Spring 2020

Studies of Group Fused Lasso and Probit Model for Right-Censored Data

Tuan Quoc Do

Studies of Group Fused Lasso and Probit model for Right-Censored data

by

Tuan Quoc Do

Bachelor of Business (Economics and Finance)
RMIT University, 2015

Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy in

Statistics

College of Arts and Sciences

University of South Carolina

2020

Accepted by:

Karl Gregory, Major Professor

Lianming Wang, Major Professor

David Hitchcock, Committee Member

Jesse Kass, Committee Member

Cheryl L. Addy, Vice Provost and Dean of the Graduate School

Acknowledgments

I will be forever thankful for the Department of Statistics for giving me a chance to be a student here. I still remember how happy I was when I received the news. I want to thank all of my professors and friends at USC. Particularly I want to thank Dr. David Hitchcock and Dr. Jesse Kass for being my committee members.

But most importantly I want to say thank you to my family and my wife, Yifan, for their support. Last but definitely not least, I would not be able to complete this dissertation without the guidance and help of the two coolest advisors, Dr. Karl Gregory and Dr. Lianming Wang.

Abstract

This document is composed of three main chapters. In the first chapter, we study the mixture of experts, a powerful machine learning model in which each expert handles a different region of the covariate space. However, it is crucial to choose an appropriate number of experts to avoid overfitting or underfitting. A group fused lasso (GFL) term is added to the model with the goal of making the coefficients of the experts and the gating network closer together. An algorithm to optimize the problem is also developed using block-wise coordinate descent in the dual counterpart. Numerical results on simulated and real world datasets show that the penalized model outperforms the unpenalized one and performs on par with many well-known models.

The second chapter studies GFL on its own and methods to solve it efficiently. In GFL, the response and the coefficient of each observation are not scalars but vectors. Thus, many fast solvers of the fused lasso cannot be applied to the GFL. Two algorithms are proposed to solve the GFL, namely Alternating Minimization and Dual Path. Results from speed trial show that our algorithms are competitive compared to other existing methods.

The third chapter proposes a better alternative to the Box-Cox transformation, a popular method to transform the response variable to have an approximately normal distribution in many cases. The Box-Cox transformation is widely applied in regression, ANOVA and machine learning for both complete and censored data. However, since it is parametric, it can be too restrictive in many cases. Our proposed method is nonparametric, more flexible and can be fitted efficiently by our novel EM algorithms which accommodate both complete and right-censored data.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This dissertation consists of three projects. The first two projects discuss about the efficient calculations of the group fused lasso problem and how it can be applied in the Mixture of Experts. The last project talks about a nonparametric alternative for Box-Cox transformation which can be applied to both full data and and right-censored data.

Chapter 2

Model selection for mixture of experts using group fused lasso

2.1 Introduction

The decision tree has been one of the most popular and widely used predictive models not only in statistics but also in almost all other scientific fields. The reasons for its success are its robustness and easy interpretability. Decision trees can be fitted to almost every kind of data (Loh, 2014) and their clear structure allows everyone to interpret the model easily. However, decision trees have the huge drawback that they are greedy algorithms, meaning each split is made to optimize a splitting criterion, without consideration of latter nodes. Trees also use the hard-split rule, which is non-smooth and cannot be trained using maximum likelihood (Breiman, 2017). There have been a few modifications to the original tree to make it non-greedy (Bennett, 1994; Norouzi et al., 2015; Grubinger et al., 2011). Nonetheless, these models are relatively time-consuming to train and difficult to understand since they do not use the user-friendly likelihood function.

Jordan and Jacobs (Jordan and Jacobs, 1994) and Jacobs et al. (Jacobs et al., 1991) introduced a tree-structured architecture called Mixture of Experts (ME) which shares the same clear representation with decision trees. ME uses the same divide-and-conquer strategy like the decision tree and multivariate adaptive regression splines (Friedman, 1991), but after we divide the input space into many subspaces, we fit a linear regression (the authors call it an expert) to each subspace. The beauty

