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

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 e□ort 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 sta□ 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 o□ering 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

List of Tables

List of Figures

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 fle

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

Chapter 2

Methodology

2.1 Marginal likelihood for Shrinkage Parameter m

Finding the MLE of $_k$ is complicated by the in nite-dimensional nuisance parameter g. Qin and Zhang [17] proved that a pro le likelihood function obtained by pro ling g out has the same form as the logistic regression (2.1), with a di erent intercept term $_k$.

$$L(_k; _k) =$$

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.

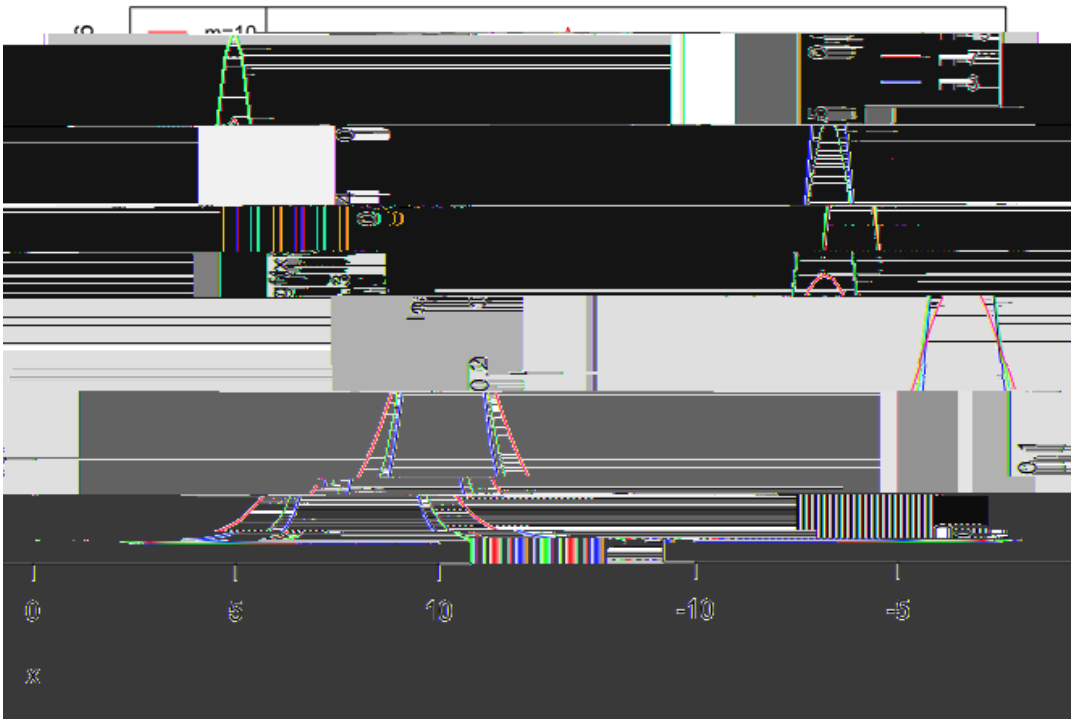


Figure 2.1: The shape of log-F distribution

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 $I(\theta) = \log P(\theta)$. If $I(\theta)$ is a smoothfunction w8th t81th

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.

