}^{K} L(\alpha_k^*, \beta_k) f(\beta_k|m), \tag{2.4}$$

where $\boldsymbol{\beta} = (\beta_1, ..., \beta_K)^T$ and $\boldsymbol{\alpha}^* = (\alpha_1^*, ..., \alpha_K^*)$. Here we assume the variants are independent, so the profile complete-data likelihood is a product over K variants. The marginal likelihood for $(\boldsymbol{\alpha}^*, m)$ is obtained by integrating out $\boldsymbol{\beta}$ from the profile complete-data likelihood, that is

$$\begin{aligned} L(\boldsymbol{\alpha}^*, m) &= \int L(\boldsymbol{\alpha}^*, \boldsymbol{\beta}) f(\boldsymbol{\beta}|m) d\boldsymbol{\beta} \\ &= \int \prod_{k=1}^{K} L(\alpha_k^*, \beta_k) f(\beta_k|m) d\boldsymbol{\beta} \\ &= \int \prod_{k=1}^{K} f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_k) f(\beta_k|m) d\beta_k \\ &= \prod_{k=1}^{K} \int f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_k) f(\beta_k|m) d\beta_k. \end{aligned} \tag{2.5}$$

Stated equivalently,

$$L(\boldsymbol{\alpha}^*, m) = \prod_{k=1}^{K} L(\alpha_k^*, m), \tag{2.6}$$

where the contribution of the k^{th} variant to $L(\boldsymbol{\alpha}^*, m)$ is

$$\begin{aligned} L(\alpha_k^*, m) &= \int f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_k) f(\beta_k|m) d\beta_k \\ &= \int \prod_{i=1}^{n} \frac{\exp(Y_i(\alpha_k^* + X_{ik}\beta_k))}{1 + \exp(\alpha_k^* + X_{ik}\beta_k)} \frac{1}{\text{Beta}(\frac{m}{2}, \frac{m}{2})} \frac{\exp(-\frac{m}{2}\beta_k)}{(1 + \exp(-\beta_k))^m} d\beta_k \end{aligned} \tag{2.7}$$

based on (2.2) and (2.3). We select the value of m that maximizes the marginal log-likelihood,

$$l(\boldsymbol{\alpha}^*, m) = \sum_{k=1}^{K} \log[L(\alpha_k^*, m)] = \sum_{k=1}^{K} l(\alpha_k^*, m). \tag{2.8}$$

Maximization is done in two stages. First, for fixed m, we maximize each $l(\alpha_k^*, m)$ over α_k^* to get $L(\hat{\alpha_k^*}(m), m)$ and $l(\hat{\boldsymbol{\alpha}}^*(m), m) = \sum_{k=1}^{K} \log[L(\hat{\alpha_k^*}(m), m)]$, where $\hat{\boldsymbol{\alpha}}^*(m) = \left(\hat{\alpha_1^*}(m), ..., \hat{\alpha_K^*}(m)\right)$. Second, we maximize $l(\hat{\boldsymbol{\alpha}}^*(m), m)$ over m. To keep computations manageable, we restrict m to a grid at values, $m = 1, 2, ..., M$, and select the m that maximizes $l(\hat{\boldsymbol{\alpha}}^*(m), m)$ over this grid.

2.3 Maximization by a Hybrid of the Monte Carlo and EM Algorithm

For a fixed value of m and k, the estimate $\hat{\alpha_k^*}(m)$ can be obtained by maximizing $l(\alpha_k^*, m)$ with respect to α_k^*. However, it is extremely difficult to solve the integral in (2.7). A commonly used approach for maximizing marginal, or observed-data log-likelihoods is the expectation-maximization (EM) algorithm [4]. In this section, we first introduce the general procedure of the EM algorithm. Next, we present the implementation of the Monte Carlo EM algorithm to maximize the marginal log-likelihood $l(\alpha_k^*, m)$ over α^* for a fixed m,where the expectation in the E-step is evaluated numerically through Monte Carlo integration. Finally, we obtain a profile likelihood for m.

2.3.1 General Procedure of the EM Algorithm

The EM algorithm is an iterative method to maximize the likelihood of parameters in a statistical model, where the model involves unobserved latent variables in addition to observed data and unknown parameters. This method was introduced by Dempster, Laird and Rubin [3] as a way to compute the MLE with missing data. The unobserved latent variables can be either missing values among the data, or the variables that the model depends on but are not directly observed. The EM algorithm consists of two steps. In the expectation (E)-step, it defines the expectation of the log-likelihood for the complete data given the current estimate of the parameters. In the maximization (M)-step, it updates the estimate of the parameters by maximizing the expected log-likelihood found in the E-step. These updated parameter estimates are then used to determine the expectation in the E-step of the next iteration. Starting at some initial value for the parameters, the EM algorithm alternates between performing the E-step and M-step, updating the parameter estimates in each iteration until convergence [25].

Given a statistical model with unknown parameters θ, let $\boldsymbol{X}$ denote a set of observed data and $\boldsymbol{Z}$ denote a set of missing values or unobserved latent variables. The MLE of θ is determined by maximizing the marginal, or observed log-likelihood $l_o(\theta) = \log[f(\boldsymbol{X}|\theta)]$. The EM algorithm finds the MLE of θ is based on the complete data log-likelihood $l_c(\theta) = \log[f(\boldsymbol{X}, \boldsymbol{Z}|\theta)]$, where the distribution of $\boldsymbol{Z}$ is conditional on $\boldsymbol{X}$ and the current estimate of θ. The EM algorithm is summarized as follows:

Algorithm 1: The EM Algorithm

1. Initialize the parameters $\theta^{(0)}$.
2. E-step: define the expected complete log-likelihood with respect to the current conditional distribution of $\boldsymbol{Z}$

$$Q\left(\theta|\theta^{(p)}\right) = \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X},\theta^{(p)}}(\log[f(\boldsymbol{X},\boldsymbol{Z}|\theta)]).$$

3. M-step: update the estimate of θ by maximizing the expectation

$$\theta^{(p+1)} = \underset{\theta}{\operatorname{argmax}}\, Q\left(\theta|\theta^{(p)}\right).$$

4. Repeat Steps 2 and 3 until convergence.

The EM algorithm is an example of a minorize-maximize algorithm [15]. Up to a constant, the function $Q(\theta|\theta^{(p)})$ minorizes the observed-data log-likelihood, $l_o(\theta)$ and is equal to $l_o(\theta)$ at $\theta = \theta^{(p)}$, as illustrated in Figure 2.2. From the figure, it is clear that moving from $\theta^{(p)}$ to the maximizer of $Q(\theta|\theta^{(p)})$ increases not only $Q(\theta|\theta^{(p)})$ but also $l_o(\theta)$. This is called the ascent property of the EM algorithm. Regularity conditions that guarantee this ascent property are given in Wu [25].

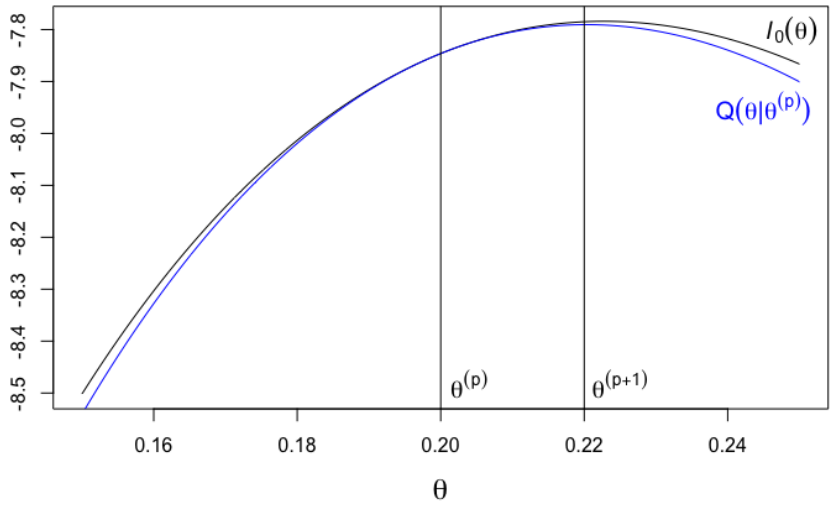


Figure 2.2: Relation of the EM algorithm to the log-likelihood function.

2.3.2 Maximization over α_k^* for Fixed m

To maximize $l(\alpha_k^*, m)$ for k^{th} covariate, we consider an EM algorithm. In our setting, $\boldsymbol{X}_{.k}$ is a set of observed data and β_k is the unobserved latent variable. For a fixed value of m and k, the EM algorithm is based on the profile complete-data log-likelihood $\log\left[f(\boldsymbol{X}_{.k}, \beta_k|\alpha_k^*, m)\right]$.