of ME is that the splits between subspaces are soft, implying that observations get assigned to all nodes or subspaces, with some probabilities which sum to 1, instead of to just one node or subspace as with decision trees. This smoothness allows the model to be estimated easily using maximum likelihood. However, ME can underfit or overfit if we choose two few or too many experts. If we choose too few linear experts, we may not have enough experts to cover complicated covariate regions. On the other hand, if there are too many experts in the model, some experts may start to specialize in noise regions, limiting generalization. Until now, the literature for model selection for ME consists mainly of three parts: growing the structure (Saito and Nakano, 1996; Fritsch et al., 1997); pruning the structure (Waterhouse and Robinson, 1995; Jacobs et al., 1997); and using Bayesian techniques (Rasmussen and Ghahramani, 2002; Ueda and Ghahramani, 2000; Kanaujia and Metaxas, 2006). The first method builds the structure slowly by adding one level or expert at a time. On the other hand, the second method begins with a large model and then tries to reduce the model complexity. The goal is to keep only the most-used branches and remove the least-used ones. The stopping time for both of the above methods is typically chosen by cross validation. Lastly, Bayesian techniques impose sparsity-promoting priors on the parameters so that the model has a smaller number of non-zero weights.

In this chapter we propose another method for pruning the structure of ME. Specifically, we want to use the group fused lasso (Bleakley and Vert, 2011) to accomplish this task. Initializing from a large structured ME, we use a fusion penalty to penalize the difference between coefficients of different components of the gating network and experts so that we will have some identical experts with identical weights as well. We then can merge these similar experts and their weights together to simplify the structure.

The chapter is organized as follows: In Section 2 we will give a review about Mixture of Experts and the penalty term that we propose. Details about how to fit

the new penalized model are included in Section 3. Numerical results on simulation and real-world datasets will be presented in Section 4.

2.2 Mixture of Experts and the Group Fused Lasso term

We first introduce and give some background on the Mixture of Experts (ME) model. Then we introduce our fused lasso penalty.

2.2.1 Mixture of Experts

In the ME model, a fixed number of experts and a gating network, which assigns weights to the experts, work together to solve a nonlinear supervised learning problem. In this chapter we will consider only the Normal regression case, but generalization to cases of classification and Poisson regression is straightforward. The gating network's job is to divide the covariate space into many small subspaces by making soft splits of the whole space. On the other hand, the job of each expert is to specialize in one of the subspaces and learn the pattern in that particular subspace. Due to soft-splitting, the model has a smooth transition from one subspace to another and the predictions in those transitioning regions are stable.

Let $\{Y, X\}$ be our sample, where Y is a vector of length N and X is a covariate matrix of dimension $N \times P$ (including the intercept column). For now we assume we work with low-dimensional data $N > P$. Modification for high-dimensional data $N < P$ will be discussed in Section 3.3.

If there are K experts in the ME model, we denote by β_k $(k = 1, ...K)$ the coefficient vector for the kth expert and by γ_k $(k = 1, ...K)$ the set of coefficients governing how the gating network assigns weights to the K experts. We also define $\beta = (\beta_1^T, \beta_2^T, \ldots, \beta_K^T)^T, \gamma = (\gamma_1^T, \gamma_2^T, \ldots, \gamma_K^T)^T$, and $\theta = (\beta^T, \gamma^T)^T$.

For a data point $(X^{(n)}, Y^{(n)})$, let $P(Y^{(n)}|k, X^{(n)}, \beta_k)$ be the conditional pdf of $Y^{(n)}$ given $X^{(n)}$ according to the k^{th} expert and let $P(k|X^{(n)}, \gamma)$ be the weight assigned to

the expert k, for $k = 1, \ldots, K$. Then the conditional pdf of $Y^{(n)}$ given $X^{(n)}$, according to the ME model is given by

$$P(Y^{(n)}|X^{(n)}, \theta) = \sum_{k=1}^{K} P(k|X^{(n)}, \gamma) P(Y^{(n)}|k, X^{(n)}, \beta_k).$$

As in the original ME paper by (Jacobs et al., 1991), we define the weights of the gating network using the softmax function such that

$$P(k|X^{(n)}, \gamma) = \frac{\exp(\gamma_k^T X^{(n)})}{\sum_{l=1}^{K} \exp(\gamma_l^T X^{(n)})}, \qquad k = 1, \ldots, K \text{ and } n = 1, \ldots, N.$$

We note that for the purpose of identifiability, we force γ_K, the coefficients of the last gating network's component, to be 0, just as in the case of multinomial logistic regression.

This choice of weight function for the gating network guarantees a positive weight for each expert across the entire covariate space (so there is soft-splitting between experts, unlike in regression trees), and the weights assigned to the experts at any point in the covariate space sum to 1. Since we consider the Normal regression case, the conditional pdf $P(Y^{(n)}|k, X^{(n)}, \beta_k)$ is the pdf of a Normal distribution with mean $X^{(n)^T}\beta_k$.