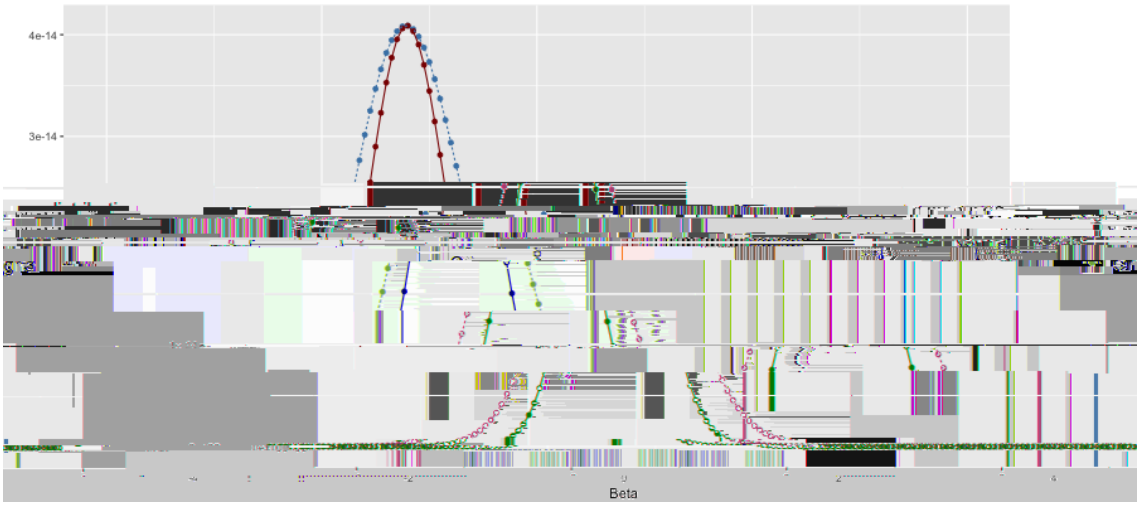


Figure 2.2: The original posterior density for β and corresponding unnormalized Normal density

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 $E_p(L(X)|)$, in which 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).

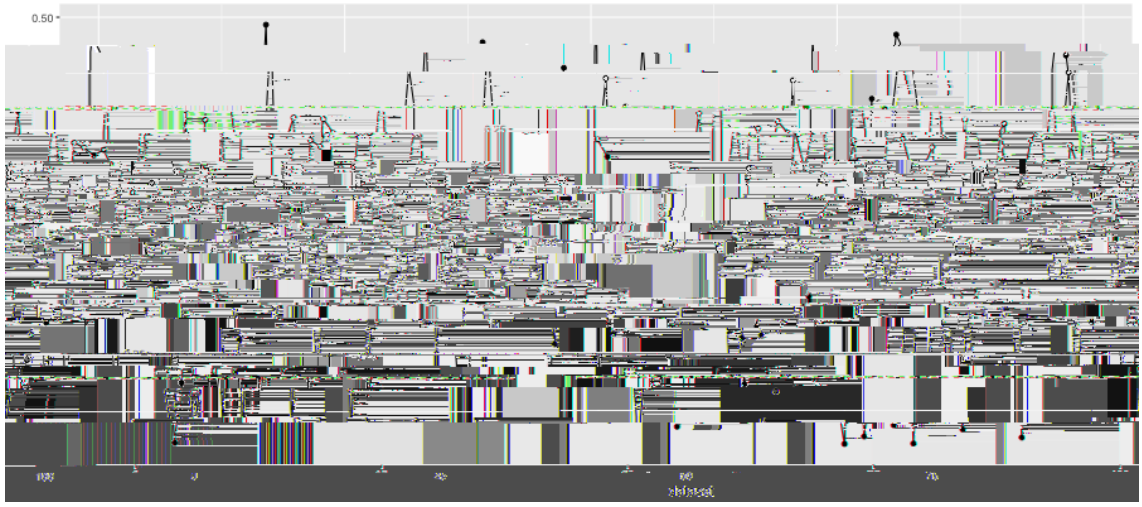


Figure 2.3: $(\hat{L}_{MC} \quad \hat{L}_{LA})/\hat{L}_{MC}$, dashed lines indicating 0.3

Finally, for a single dataset we compared the MC and LA estimates of the marginal likelihood of $\alpha = 3$ and a grid of $m = (0.5, 1, \ldots, 10)$. 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 $\hat{L}(\alpha, m)$ to estimate (α, m). Calculation of derivatives of $\hat{L}(\alpha, m)$ 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 of R^p.