In the E-step of the $(p+1)^{th}$ iteration, one has to determine

$$\begin{aligned} Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right) &= \mathbb{E}_{\beta_k|\boldsymbol{X}_{.k},\alpha_k^{*(p)},m}(\log[f(\boldsymbol{X}_{.k}, \beta_k|\alpha_k^*, m)]) \\ &= \int \log\left[f(\boldsymbol{X}_{.k}, \beta_k|\alpha_k^*, m)\right] f(\beta_k|\boldsymbol{X}_{.k}, \alpha_k^{*(p)}, m) d\beta_k. \end{aligned} \tag{2.9}$$

The profile complete-data log-likelihood, $\log\left[f(\boldsymbol{X}_{.k}, \beta_k|\alpha_k^*, m)\right]$, is integrated over the density $f(\beta_k|\boldsymbol{X}_{.k}, \alpha_k^{*(p)}, m)$, which is calculated using the current trial values of the parameters. Therefore, $Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right)$ is a conditional expectation of $\log\left[f(\boldsymbol{X}_{.k}, \beta_k|\alpha_k^*, m)\right]$, given the observed data $\boldsymbol{X}_{.k}$ and the estimate $\alpha_k^{*(p)}$ from the previous iteration and the fixed m. Equivalently, $Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right)$ is the weighted average of the profile complete-data log-likelihood, using $f(\beta_k|\boldsymbol{X}_{.k}, \alpha_k^{*(p)}, m)$ as weights.

Note that the profile complete-data likelihood $f(\boldsymbol{X}_{.k}, \beta_k|\alpha_k^*, m)$ is comprised of the profile likelihood for the observed data and the unobserved latent variable distribution, therefore

$$\log\left[f(\boldsymbol{X}_{.k}, \beta_k|\alpha_k^*, m)\right] = \log[f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_k) f(\beta_k|m)]. \tag{2.10}$$

The density $f(\beta_k|\boldsymbol{X}_{.k}, \alpha_k^{*(p)}, m)$ denotes the posterior distribution

$$f(\beta_k|\boldsymbol{X}_{.k}, \alpha_k^{*(p)}, m) = \frac{f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)}, \beta_k) f(\beta_k|m)}{\int f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)}, \beta_k) f(\beta_k|m) d\beta_k}, \tag{2.11}$$

which is obtained after applying the Bayes' rule. Inserting (2.10) and (2.11) into (2.9), $Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right)$ simplifies to

$$Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right) = \int \log[f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_k) f(\beta_k|m)] f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)}, \beta_k) f(\beta_k|m) d\beta_k. \tag{2.12}$$

The EM algorithm maximizes $Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right)$ to find the next new value of α_k^*. In the M-step of the $(p+1)^{th}$ iteration, the value of the parameter is updated as follows

$$\alpha_k^{*(p+1)} = \underset{\alpha_k^*}{\operatorname{argmax}}\, Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right). \tag{2.13}$$

2.3.3 Estimation by the Monte Carlo EM Algorithm

In general, carrying out the E-step (2.12) is troublesome since the integral cannot be solved analytically. However, we can approximate the integral numerically through Monte Carlo (MC) methods, which is the so-called Monte Carlo EM (MCEM) algorithm introduced by Wei and Tanner [23]. The method relies on MC integration in the E-step via sampling from the posterior distribution [16, 23]. Note that the integral in (2.12) is with respect to the latent variable β_k. If we obtain a sample of $\beta_{k1}, ..., \beta_{kN}$ from the distribution $f(\beta_k|m)$, MC integration yields the approximation

$$\begin{aligned} Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right) &\approx Q_{MC}\left(\alpha_k^*|\alpha_k^{*(p)}, m\right) \\ &= \frac{1}{N}\sum_{j=1}^{N} \log[f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_{kj}) f(\beta_{kj}|m)] f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)}, \beta_{kj}) \\ &= \frac{1}{N}\sum_{j=1}^{N} (\log[f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_{kj})] + \log[f(\beta_{kj}|m)]) f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)}, \beta_{kj}). \end{aligned} \tag{2.14}$$

The function $Q_{MC}\left(\alpha_k^*|\alpha_k^{*(p)}, m\right)$ in (2.14) will converge pointwise to $Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right)$ in (2.12) as N increases by the law of large numbers [16]. The M-step of the MCEM algorithm maximizes (2.14) as usual. Note that $\log[f(\beta_{kj}|m)]$ is independent from the parameter α_k^*, so maximizing (2.14) is equivalent to maximizing

$$\frac{1}{N}\sum_{j=1}^{N} \log[f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_{kj})] f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)}, \beta_{kj}). \tag{2.15}$$

2.3.4 Profile Likelihood for m

We denote $\hat{\alpha_k^*}(m)$ as the output of the MCEM algorithm. Having obtained the estimate $\hat{\alpha_k^*}(m)$, the next task is to insert this value into $L(\alpha_k^*, m) = \int f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_k) f(\beta_k|m) d\beta_k$ to obtain $L(\hat{\alpha_k^*}(m), m) = \int f(\boldsymbol{X}_{.k}|\hat{\alpha_k^*}(m), \beta_k) f(\beta_k|m) d\beta_k$. Though we have maximized $L(\alpha_k^*, m)$ with respect to α_k^* with the EM algorithm, we have done so without evaluating it.

We can interpret $L(\hat{\alpha_k^*}(m), m)$ as the expectation of $f(\boldsymbol{X}_{.k}|\hat{\alpha_k^*}(m), \beta_k)$ with respect to the log-$F(m, m)$ density $f(\beta_k|m)$. An estimate of $L(\hat{\alpha_k^*}(m), m)$ using MC integration is

$$\begin{aligned} L(\hat{\alpha_k^*}(m), m) &\approx L_{MC}(\hat{\alpha_k^*}(m), m) \\ &= \frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{X}_{.k}|\hat{\alpha_k^*}(m), \beta_{kj}), \end{aligned} \tag{2.16}$$

where $\beta_{k1}, ..., \beta_{kN}$ are simulated from the log-$F(m, m)$ distribution with density $f(\beta_k|m)$.

According to equation (2.8), $l(\boldsymbol{\alpha}^*, m) = \sum_{k=1}^{K} \log[L(\alpha_k^*, m)]$, and an estimate of the profile log-likelihood $l(\hat{\boldsymbol{\alpha}}^*(m), m)$ can be obtained as a sum of these estimates

$$\begin{aligned} l(\hat{\boldsymbol{\alpha}}^*(m), m) &= \sum_{k=1}^{K} \log[L(\hat{\alpha_k^*}(m), m)] \\ &\approx \sum_{k=1}^{K} \log[L_{MC}(\hat{\alpha_k^*}(m), m)] \\ &= l_{MC}(\hat{\boldsymbol{\alpha}}^*(m), m). \end{aligned} \tag{2.17}$$

We will plot th estimated profile log-likelihood $l_{MC}(\hat{\boldsymbol{\alpha}}^*(m), m)$ versus m and select the m that maximizes $l_{MC}(\hat{\boldsymbol{\alpha}}^*(m), m)$ over a grid of values for $m = 1, 2, ..., M$.

2.4 Summary

Combining Section 2.2 and 2.3, our methodology can be summarized as follows:

Algorithm 2: Summary of the methodology

For m **in** $1:M$**, do:**
For k **in** $1:K$**, do:**

1. Maximize $l(\alpha_k^*, m)$ over α^* by the MCEM algorithm:

   (a) Initialize the value of α_k^* and N.

   (b) Generate $\beta_{k1}, ..., \beta_{kN}$ independently from $f(\beta_k|m)$.

   (c) E-step: estimate $Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right)$ by

   $$Q_{MC}\left(\alpha_k^*|\alpha_k^{*(p)}, m\right) = \frac{1}{N}\sum_{j=1}^{N} \log[f(\boldsymbol{X}_{.k}|\alpha_k^*, \beta_{kj}) f(\beta_{kj}|m)] f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)}, \beta_{kj}).$$

   (d) M-step: maximize $Q_{MC}\left(\alpha_k^*|\alpha_k^{*(p)}, m\right)$ to get

   $$\alpha_k^{*(p+1)} = \underset{\alpha_k^*}{\operatorname{argmax}}\, Q\left(\alpha_k^*|\alpha_k^{*(p)}, m\right).$$

   (e) Repeat Steps (c)-(e) until convergence and obtain $\hat{\alpha_k^*}(m)$.

2. Estimate $L(\hat{\alpha_k^*}(m), m)$ by MC integration:

   $$L_{MC}(\hat{\alpha_k^*}(m), m) = \frac{1}{N}\sum_{j=1}^{N} f(\boldsymbol{X}_{.k}|\hat{\alpha_k^*}(m), \beta_{kj}),$$

   where $\beta_{k1}, ..., \beta_{kN}$ are sampled independently from $f(\beta_k|m)$.