Under this setting, we can write down the likelihood function as

$$\mathbf{L}(\theta, \boldsymbol{\sigma^2}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \frac{e^{\gamma_k^T X^{(n)}}}{\sum_{l=1}^{K} e^{\gamma_l^T X^{(n)}}} \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp\left(- \frac{(Y^{(n)} - \beta_k^T X^{(n)})^2}{2\sigma^2} \right)$$

and the log-likelihood as

$$\ell(\theta, \boldsymbol{\sigma^2}) = \sum_{n=1}^{N} \ln \sum_{k=1}^{K} \frac{e^{\gamma_k^T X^{(n)}}}{\sum_{l=1}^{K} e^{\gamma_l^T X^{(n)}}} \cdot \frac{1}{\sqrt{2\pi}\sigma} \exp\left(- \frac{(Y^{(n)} - \beta_k^T X^{(n)})^2}{2\sigma^2} \right).$$

The values of θ and σ^2 which maximize the log-likelihood function cannot be found analytically, so an EM algorithm is typically used to fit the ME model (Dempster et al., 1977). For $n = 1, \ldots, N$, let $Z_1^{(n)}, \ldots, Z_K^{(n)} \in \{0, 1\}$ such that $\sum_{k=1}^{K} Z_k^{(n)} = 1$ so that only one of the $Z_1^{(n)}, \ldots, Z_K^{(n)}$ is equal to 1 while the rest are equal to 0. These

indicator variables are labels that indicate which expert in the model generated the data point.

After dropping constants, we may write the full log-likelihood as

$$\ell_{\mathbf{f}}(\theta, \sigma^{\mathbf{2}}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_k^{(n)} \ln \left[\frac{e^{\gamma_k^T X^{(n)}}}{\sum_{l=1}^{K} e^{\gamma_l^T X^{(n)}}} \cdot \frac{1}{\sigma} \exp \left(- \frac{(Y^{(n)} - \beta_k^T X^{(n)})^2}{2\sigma^2} \right) \right]. \tag{2.1}$$

Next, following (Jordan and Jacobs, 1994), as the E-step, we take the conditional expectation given Y and X of the full log-likelihood (2.1), which gives

$$\mathbf{Q}(\theta, \sigma^{\mathbf{2}}) = \sum_{n=1}^{N} \sum_{k=1}^{K} h_k^{(n)} \ln \left[\frac{e^{\gamma_k^T X^{(n)}}}{\sum_{l=1}^{K} e^{\gamma_l^T X^{(n)}}} \cdot \frac{1}{\sigma} \exp \left(- \frac{(Y^{(n)} - \beta_k^T X^{(n)})^2}{2\sigma^2} \right) \right], \tag{2.2}$$

where

$$h_k^{(n)} = \frac{\exp(\gamma_k^T X^{(n)}) \exp\Big(- (Y^{(n)} - \beta_k^T X^{(n)})^2 / 2\sigma^2 \Big)}{\sum_{l=1}^{K} \exp(\gamma_l^T X^{(n)}) \exp\Big(- (Y^{(n)} - \beta_l^T X^{(n)})^2 / 2\sigma^2 \Big)}.$$

To perform the M-step we maximize $\mathbf{Q}(\theta, \sigma^{\mathbf{2}})$ with respect to (θ, σ^2). We observe that (2.2) can be decomposed as

$$\mathbf{Q}(\theta, \sigma^{\mathbf{2}}) = \sum_{n=1}^{N} \sum_{k=1}^{K} h_k^{(n)} \ln \left[\frac{e^{\gamma_k^T X^{(n)}}}{\sum_{l=1}^{K} e^{\gamma_l^T X^{(n)}}} \right] - N \ln(\sigma) - \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{h_k^{(n)} (Y^{(n)} - \beta_k^T X^{(n)})^2}{2\sigma^2}, \tag{2.3}$$

where the first term involves only gating network parameters and the remaining two terms involve only expert parameters. We call the first term the gating network part and the remaining two terms the experts part. In the unpenalized version of ME, we can maximize each part separately using an iteratively reweighted least-squares algorithm.