2.3.1 Nelder-Mead

The Nelder-Mead extended simplex method is most easily described for the case $p = 2$. Ster.

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

expanded or contracted at different iterations [13], to change the speed at which we move through . For p-dimensional 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 func-

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 P' from P;
    if crossover condition satisfied then
        P' = crossover(P');
    end
    if mutation condition satisfied then
        P' = mutation(P');
    end
    P = E ∪ P';
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 . 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* $x = (s, v) \in P$ do
    update the velocity v;
    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 I for all particles following given topology in C;
end

2.4 Summary of Maximum Marginal Likelihood Estimator of m

The approximate marginal likelihood is $\hat{L}(\alpha, m) = \prod_{k=1}^{K} \hat{E}[L|\alpha, m]$, where $\hat{E}[L|\alpha, m]$ is the Laplace approximation to $\int L(\alpha, \beta) p(\beta|m) d\beta$. Each approximate integral $\hat{E}[L|\alpha, m]$

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 a□ect the bias and variance of our estimator of m?

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

True m	Method	Mean of $\hat{m}$	Variance of $\hat{m}$
2			

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

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

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

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]

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

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

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

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 $e^{\alpha} = \exp(\alpha_k), e^{\beta_k} = \exp(\beta_k)$, we have

$$\sum_{i=1}^{n} (Y_i X_i^k - \frac{e^{\alpha_k} e^{X_i^k} X_i^k}{1 + e^{\alpha_k} e^{X_i^k}}) - \frac{m}{2} + m \frac{1}{1 + 1/e^{\beta_k}} = \sum_{i=1}^{n} (Y_i X_i^k - X_i^k + X_i^k$$

Consider when $_k \to 1$; $e_k \to 0$, $\frac{X_i^k}{1+e\ e_k^{X_i^k}} \to X_i^k$ when $X_i^k > 0$; $\to 0$ when $X_i^k < 0$. Then