EndFor
Obtain an estimate of the profile log-likelihood:

$$\begin{aligned} l(\hat{\boldsymbol{\alpha}}^*(m), m) &\approx \sum_{k=1}^{K} \log[L_{MC}(\hat{\alpha_k^*}(m), m)] \\ &= l_{MC}(\hat{\boldsymbol{\alpha}}^*(m), m). \end{aligned}$$

EndFor
Plot $l_{MC}(\hat{\boldsymbol{\alpha}}^*(m), m)$ versus m and select m to be the maximizer of $l_{MC}(\hat{\boldsymbol{\alpha}}^*(m), m)$.

2.5 Computational Considerations

Recall that we intend to maximize (2.15) in the M-step of the MCEM. The `R` function `optimize()` may be used to find the maximum of (2.15) with respect to α_k^*. However, we found `optimize()` to be too slow to be practical, and instead resorted to the weighted logistic regression approach described next.

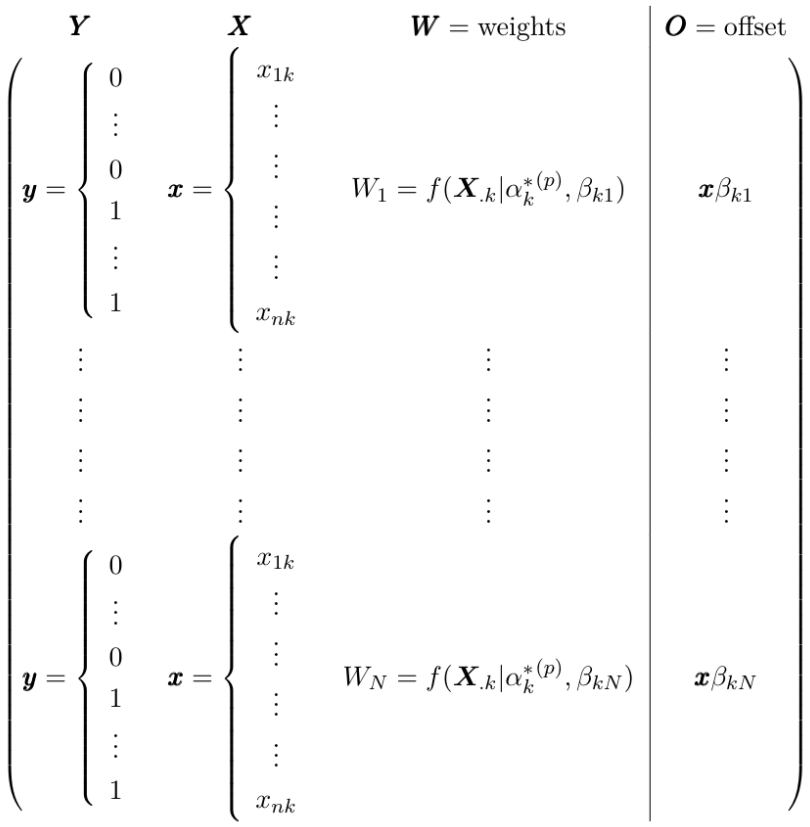


Figure 2.3: $\boldsymbol{Y}_{(Nn\times 1)}$ is a vector containing N replicates of $\boldsymbol{y}$ and $\boldsymbol{X}_{(Nn\times 1)}$ is a vector containing N replicates of $\boldsymbol{x}$. $\boldsymbol{W}$ stands for the weights for each Monte Carlo replicate such that $W_j = f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)}, \beta_{kj})$ and the offset term $\boldsymbol{O} = \{\boldsymbol{x}\beta_{kj}\}_{j=1}^N$.

Equation (2.15) is to be maximized over α_k^*. This equation is a weighted average of logistic regression likelihoods, with weights given by the density values $f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)}, \beta_{kj})$. Each likelihood is itself a sum over the n subjects in the dataset. Our approach is to write equation (2.15) as a weighted likelihood comprised of $N \times n$ observations and use standard logistic regression software to maximize over α_k^*. One way to do this is to "stack" the response vector and covariates N times over as illustrated in Figure 2.3 and associate with each observation in this augmented dataset a weight and an offset. The weight for each observation in the j^{th} replicate of the dataset is the weight $f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)}, \beta_{kj})$ from the weighted average in equation (2.15). The offsets account for known quantities in the logistic model. In particular, the

linear prediction in the logistic model for observation i in the j^{th} replicate of the dataset is $\alpha^* + x_{ik}\beta_{kj}$, where β_{kj} is drawn from the log-$F(m, m)$ distribution and is considered fixed in equation (2.15). Thus the term $x_{ik}\beta_{kj}$ is a known offset.

By constructing the augmented dataset in Figure 2.3, maximizing (2.15) over α_k^* is equivalent to estimating the intercept of a logistic regression and we can use standard logistic regression software, such as `glm()` in `R`, to do this.

Chapter 3

Simulation Study

The simulation study is conducted to determine how many genetic variants are required to get a reliable estimate of m. In this chapter, we will introduce the design of our simulation study and present our simulation results.

3.1 Design of Study

In this section, we set up simulation studies to compare the performance of our method introduced in Chapter 2 using different numbers of genetic variants, denoted as K. We consider $K = 10, 20, 30, 40, 50$. We set $n = 200$ (small to medium sample size), and $N = 1000$ Monte Carlo replicates for both the MCEM algorithm and the MC estimates of the likelihood.

In each scenario, we intend to generate 200 balanced case-control datasets ($n_0 : n_1 = 1 : 1$) with $Y_1 = ... = Y_{100} = 0$ and $Y_{101} = ... = Y_{200} = 1$. The log-OR parameter of each genetic variant β_k, for $k = 1, .., K$, is generated by taking the log of a random variable sampled from the $F(m, m)$ distribution independently. The covariate data $\boldsymbol{X}$ is generated as follows. Following [26], the conditional density function for the covariate in the controls and cases are

$$P(X = x|Y = 0) = g(x) \text{ and} \tag{3.1}$$

$$P(X = x|Y = 1) = h(x) = c(\beta, g) \exp(x\beta)g(x), \tag{3.2}$$

respectively. Here $c(\beta, g)$ is a constant term respect to x. If we take $g(x)$ to be the standard normal density function [26]

$$\begin{aligned} g(x) &= \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \\ &\propto e^{-x^2/2}, \end{aligned} \tag{3.3}$$

and for a given β, the conditional density function for the covariate in cases is

$$\begin{aligned} P(X = x|Y = 1) = h(x) &\propto \exp(x\beta)g(x) \\ &\propto \exp(-\frac{x^2}{2} + x\beta) \\ &= \exp(-\frac{1}{2}(x^2 - 2x\beta + \beta^2) + \frac{\beta^2}{2}) \\ &= \exp(-\frac{1}{2}(x - \beta)^2). \end{aligned} \tag{3.4}$$

Thus, h(x) is the normal density function with mean β and standard deviation 1. Therefore, in our simulation study, the covariates $X_1, ..., X_{100}$ in the control group are sampled from the standard normal distribution $N(0, 1)$ and the covariates $X_{101}, ..., X_{200}$ in the case group are sampled from the normal distribution $N(\beta, 1)$ [24]. Our method is then applied to all the generated datasets to select m.

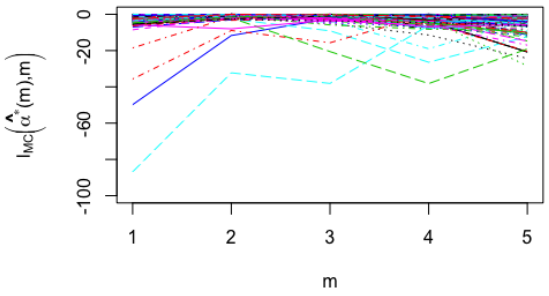
3.2 Simulation Results

In each scenario,200 independent balanced case-control datasets (i.e. $n_0 : n_1 = 1 : 1$) are generated through the simulation procedures described above. For each generated dataset, we plot $l_{MC}(\hat{\boldsymbol{\alpha}}^*(m), m)$ (2.17) versus m and select m as the maximizer of the log-likelihood function over the grid $\{1, 2, 3, 4, 5\}$. The true value of m is set to be 2 because it is in the center of the grid. The performance of our method is summarized by the proportion of 200 simulations that are able to correctly identify the true value of m.

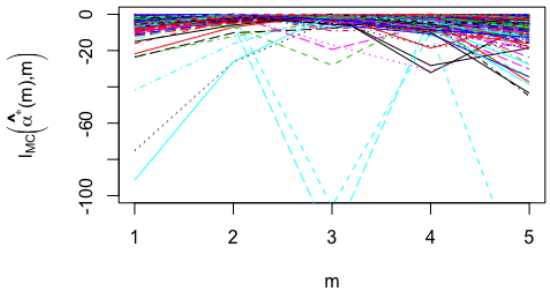
Figure 3.1 demonstrates the estimated profile log-likelihood $l_{MC}(\hat{\boldsymbol{\alpha}}^*(m), m)$ constructed over 200 simulations using different numbers of genetic variants. We expect the log-likelihood generated using our proposed method should be a concave function with a maximum at one of the values within the grid $\{1, 2, 3, 4, 5\}$. As shown in Figure 3.1, the majority of the log-likelihoods are concave-shaped curves, but a few show non-concave trends and some extreme values, presumably due to Monte Carlo error. In general, as the number of genetic variants increases, we can see the log-likelihoods have more curvature and are more concentrated at $m = 2$, which is the true value of m.