2.2.2 The group fused lasso penalty term

First we define two norms that will be used later. Given an arbitrary vector $b = (b_1^T, ..., b_m^T)^T$ where each block $b_i, i = 1, 2, ...m$, has length $2P$, let

$$||b||_{2,1} = \sum_{j=1}^{m} ||b_j||_2 \quad \text{and} \quad ||b||_{2,\infty} = \max_{1 \le j \le m} ||b_j||_2.$$

Even though ME is an extremely powerful and flexible model, it can potentially underfit or overfit if there are too few or too many experts in the model. We aim to alleviate this drawback by first initializing the model with a large number of experts and then adding to (2.3) the penalty term

$$\Omega_\lambda(\theta) = \lambda \sum_{1 \le i < j \le K} \left|\left| \begin{pmatrix} \beta_i \\ \gamma_i \end{pmatrix} - \begin{pmatrix} \beta_j \\ \gamma_j \end{pmatrix} \right|\right|_2, \tag{2.4}$$

where λ is a non-negative tuning parameter that controls the regularization level. By initializing the model with a large enough number of experts (the number of experts is a subjective choice; based on our experiments, 6 to 10 experts seemed reasonable for the datasets we considered in Section 5), we can ensure that, for some value of λ, the penalty will admit a model complex enough to fit the data. Next, we increase λ incrementally from 0. As λ gets larger, this penalty term will shrink the coefficients of different experts together as well as the coefficients governing the weights assigned to them by the gating network. When λ is large enough, pairs of experts and the corresponding pairs of functions in the gating network assigning weights to them will become identical, preventing overfitting and providing a natural way to choose the appropriate number of experts at the same time. Essentially we are fusing groups of coefficients together.

We find it useful to rewrite the penalty term in the following way. Define the matrix $D = D'C_p$, where D' is the $2P(2K-1) \times 2P(K-1)$ matrix given by

$$D' = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 1 & 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 & -1 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \otimes I_{2P}$$

and C_p is the matrix such that

$$C_p\theta = (\beta_1^T, \gamma_1^T, \beta_2^T, \gamma_2^T, ..., \beta_K^T, \underset{\sim}{0}^T)^T.$$

Then we may rewrite the penalty in (2.4) as

$$\Omega_\lambda(\theta) = ||D\theta||_{2,1},$$

which strongly resembles the penalty term used in the generalized lasso (Tibshirani, 2011). In the next section we will express the log-likelihood in a way that facilitates computation of the penalized maximization step.

2.3 Reformulating the likelihood

Since the M step in (2.3) can be divided into 2 parts, the experts part and the gating network part, in this section we will deal with each part separately.

2.3.1 The experts part

We now consider the last term in (2.3) since it is the only term in (2.3) that involves β. We will delay the treatment of σ until later since it is relatively easy to

find the update for σ and it does not appear in the penalty term. We may express the last term in (2.3) (without the negative sign) as

$$\mathbf{Q_E}(\beta, \sigma^{\mathbf{2}}) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} \sum_{k=1}^{K} h_k^{(n)} (Y^{(n)} - \beta_k^T X^{(n)})^2 = \sum_{k=1}^{K} ||W_k^{\frac{1}{2}} (Y - X\beta_k)||^2,$$

where $W_k = (2\sigma^2)^{-1} \text{diag}(h_k^{(1)}, \ldots, h_k^{(N)})$ for $k = 1, \ldots, K$. Letting

$$Y^* = \begin{pmatrix} W_1^{\frac{1}{2}} Y \\ \vdots \\ W_K^{\frac{1}{2}} Y \end{pmatrix} \text{ and } X^* = \begin{pmatrix} W_1^{\frac{1}{2}} X & & \\ & \ddots & \\ & & W_K^{\frac{1}{2}} X \end{pmatrix},$$

we may write

$$\mathbf{Q_E}(\beta, \sigma^{\mathbf{2}}) = \|Y^* - X^* \beta\|^2.$$

So in the case of no regularization, the EM algorithm update for the expert parameters is

$$\beta^{\text{new}} = \arg\min_{\beta} ||Y^* - X^* \beta||^2. \tag{2.5}$$

2.3.2 The gating network part

The first term in (2.3), the gating network part, is slightly more complicated, since we cannot express the update as the solution to a least-squares problem. We have

$$\mathbf{Q_g}(\gamma) = \sum_{n=1}^{N} \sum_{k=1}^{K} h_k^{(n)} \ln \left[\frac{e^{\gamma_k^T X^{(n)}}}{1 + \sum_{l=1}^{K-1} e^{\gamma_l^T X^{(n)}}} \right].$$