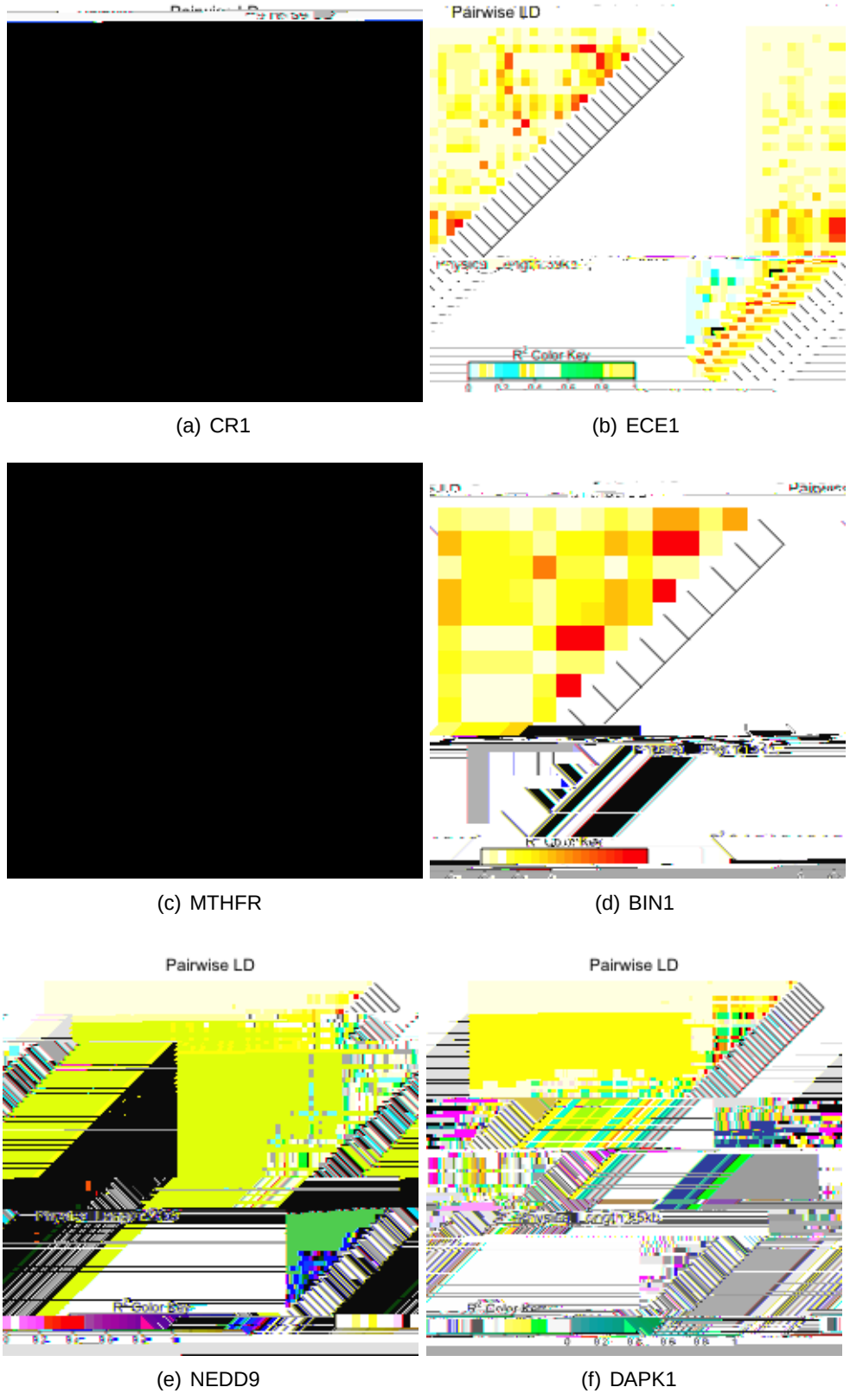
$$\sum_{i=1}^{n} (Y_i X_i^k \quad X_i^k + \frac{X_i^k}{1+e\ e_k^{X_i^k}}) \quad \frac{m}{2} + \frac{m}{1+e_k}$$
$$\to \sum_{i=1}^{n} (Y_i X_i^k \quad X_i^k) + \sum_{i=1}^{n} X_i^k I(X_i^k > 0) + \frac{m}{2} \qquad \text{(A.5)}$$
$$= \sum_{i=1}^{n} (Y_i X_i^k) \quad \sum_{i=1}^{n} X_i^k I(X_i^k < 0) + \frac{m}{2}$$
$$= \sum_{i=1}^{n} (X_i^k I(Y_i = 1) I(X_i^k > 0) \quad X_i^k (Y_i = 0) I(X_i^k < 0)) + \frac{m}{2} > 0$$

Similarly, when $_k \to 1$; $e_k \to 1$, $\frac{X_i^k}{1+e\ e_k^{X_i^k}} \to 0$ when $X_i^k > 0$; $\to X_i^k$ when $X_i^k < 0$. We have

$$\sum_{i=1}^{n} (Y_i X_i^k \quad X_i^k + \frac{X_i^k}{1+e\ e_k^{X_i^k}}) \quad \frac{m}{2} + \frac{m}{1+e_k} \qquad \text{(A.6)}$$
$$\to \sum_{i=1}^{n} (X_i^k I(Y_i = 1) I(X_i^k < 0) \quad X_i^k (Y_i = 0) I(X_i^k > 0)) \quad \frac{m}{2} < 0$$

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

$$\frac{\partial^2}{\partial_k^2}\log(L_p) = \sum_{i=1}^{n} (\quad \exp(_k + X_i^k \ _k)(X^{=0}$$ must exist according to inter- X = 0 must exist accord80 Td [(99 W 0 0 m 9.578 0 l S Q BT1]T