Table 3.1 shows the frequency of the value of m estimated over 200 simulations using different number of genetic variants. Under the true value of $m = 2$, the model with 10 genetic variants ($K = 10$) has only 38% rate of accuracy. As K increases from 10 to 50, our method is able to estimate the value of m more accurately; specifically, the corresponding rate of accuracy increases from 38% to 59.5%. Following this increasing trend, we would expect that our method can provide a reliable estimation for m with enough number of

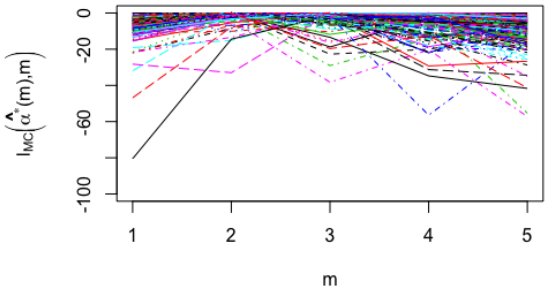
genetic variants in the model.



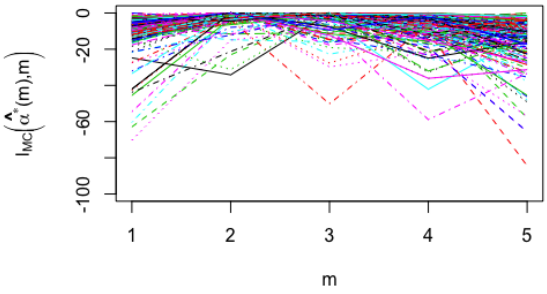
(a) $K = 10$



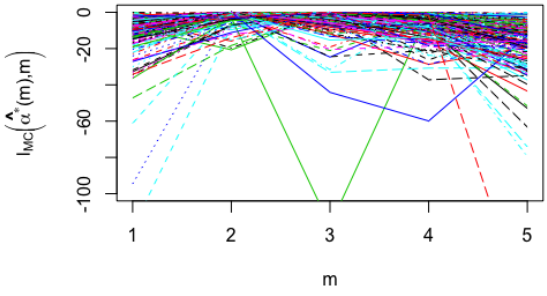
(b) $K = 20$



(c) $K = 30$



(d) $K = 40$



(e) $K = 50$

Figure 3.1: The estimated profile log-likelihood $l_{MC}(\hat{\boldsymbol{\alpha}}^*(m), m)$ constructed over 200 simulations with true $m = 2$. The curve-wise maxima, $l_{MC}(\hat{\boldsymbol{\alpha}}^*(\hat{m}), \hat{m})$, have been subtracted from each curve so that they all have maximum value zero.

True value of m	**Number of genetic variants**	**Estimated value of m**					**Rate of accuracy**
		1	2 (true)	3	4	5	
m=2	K=10	22	**76**	50	32	20	38%
	K=20	12	**91**	62	24	11	45.5%
	K=30	6	**104**	70	18	2	52%
	K=40	3	**110**	75	10	2	55%
	K=50	4	**119**	66	10	1	59.5%

Table 3.1: Summary of the simulation results under $K = 10, 20, 30, 40, 50$ with true $m = 2$.

Based on our simulation results, a sufficient number of genetic variants will give a reasonable estimate of m. However, as the value of K increases, so does the computational cost associated with running this method. For instance, it takes approximately four to five times longer to get the log-likelihood function using 50 genetic variants than using 10 genetic variants. To be more specific, with the same balanced case-control data of sample size ($n = 200$) and the same number of Monte Carlo replicates ($N = 1000$), it takes about 1 and 5 hours to implement our method with 10 and 50 genetic variants, respectively. In addition, a large number of K requires large amount of computing memory.

Chapter 4

Real Data Application

In this chapter, we discuss the application of our proposed method to the genetic data, including detailed description of our dataset and the estimation results.

4.1 Data Description

Alzheimer's Disease (AD) is the most common cause of dementia (a general form of memory loss) and loss of cognitive abilities. AD is a progressive disease, in which symptoms worsen over time. Currently, AD is one of the biggest health challenges in the world as there is neither effective drugs nor medications to treat the disease. Multiple studies indicate that genetic factors play an important role in the development of AD. Therefore, it is of interest to investigate the relationship between AD and genetic variants.

The Alzheimer's Disease Neuroimaging Initiative (ADNI) is a longitudinal multicenter study designed to identify significant genetic variants for the early detection and tracking of AD [1]. We illustrate our methodology by applying it to a dataset obtained from the first phase of ADNI study (abbreviated ADNI-1), which is a five-year study with an initial participant pool of 800 subjects recruited across North America. Of these subjects, 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.

In this study we are interested in identifying SNPs that are associated with AD. For this purpose we use only the CN and AD subjects from ADNI-1. Of the 200 CN and 200 AD subjects, we use a subset of 179 CN and 144 AD subjects from a dataset prepared for Greenlaw et al. [9]. The genetic data in this dataset are SNPs in the top 40 candidate genes for AD listed on the AlzGene database as of June 10, 2010 [9]. After quality control, imputation and filtering out SNPs with uncertain genotypes after imputation, 490 SNPs from 33 genes were

available for analysis (see Table 4.1). Each SNP is coded as 0, 1 or 2 copies of the minor allele.

Recall that our method for selecting the shrinkage parameter combines information across independent SNPs. To get a better idea of the correlation between SNPs in our dataset, we generated LDheatmaps [22] by gene. LDheatmaps display measures of linkage disequilibrium (LD) between pairs of SNPs as coloured pixels on an image. Figure 4.1 shows one such heatmap using the R^2 measure of LD for the 69 SNPs in NEDD9. Though there is some correlation among nearby SNPs, there is generally low correlation across this gene. LDheatmaps for other genes show similar patterns, with relatively low correlation between SNPs except for those close to each other.

	Gene	Chromosome	Basepair Start	Basepair End	Number of SNPs
1	CHRNB2	1	152817656	152817656	1
2	CR1	1	205737551	205881074	15
3	ECE1	1	21417188	21546592	37
4	MTHFR	1	11768839	11789631	10
5	TF	1	94776672	94777808	3
6	BIN1	2	127534016	127575911	12
7	IL1A	2	113253694	113258278	2
8	IL1B	2	113306861	113306861	1
9	NEDD9	6	113306861	11491962	69
10	PGBD1	6	28358215	28377642	6
11	TNF	6	31652168	31652168	1
12	CLU	8	27520436	27530121	2
13	DAPK1	9	89300645	89511843	82
14	IL33	9	6203387	6241507	14
15	CALHM1	10	105203755	105208349	3
16	CH25H	10	90958083	90958083	1
17	ENTPD7	10	101417040	101456596	4
18	SORCS1	10	108334549	108911743	94
19	TFAM	10	59818698	59829368	6
20	GAB2	11	77608440	77801479	18
21	PICALM	11	85349350	85458970	23
22	SORL1	11	120826964	121006965	33
23	ADAM10	15	56690061	56829304	19
24	ACE	17	58911961	58927493	7
25	GRN	17	39779191	39779191	1
26	THRA	17	35470048	35501880	3
27	TNK1	17	7223868	7233450	3
28	APOE	19	50100676	50100676	1
29	EXOC3L2	19	50417946	50421115	2
30	GAPDHS	19	40715645	40723238	3
31	LDLR	19	11063306	11103658	9
32	CST3	20	23559359	23561750	1
33	PRNP	20	4613262	4630507	4
	Total				490

Table 4.1: Summary table of SNPs in ADNI-1 study. Adapted from Table 2.2 of [20].