Maximizing the above term is quite similar to maximizing the likelihood for multinomial logistic regression and Cox regression, so we consider a Newton-Raphson algorithm. The first and second derivatives of Q_g are

$$\begin{aligned} \frac{\partial Q_g}{\partial \gamma_{ij}} &= \sum_{n=1}^{N} x_j^{(n)} \Big[h_i^{(n)} - p_i(x^{(n)}; \gamma) \Big] \\ \frac{\partial^2 Q_g}{\partial \gamma_{ij} \partial \gamma_{mp}} &= -\sum_{n=1}^{N} x_j^{(n)} x_p^{(n)} p_i(x^{(n)}; \gamma) \Big[I(i = m) + p_m(x^{(n)}; \gamma) \Big], \end{aligned}$$

for $1 \leq i, j, m, p \leq K-1$, where $p_i(x^{(n)}; \gamma) = e^{\gamma_k^T X^{(n)}} / [1 + \sum_{l=1}^{K-1} e^{\gamma_l^T X^{(n)}}]$. We can express the first and second derivatives in matrix form as

$$\frac{\partial Q_g}{\partial \gamma} = \tilde{X}^T (h - p) \quad \text{and} \quad \frac{\partial^2 Q_g}{\partial \gamma \partial \gamma^T} = -\tilde{X}^T W \tilde{X},$$

where $\tilde{X} = I_{K-1} \otimes X$ and $h = (h_1^T, \ldots, h_{K-1}^T)$ and $p = (p_1^T, \ldots, p_{K-1}^T)$, with

$$p_k = \Big(p_k(x^{(1)}; \gamma), \ldots p_k(x^{(N)}; \gamma)\Big)^T$$
$$h_k = \Big(h_k^{(1)}, \ldots, h_k^{(N)}\Big)^T,$$

for $k = 1 \ldots, K-1$, and $W = \Big(W_{ij}\Big)_{1 \leq i,j \leq K-1}$, where W_{ij} is an $N \times N$ diagonal matrix with diagonal elements given by

$$\begin{cases} p_i * (\mathbf{1}_N - p_i) & \text{if } i = j \\ -p_i * p_j & \text{if } i \neq j, \end{cases}$$

where we denote by $\mathbf{1}_N$ a vector of length N with all entries equal to 1 and by $*$ the element-wise multiplication between two vectors.

It is well-known that we may run into numerical issues when fitting multinomial logistic regression models using Newton-Raphson (Allison, 2008). It is indeed the case in ME since the matrix W almost always becomes computationally singular after some iterations. This is a problem because we need to invert W to compute the update. Böhning (Böhning, 1992) suggests to fix the Hessian ($-\tilde{X}^T W \tilde{X}$), or choose a fixed matrix, say $\overline{W}$, with which to replace W across all Newton-Raphson iterations. More specifically, we need to choose a fixed matrix $\overline{W}$ so that $(-\tilde{X}^T W \tilde{X}) - (-\tilde{X}^T \overline{W} \tilde{X})$ is positive semi-definite. The choice of $\overline{W}$ prescribed by Böhning has entries given by

$$\overline{W}_{ij} = \begin{cases} \left(\frac{1}{2} - \frac{1}{2K}\right) I_N & \text{if} \quad i = j \\ \left(-\frac{1}{2K}\right) I_N & \text{if} \quad i \neq j \end{cases} \qquad \text{for } 1 \leq i, j, \leq K-1.$$

With this choice of W, each Newton-Raphson update is (when there is no regularization)

$$\gamma^{new} = \gamma^{old} - \left(\frac{\partial^2 Q_g}{\partial \gamma \partial \gamma^T}\right)^{-1} \frac{\partial Q_g}{\partial \gamma} = \gamma^{old} + \left(\tilde{X}^T \overline{W} \tilde{X}\right)^{-1} \tilde{X}^T (h - p) = \left(\tilde{X}^T \overline{W} \tilde{X}\right)^{-1} \tilde{X}^T \overline{W} t,$$

where $t = \tilde{X}\gamma^{old} + \overline{W}^{-1}(h - p)$. Using the Cholesky decomposition $\overline{W} = LL^T$ of $\overline{W}$, the update γ^{new} can be expressed as the least-squares solution

$$\gamma^{new} = \underset{\gamma}{\operatorname{argmin}} ||\tilde{t} - \tilde{X}_2\gamma||^2, \tag{2.6}$$

where $\tilde{t} = L^T t$ and $\tilde{X}_2 = L^T\tilde{X}$.

2.4 Penalized Mixture of Experts

In this section, we introduce the penalty term into the reformulated likelihood.