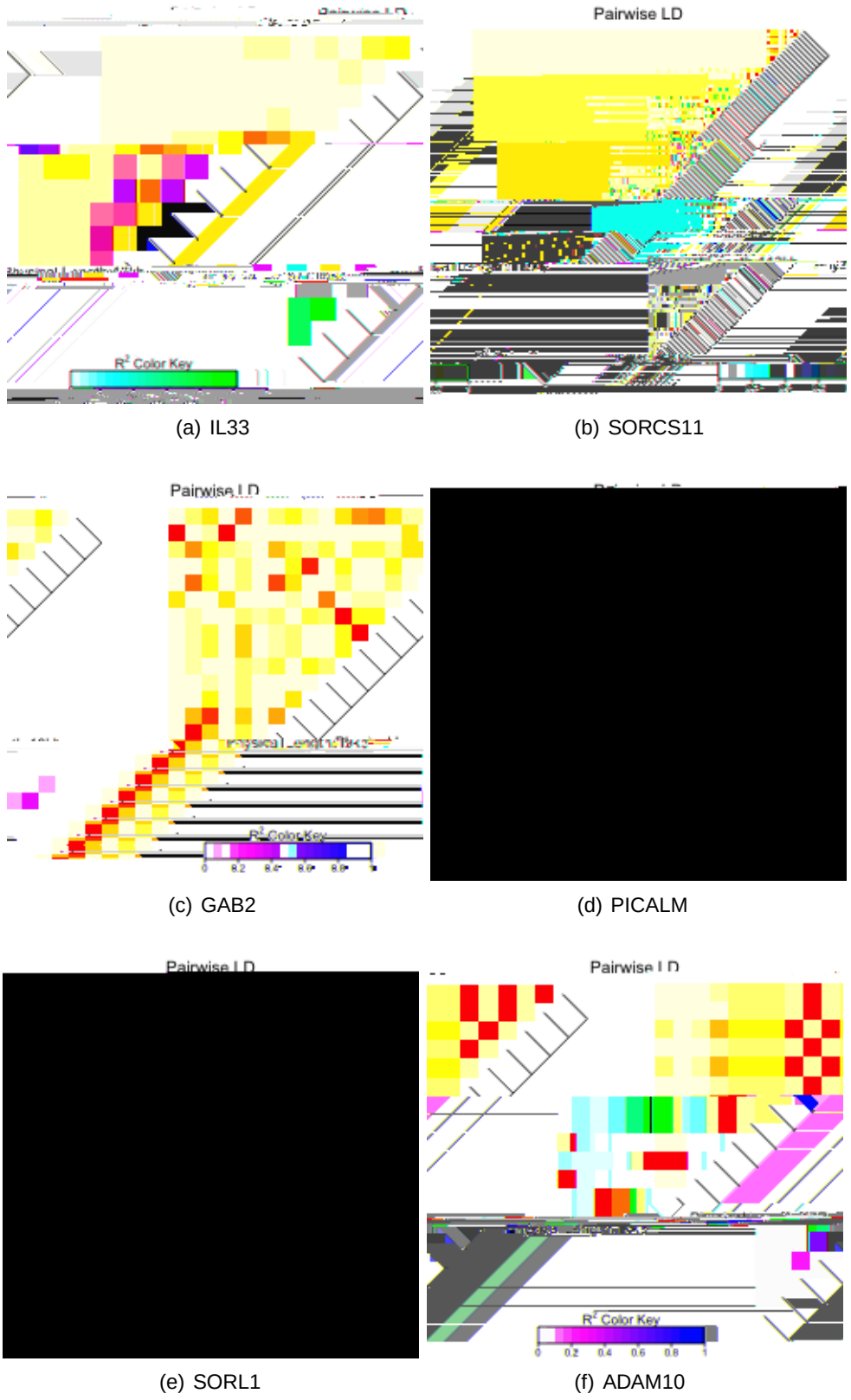


(a) CR1 (b) ECE1

(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*

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
```

```
            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
```