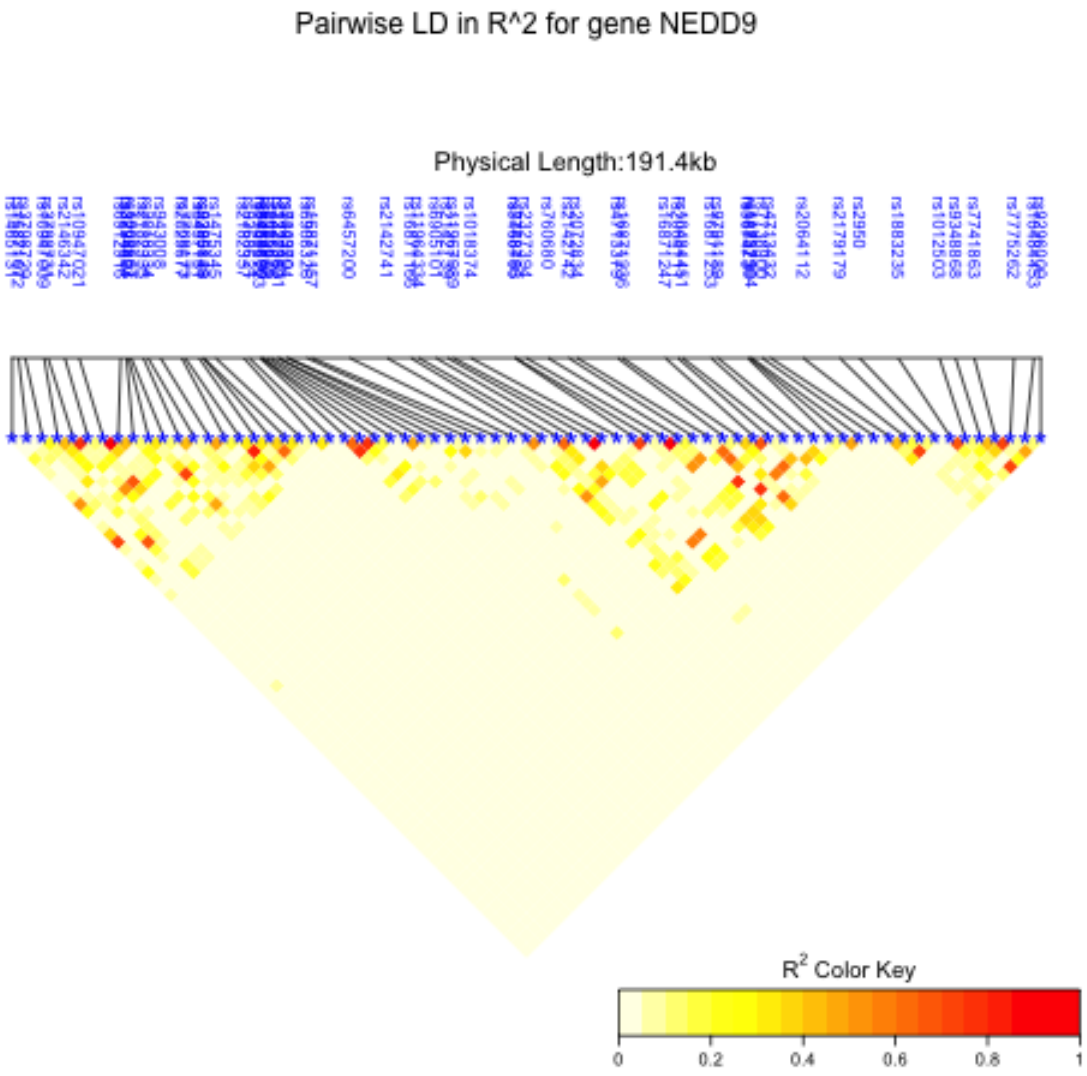


Figure 4.1: Heat map of pairwise R^2 LD measures for groups of nearby SNPs, with 69 SNPs in gene NEDD9. The LD heat map shows typical triangle-shaped blocks, darker colours indicating higher pairwise correlation.

4.2 Determining m

To estimate the regression coefficients of the 490 SNPs, we first need to estimate the shrinkage parameter m of the penalized logistic regression. Recall that the marginal likelihood introduced in Section 2.2 for selecting m is based on a set of independent SNP.

Among the 490 SNPs, we randomly select 35 SNPs with MAF less than 0.1. The rationale behind selecting low-frequency SNPs is that they are most prone to spare data bias, and are therefore most useful for selecting the amount of shrinkage. According to our simulation results, we make a compromise between the computational costs and the estimation accuracy, thus choosing the number of SNPs to be 35 for approximately good estimation but without heavy computations. The LDheatmap in Figure 4.2 displays the within-gene pairwise R^2 of these 35 SNPs. From Figure 4.2, except for those SNPs very close to each other, there is generally no correlation across these 35 SNPs. The pairwise R^2 measures suggests that assuming the SNPs are independent is reasonable in this case.

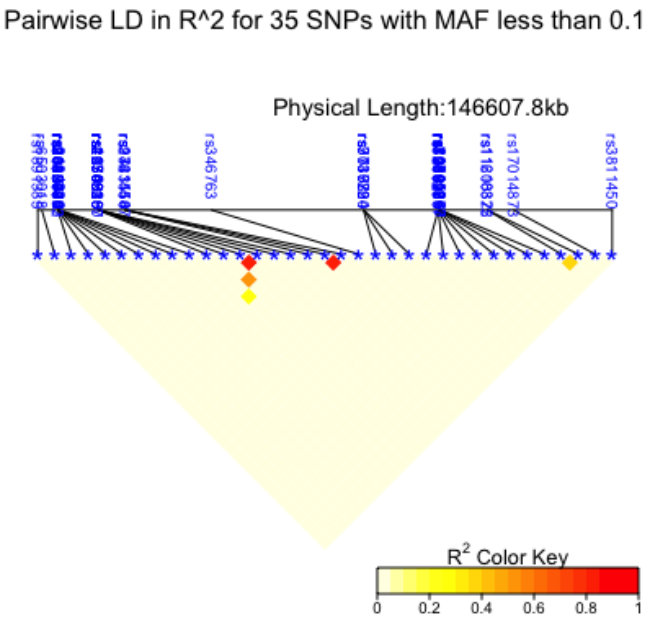


Figure 4.2: Heat map of pairwise LD measures between 35 selected SNPs with MAF less than 0.1. All pairwise R^2 are lower than 0.4.

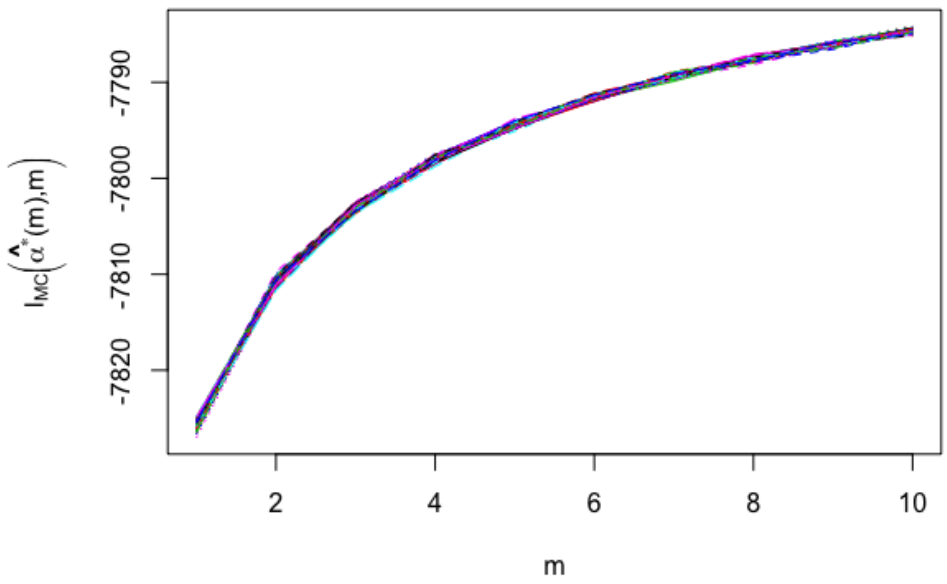


Figure 4.3: The estimated profile log-likelihood of m constructed using the ADNI-1 case-control data. We repeat the procedure 100 times.

Figure 4.3 displays 100 Monte Carlo estimates of the profile log-likelihood to select m over the grid of values $m = 1, 2, ..., 10$. The likelihood function is monotonically increasing with the value of m. And the likelihood keeps increasing for $m \geq 10$. Future work is required to understand these results. Without an guidance on an appropriate value of m for these data, we tried analysis of single-SNP association for $m = 1, 5, \text{or}, 10$. The results are shown in Appendix B.

Chapter 5

Discussion

This project proposes a method to select the shrinkage parameter m in penalized logistic regression analysis for case-control studies with log-$F(m, m)$ prior. Information about m accrues with the number, K, of independent genetic markers used for estimation. The value of m is the maximizer of a marginal likelihood obtained by integrating the random effects out of the joint distribution of the genetic data and random effects. Our maximization algorithm is a hybrid of an EM algorithm and brute-force maximization of Monte Carlo estimates of the marginal likelihood.

Our simulation results show a trade-off between computational cost and estimation accuracy as we increase the different number of genetic variants used to estimate m. The techniques used in our estimation procedure, such as the EM algorithm and the Monte Carlo integration, are known to be time-consuming, so our simulation studies are limited with results only from $K = 10$ to $K = 50$. Extrapolating from our simulation results (see Table 3.1 in Section 3.2), we expect our proposed method would achieve 80% accuracy when K is large enough (i.e. $K = 80$) but the computational cost incurred by increasing the number of genetic variants is prohibitive. Thus, improving the computational efficiency of this estimation procedure is a potential direction for future work.

Recall that our simulation studies generate the covariates from Gaussian distributions. This does not mimic the sparse covariates that motivated development of our methods. Therefore, we propose to conduct further simulations where each covariate counts the number of copies (0, 1 or 2) of the minor allele of a simulated rare genetic variant. Details of the simulation methods are still being determined.

A shortcoming of this method is that we are not able to find standard errors of the estimator of m as we restrict m to be a discrete integer. A possible solution is to perform bootstrap sampling to evaluate the estimator uncertainty. To assess uncertainty we can get the distribution of the estimator of m from multiple bootstrap samples, where sampling is stratified

by case-control status. However, the procedure is already computationally intensive, and bootstrapping would make this worse.

Another downside of using this method is the existence of underflow during the computing process. Underflow occurs when a mathematical operation resulting in a number which is smaller than the computer can actually represent. When underflow occurs, the EM algorithm is killed by round-off error. Recall that, for each k, we intend to maximize $\frac{1}{N}\sum_{j=1}^{N}\log[f(\boldsymbol{X}_{.k}|\alpha_k^*,\beta_{kj})]f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)},\beta_{kj})$ (2.15) during the MCEM algorithm. When the β_{kj} simulated from the log-$F(m,m)$ distribution is incompatible when the generated data, the case-control profile likelihood $f(\boldsymbol{X}_{.k}|\alpha_k^{*(p)},\beta_{kj})$ is extremely small and is rounded to 0. The problem worsens as the number of genetic variants increases. Future work can consider using log scale to prevent numerical underflow.

Bibliography

[1] ADNI procedures manual. Retrieved from `https://adni.loni.usc.edu/wp-content/uploads/2010/09/ADNI_GeneralProceduresManual.pdf` in (February, 2019), (March, 2006). University of California, San Diego.

[2] Norman E Breslow, Nicholas E Day, W Davis, et al. *Statistical methods in cancer research: volume 1-the analysis of case-control studies*, volume 32. IARC, 1980.

[3] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[4] Ludwig Fahrmeir and Gerhard Tutz. *Multivariate statistical modelling based on generalized linear models*. Springer Science & Business Media, 2013.

[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 Norman Breslow, Oernulf Borgan, Nilanjan Chatterjee, Mitchell H. Gail, Alastair Scott, and Christopher John Wild, editors, *Handbook of Statistical Methods for Case-Control Studies*, Chapman & Hall/CRC Handbooks of Modern Statistical Methods, chapter 9, pages 134–162. Chapman and Hall/CRC Press, Boca Raton, Florida, 2018.

[7] Sander Greenland. Prior data for non-normal priors. *Statistics in medicine*, 26(19):3578–3590, 2007.

[8] 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.

[9] 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.

[10] Mary N Haan and Elizabeth R Mayeda. Apolipoprotein e genotype and cardiovascular diseases in the elderly. *Current cardiovascular risk reports*, 4(5):361–368, 2010.

[11] Georg Heinze and Michael Schemper. A solution to the problem of separation in logistic regression. *Statistics in medicine*, 21(16):2409–2419, 2002.

[12] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[13] Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007):453–461, 1946.

[14] M. C. Jones. Families of distributions arising from distributions of order statistics. *Test*, 13(1):1–43, Jun 2004.

[15] Kenneth Lange. *Numerical analysis for statisticians*. Springer Science & Business Media, 2010.

[16] Richard A Levine and George Casella. Implementations of the monte carlo em algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001.

[17] Chia-Chen Liu, Takahisa Kanekiyo, Huaxi Xu, and Guojun Bu. Apolipoprotein e and alzheimer disease: risk, mechanisms and therapy. *Nature Reviews Neurology*, 9(2):106, 2013.

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

[19] 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.

[20] Haoyao Ruan. Covariance-adjusted, sparse, reduced-rank regression with application to imaging-genetics data. Master's thesis, Simon Fraser University, 2019.

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

[22] Ji-Hyung Shin, Sigal Blay, Brad McNeney, Jinko Graham, et al. Ldheatmap: an r function for graphical display of pairwise linkage disequilibria between single nucleotide polymorphisms. *Journal of Statistical Software*, 16(3):1–10, 2006.

[23] Greg CG Wei and Martin A Tanner. A monte carlo implementation of the em algorithm and the poor man's data augmentation algorithms. *Journal of the American statistical Association*, 85(411):699–704, 1990.

[24] Jiying Wen. Penalized logistic regression in case-control studies. Master's thesis, Simon Fraser University, 2016.

[25] CF Jeff Wu et al. On the convergence properties of the em algorithm. *The Annals of statistics*, 11(1):95–103, 1983.

[26] 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

Derivation of the profile likelihood function for case-control data due to Hosmer et.al

The standard logistic regression model assumes that the binary outcome variable $Y_i \sim$ Bernoulli(π_i) for each subject i, where π_i is the probability of Y_i takes on value 1 given subject's covariate data X_i.

$$\pi_i = P(Y_i = 1|X_i) = \frac{\exp(\alpha + X_i\beta)}{1 + \exp(\alpha + X_i\beta)}. \tag{A.1}$$

In a case-control study, the likelihood function is based on the subjects that are selected, so we define a variable S to indicate whether the subject has been selected from the population:

$$\begin{cases} S_i = 1, & \text{denote selection of subject } i; \\ S_i = 0, & \text{denote non-selection of subject } i. \end{cases}$$

Then $p_1 = P(S_i = 1|Y_i = 1)$ is the probability that the subject i is sampled as a case and similarly, $p_0 = P(S_i = 1|Y_i = 0)$ is the probability that the subject i is sampled as a control. And let $P(Y_i = 1|S_i = 1)$ and $P(Y_i = 0|S_i = 1)$ denote the proportion of selected subjects belonging to cases and controls respectively, we have

$$P(Y_i = 1|S_i = 1) = \frac{n_1}{n_0 + n_1} \text{ and } P(Y_i = 0|S_i = 1) = \frac{n_0}{n_0 + n_1}. \tag{A.2}$$

With n_1 cases and n_0 controls, we can rewrite the expression of p_1 and p_0 by Bayes' rule as

$$p_1 = P(S_i = 1|Y_i = 1) = \frac{P(Y_i = 1|S_i = 1)P(S_i = 1)}{P(Y_i = 1)} = \frac{n_1}{n_0 + n_1}\frac{P(S_i = 1)}{P(Y_i = 1)} \tag{A.3}$$

$$p_0 = P(S_i = 1|Y_i = 0) = \frac{P(Y_i = 0|S_i = 1)P(S_i = 1)}{P(Y_i = 0)} = \frac{n_0}{n_0 + n_1}\frac{P(S_i = 1)}{P(Y_i = 0)} \tag{A.4}$$

The joint probability of observing the case-control data under the prospective sampling design is

$$\prod_{i=1}^{n_0} P(Y_i = 1|X_i, S_i = 1) \prod_{i=n_0+1}^{n_0+n_1} P(Y_i = 0|X_i, S_i = 1). \tag{A.5}$$

For an individual term in the likelihood function shown in (2.6), by conditional probability and assuming that the selection of cases and controls is independent of the covariates, we get

$$\begin{aligned} P(Y_i = 1|X_i, S_i = 1) &= \frac{P(S_i = 1|X_i, Y_i = 1)P(Y_i = 1|X_i)}{P(S_i = 1|X_i, Y_i = 0)P(Y_i = 0|X_i) + P(S_i = 1|X_i, Y_i = 1)P(Y_i = 1|X_i)} \\ &= \frac{P(S_i = 1|Y_i = 1)P(Y_i = 1|X_i)}{P(S_i = 1|Y_i = 0)P(Y_i = 0|X_i) + P(S_i = 1|Y_i = 1)P(Y_i = 1|X_i)} \\ &= \frac{p_1 \cdot P(Y_i = 1|X_i)}{p_0 \cdot P(Y_i = 0|X_i) + p_1 \cdot P(Y_i = 1|X_i)}. \end{aligned} \tag{A.6}$$

Substituting p_0 and p_1 and using (2.2), we can further expand (2.7) as

$$\begin{aligned} P(Y_i = 1|X_i, S_i = 1) &= \frac{p_1 \cdot P(Y_i = 1|X_i)}{p_0 \cdot P(Y_i = 0|X_i) + p_1 \cdot P(Y_i = 1|X_i)} \\ &= \frac{p_1 \cdot \frac{\exp(\alpha + X_i\beta)}{1+\exp(\alpha+X_i\beta)}}{p_0 \cdot \frac{1}{1+\exp(\alpha+X_i\beta)} + p_1 \cdot \frac{\exp(\alpha+X_i\beta)}{1+\exp(\alpha+X_i\beta)}} \\ &= \frac{p_1 \cdot \exp(\alpha + X_i\beta)}{p_0 + p_1 \cdot \exp(\alpha + X_i\beta)} \\ &= \frac{\frac{p_1}{p_0} \cdot \exp(\alpha +_i \beta)}{1 + \frac{p_1}{p_0} \cdot \exp(\alpha + X_i\beta)} \\ &= \frac{\exp\left(\alpha + \log\left(\frac{p_1}{p_0}\right) + X_i\beta\right)}{1 + \exp\left(\alpha + \log\left(\frac{p_1}{p_0}\right) + X_i\beta\right)} \\ &= \frac{\exp(\alpha^* + X_i\beta)}{1 + \exp(\alpha^* + X_i\beta)}. \end{aligned} \tag{A.7}$$

Here $\alpha^* = \alpha + \log\left(\frac{p_1}{p_0}\right)$, and by substituting p_1 and p_0 defined in (2.4) and (2.5), we obtain

$$\alpha^* = \alpha + \log\left(\frac{n_1}{n_0}\right) - \log\left(\frac{P(Y = 1)}{P(Y = 0)}\right), \tag{A.8}$$

where α is the intercept term in the logistic regression model corresponding to a prospective study, and $P(Y = 1)$ and $P(Y = 0)$ are the population probabilities of having and not having the disease, respectively [21]. By a similar argument, we have

$$P(Y_i = 0|X_i, S_i = 1) = \frac{1}{1 + \exp(\alpha^* + X_i\boldsymbol{\beta})}. \tag{A.9}$$

Next, inserting $P(Y_i = 1|X_i, S_i = 1)$ (2.8) and $P(Y_i = 0|X_i, S_i = 1)$ (2.10) into (2.6) produces

$$\begin{aligned} L(\alpha^*, \beta) &= f(\boldsymbol{X}|\alpha^*, \beta) \\ &= \prod_{i=1}^{n_0} \frac{1}{1 + \exp(\alpha^* + X_i\beta)} \prod_{i=n_0+1}^{n_0+n_1} \frac{\exp(\alpha^* + X_i\beta)}{1 + \exp(\alpha^* + X_i\beta)} \\ &= \prod_{i=1}^{n} \frac{\exp(Y_i(\alpha^* + X_i\beta))}{1 + \exp(\alpha^* + X_i\beta)} \end{aligned} \tag{A.10}$$

Appendix B

Log-$F(m, m)$ penalized-likelihood method applied to ADNI-1 dataset

B.1 Implementing log-F(m,m) by Data Augmentation

Penalization by a log-$F(m, m)$ prior can be achieved by standard GLM through data augmentation suggested by Greenland and Mansournia [8]. According to [8], a case-control study of the relation of a single covariate to the response can be analyzed as a single binomial observation of the number of successes y in n trials using logistic regression, each having success probability $\pi = \frac{\exp(\beta x)}{(1+\exp(\beta x))}$. After reparametrization, the penalized log-likelihood using log-$F(m, m)$ prior is

$$l^*(\beta) = (y + \frac{m}{2})\beta x - (n + m)\log(1 + \exp(\beta x)) + C. \tag{B.1}$$

Thus, the penalized log-likelihood $l^*(\beta)$ is equivalent to the unpenalized log-likelihood obtained by adding $\frac{m}{2}$ to Y and m to n; in other words, we add m observations to the dataset, in which we have $\frac{m}{2}$ successes and $\frac{m}{2}$ failures. In the multiple-covariate logistic model, the prior distribution for each regression parameter is a log-$F(m, m)$ density, except the intercept. The penalty term is the product of independent log-$F(m, m)$ priors [8].

In our analysis, we include one SNP each time into our logistic regression model, adjusting for other covariates (see Section B.2 for details). As shown in Figure B.1, suppose the original dataset contains n observations, one covariate encoding the minor allele count (0, 1 or 2) of the SNP and p other covariates for adjustment. We only penalize the coefficient associated with the SNP, so the augmented dataset can be constructed as follows: we add one pseudo-observation with $\frac{m}{2}$ success and $\frac{m}{2}$ failures to the response (even if m is an odd number) and a single row to the design matrix consisting all zeros except for a one indicating the index of the SNP. Analyzing the augmented dataset with standard logistic regression will give us the penalized estimates and the corresponding standard errors.

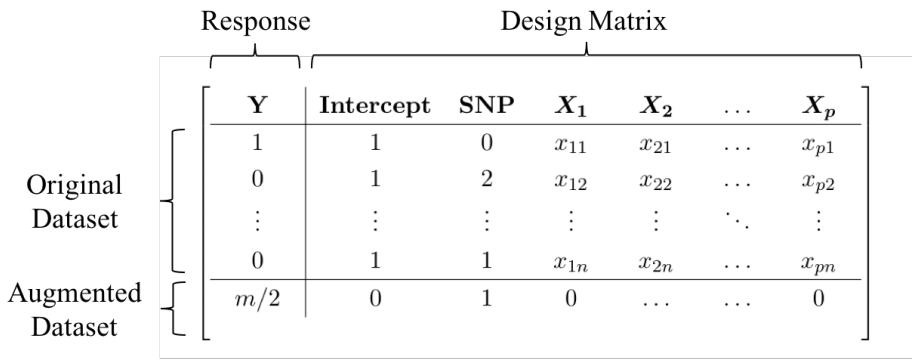


Figure B.1: Illustration of data augmentation in implementation of log-$F(m, m)$ penalization.

B.2 Estimation Results

The association between SNPs and AD phenotype is estimated by the penalized logistic regression with log-$F(m, m)$ prior, adjusting for age, gender, Apolipoprotein E (APOE) genotype and the first 10 multidimensional scaling components (MDS). The APOE genotype is a genetic risk factor for AD, which is located in chromosome 19 and contains three different alleles: e2, e3, e4 [10]. Individuals carrying the e4 allele have a higher risk of developing AD than those carrying the more common e3 allele, whereas the e2 allele decreases the risk [17]. The first 10 MDS components are a set of low dimensional principal components used to correct for ancestry and population stratification. The logistic regression model can be expressed as:

$$\text{Y} = \text{a single SNP} + \text{Age} + \text{gender (0/1)} + \text{APOE genotypes (6 levels)} + 10 \text{ MDS}.$$

According to the estimation results in Section 4.2, the MLE of m dose not exist since the likelihood is monotone without a maximum. Thus, we choose different values of m to illustrate the estimation results using penalized logistic regression with log-$F(m, m)$ prior. Table B.1, B.2 and B.3 show the information for the significant SNPs selected under $m = 1, 5, 10$, respectively. From left to right, the columns show: SNP ID, the estimated coefficient, the p-value, the gene where the SNP belongs to and the corresponding chromosome. As m increases, few SNPs are selected and the amount of shrinkage applied to each SNP increases such that the estimated coefficients of SNPs are pulled towards zero. It is interesting that same set of significant SNPs will be selected when m is greater than 8, illustrating the convergence of the shrinkage introduced by log-$F(m, m)$ prior as m increases.

In our results, multiples significant SNPs are from gene *MTHFR*, *ECE1*, *CR1*, *DAPK1*, *SORCS1*, *SORL1*. Ruan [20] prioritizes the genes of interest by looking at the importance probability in bootstrapping [20], and the above genes are on the top-ten list. As [20] suggests, more SNPs tends to be selected from genes with a large set of SNPs, such as *DAPR1* (82 SNPs) and *SORCS1* (94 SNPs).

Based on our findings, SNPs *rs2184226*, *rs6541003*, *rs3818361*, *rs6701713*, *rs1408077* and *rs689021* are the most significant SNPs with P-value ≤ 0.01. By looking at the estimated coefficients, most selected SNPs have positive coefficients, implying that individuals carrying the minor allele of the SNP are at increased risk of AD, whereas the SNPs with negative coefficients decreases the risk.

	RSID	**Coefficient**	**P-value**	**Gene**	**Chromosome**
1	rs4846048	0.52357795	0.00806762	MTHFR	1
2	rs2184226	1.06773637	0.00244728	MTHFR	1
3	rs6541003	0.50622223	0.00613758	MTHFR	1
4	rs1572151	0.80782317	0.01720435	MTHFR	1
5	rs3026913	0.60317381	0.02609333	ECE1	1
6	rs212524	-0.38371404	0.04505582	ECE1	1
7	rs212525	-0.38106075	0.04502948	ECE1	1
8	rs3818361	0.72738411	0.00190195	CR1	1
9	rs6701713	0.72738411	0.00190195	CR1	1
10	rs1408077	0.79157118	0.00114537	CR1	1
11	rs3783526	-0.41805488	0.04330159	IL1A	2
12	rs2058882	0.55028673	0.01863757	DAPK1	9
13	rs10125534	0.49882579	0.03617202	DAPK1	9
14	rs10868609	0.49882579	0.03617202	DAPK1	9
15	rs1316489	0.49882579	0.03617202	DAPK1	9
16	rs10786978	-0.42176809	0.03244574	SORCS1	10
17	rs2418828	-0.37195628	0.04290330	SORCS1	10
18	rs7920985	-0.36126389	0.04669385	SORCS1	10
19	rs661057	-0.45656244	0.01449630	SORL1	11
20	rs4631890	0.47892605	0.00947936	SORL1	11
21	rs676759	-0.43170093	0.02460761	SORL1	11
22	rs689021	-0.52423234	0.00615750	SORL1	11
23	rs666004	0.39793949	0.02570043	SORL1	11
24	rs6511720	-0.62217964	0.03295079	LDLR	19
25	rs6084833	0.78812978	0.04977574	PRNP	20

Table B.1: Information for the significant SNPs (P-value ≤ 0.05) estimated using log-$F(m, m)$ penalization under $m = 1$.

	RSID	Coefficient	P-value	Gene	Chromosome
1	rs4846048	0.50432104	0.00925193	MTHFR	1
2	rs2184226	0.95810640	0.00404732	MTHFR	1
3	rs6541003	0.48985961	0.00696719	MTHFR	1
4	rs1572151	0.72819311	0.02331858	MTHFR	1
5	rs3026913	0.56294109	0.03155624	ECE1	1
6	rs212524	-0.37031179	0.04874264	ECE1	1
7	rs212525	-0.36792933	0.04866193	ECE1	1
8	rs3818361	0.69103705	0.00243329	CR1	1
9	rs6701713	0.69103705	0.00243329	CR1	1
10	rs1408077	0.74933910	0.00152013	CR1	1
11	rs3783526	-0.40114245	0.04750612	IL1A	2
12	rs2058882	0.52238996	0.02181359	DAPK1	9
13	rs10125534	0.47256027	0.04135426	DAPK1	9
14	rs10868609	0.47256027	0.04135426	DAPK1	9
15	rs1316489	0.47256027	0.04135426	DAPK1	9
16	rs10786978	-0.40621541	0.03556522	SORCS1	10
17	rs2418828	-0.35995098	0.04623175	SORCS1	10
18	rs661057	-0.44143822	0.01609432	SORL1	11
19	rs4631890	0.46343917	0.01060392	SORL1	11
20	rs676759	-0.41657473	0.02709044	SORL1	11
21	rs689021	-0.50612613	0.00701493	SORL1	11
22	rs666004	0.38582729	0.02791029	SORL1	11
23	rs6511720	-0.57481374	0.03976784	LDLR	19

Table B.2: Information for the significant SNPs (P-value ≤ 0.05) estimated using log-$F(m, m)$ penalization under $m = 5$.

	RSID	Coefficient	P-value	Gene	Chromosome
1	rs4846048	0.48216252	0.01084784	MTHFR	1
2	rs2184226	0.84797337	0.00671625	MTHFR	1
3	rs6541003	0.47084959	0.00808443	MTHFR	1
4	rs1572151	0.64809336	0.03181508	MTHFR	1
5	rs3026913	0.51949391	0.03880981	ECE1	1
6	rs3818361	0.65038300	0.00321559	CR1	1
7	rs6701713	0.65038300	0.00321559	CR1	1
8	rs1408077	0.70245443	0.00209005	CR1	1
9	rs2058882	0.49122152	0.02604729	DAPK1	9
10	rs10125534	0.44334407	0.04806279	DAPK1	9
11	rs10868609	0.44334407	0.04806279	DAPK1	9
12	rs1316489	0.44334407	0.04806279	DAPK1	9
13	rs10786978	-0.38834556	0.03957640	SORCS1	10
14	rs661057	-0.42390914	0.01819271	SORL1	11
15	rs4631890	0.44546311	0.01209711	SORL1	11
16	rs676759	-0.39911561	0.03030652	SORL1	11
17	rs689021	-0.48521974	0.00817201	SORL1	11
18	rs666004	0.37170165	0.03075800	SORL1	11
19	rs6511720	-0.52494568	0.04870902	LDLR	19

Table B.3: Information for the significant SNPs (P-value $\leq$ 0.05) estimated using log-$F(m, m)$ penalization under $m = 10$.

Appendix C

Code

```
## MCEM Algorithm
MCEM=function(m,data,N) {
  # Input:
  # - m is the value of m
  # - data is the simulated case-control data obtained by simUnmatched()
  # - N is number of Monte Carlo replicates

  model=glm(case~.,data=data,family=binomial(link="logit"))
  alpha_star_initial=model$coefficients[1]

  AlphaStar=numeric()
  AlphaStar[1]=0
  AlphaStar[2]=alpha_star_initial

  p=2
  threshold=1E-04

  K_Beta=function(beta,con,case) {
    sum(-log(1+exp(AlphaStar[p]+con*beta)))+sum(AlphaStar[p]+case*beta-log(1+exp(AlphaStar[p]+case*beta)))
  }

  caseX=data %>% filter(case=="1") %>% select(-case)
  conX=data %>% filter(case=="0") %>% select(-case)
  Y=rep(data$case,times=N)
  betas=log(rf(N,m,m))

  O=numeric()
  for (i in 1:N) {
    O=c(O,data$X*betas[i])
  }

  while(abs(AlphaStar[p]-AlphaStar[p-1])>=threshold) {
    W_t=numeric()
    for (i in 1:N) {
      W_t[i]=exp(K_Beta(betas[i],conX,caseX))
    }
    W=rep(W_t,each=dim(data)[1])

    g=glm(Y~offset(O),weights=W,family=binomial(link="logit"))
    cat("EM iteration",p-1,":",g$coefficients,"\n")
    p=p+1
    AlphaStar[p]=g$coefficients
  }
  return(last(AlphaStar))
}
## ------------------------------------------------------
## Estimate the Profile Likelihood
profilelkhd=function(data,mvals,N) {
  # Input:
  # - data is the simulated case-control data obtained by simUnmatched()
  # mvals is a set of values of m
  # - N is number of Monte Carlo replicates

  ll=rep(NA,length(mvals))
  for(m in 1:length(mvals)) {
    cat("Estimating profile log-likelihood for m =",m,"\n")
    ll[m]=0
    for(k in 1:K) {
      data_k=data[,c(1,k+1)] # first column is case, then K covars
      names(data_k)=c("case","X")
      alpha_k=MCEM(m,data_k,N) # estimating alpha_k by MCEM
      ll[m]=ll[m]+lkhdk(alpha_k,data_k,m,N)
    }
```

```
  }
  return(ll)
}

lkhdk=function(alpha_k,data,m,N) {
  # Input:
  # - alpha_k is the output of MCEM
  # - data is the simulated case-control data obtained by simUnmatched()
  # - m is the value of m
  # - N is number of Monte Carlo replicates

  betas=log(rf(N,m,m))
  lvec=rep(NA,N)
  for(j in 1:N) {
    lvec[j]=prod(exp(data$case*(alpha_k+data$X*betas[j]))/(1+exp(alpha_k+data$X*betas[j])))
  }
  return(log(mean(lvec)))
}
## ------------------------------------------------------
## Log-F(m,m)-penalized Likelihood Inference by Data Augmentation
logF = function(form,dat,m,control=glm.control()) {
  # Input:
  # - form is an R formula
  # - dat is the data
  # - m is the numerator and denominator degrees of freedom for the log-F prior
  # - control is algorithm control arguments to be passed to glm().

  # Step 1: Extract (i) the response and (ii) the design matrix
  # from the input formula and data frame so that we can augment them.
  mf = model.frame(form,dat)
  D = model.response(mf)
  X = model.matrix(form,dat)

  # Step 2 (augmentation): one pseudo-observation for each covariate (in this case the SNP),
  # where the response is m/2 successes and m/2 failures (even if
  # m is an odd number) and the covariates are all zeros except for
  # a one indicating the index of the covariate.
  # Following the recommendation of Greenland and Mansournia (2015; p. 3139)
  # we do not penalize the intercept.
  n = rep(1,length(D))
  zeros = rep(0,ncol(X))
  pseudoD = m/2; pseudoN = m
  D=c(D,pseudoD); n=c(n,pseudoN)
  pseudoX = zeros; pseudoX[2]=1; X = rbind(X,pseudoX)
  # assume that the SNP is the first covariate in the formula

  # Step 3: Setup a response matrix with columns for number of successes
  # and number of failures.
  Y = cbind(D,n-D)

  # Step 4: Set up X's as a data.frame with null rownames and correct colnames.
  rownames(X) = NULL
  xvars = all.vars(form)[-1] # exclude response
  X = data.frame(X)
  names(X) = c("Int",xvars)

  # Step 5: set up a formula and call glm()
  form = formula(paste("Y~ -1 + Int + ",paste0(xvars,collapse="+")))
  out = glm(form,data=X,family=binomial(link="logit"),control=control)
  return(out)
}
```