2.4.1 Updating experts and the gating network together

Combining (2.5) and (2.6), we can further simplify the update of all coefficients of the experts and the gating network (again assuming there is no regularization) by writing

$$||Y^* - X^*\beta||^2 + ||\tilde{z} - \tilde{X}_2\gamma||^2 = \begin{pmatrix} Y^* - X^*\beta \\ \tilde{z} - \tilde{X}_2\gamma \end{pmatrix}^T \begin{pmatrix} Y^* - X^*\beta \\ \tilde{z} - \tilde{X}_2\gamma \end{pmatrix} = ||Y^{**} - X^{**}\theta||^2,$$

where

$$Y^{**} = \begin{pmatrix} Y^* \\ \tilde{z} \end{pmatrix} \quad \text{and} \quad X^{**} = \begin{pmatrix} X^* & . \\ . & \tilde{X}_2 \end{pmatrix}.$$

Then we may write

$$\theta^{new} = \arg\min_{\theta} ||Y^{**} - X^{**}\theta||^2. \tag{2.7}$$

Note that (2.5) is an update for each EM iteration while (2.6) is an update for each Newton Raphson iteration. We combine them together into (2.7) because in the penalty term, we cannot separate the coefficients of the gating network from experts' ones. Therefore we need to optimize both parts jointly. We discuss this in the next subsection.

2.4.2 Adding the penalty term

With the penalty term (2.4) added to the model, at each Newton-Raphson iteration in each Maximization step of the EM algorithm, we solve the optimization problem

$$\underset{\theta}{\text{minimize}}\ ||Y^{**} - X^{**}\theta||^2 + \lambda||D\theta||_{2,1}. \tag{2.8}$$

It is apparent that when $N > P$, X^{**} has full-column rank based on the way we construct X^{**}. With this result, we will employ the same strategy as the one in (Tibshirani, 2011) because our penalty is like the group-generalization of the generalized lasso penalty. The minimization problem (2.8) is equivalent to the problem

$$\underset{\theta}{\text{minimize}}\ ||Y^{**} - X^{**}\theta||^2 + \lambda||z||_{2,1} \quad \text{subject to } z = D\theta.$$

The Lagrangian form is

$$\mathcal{L}(\theta, z, u) = ||Y^{**} - X^{**}\theta||^2 + \lambda||z||_{2,1} + u^T(D\theta - z), \tag{2.9}$$

and the dual problem for this is

$$\underset{u}{\text{minimize}}\ ||\widetilde{Y} - \widetilde{D}^T u||^2 \text{ subject to } ||u||_{2,\infty} \leq \lambda, \tag{2.10}$$

where $\widetilde{Y} = X^{**}X^{**+}Y^{**}$ and $\widetilde{D} = DX^{**+}$. Here the pseudoinverse of a matrix A is calculated as $A^+ = (A^TA)^{-1}A^T$. Since we are in the low-dimensional setting where $N > P$, X^{**+} exists.

The dual problem in (3.4) is convex and thus can be solved using any convex solver. Nonetheless, we develop our own algorithm to solve it using blockwise coordinate descent. We minimize over each block of length $2P$ of u, with all other elements of u fixed, until convergence. The update for u_j is given by

$$u_j = T_\lambda\left[\left((\widetilde{D}^T_{\cdot j})^T\widetilde{D}^T_{\cdot j}\right)^{-1}(\widetilde{D}^T_{\cdot j})^T(\widetilde{y} - \widetilde{D}^T_{-\cdot j}u_{-j})\right] \quad \text{for } j = 1, 2, \ldots, \frac{K(K-1)}{2}, \tag{2.11}$$

where T_s is the truncating function

$$T_s(t) = \begin{cases} s * t/||t||, & \text{if } ||t|| > s \\ t, & \text{if } ||t|| \leq s. \end{cases}$$

Here u_j and u_{-j} denote the jth block of u and the vector u after removing the jth block, respectively. Similarly, $\widetilde{D}^T_{.j}$ and $\widetilde{D}^T_{-.j}$ represent the jth column-block of the matrix $\widetilde{D}^T$ and the whole matrix $\widetilde{D}^T$ after removing the jth column-block, respectively. We obtain (2.11) by first differentiating the least-squares term in (3.4) with respect to u_j, setting the first derivative to 0, and solving for u_j. Since we have a box constraint on u, we apply the truncating function to this value of u_j.

After u has converged, we recover the primal solution via

$$\theta^{new} = X^{**+}(\widetilde{y} - \widetilde{D}^T u^{new}).$$

This primal-dual relationship is derived by taking the gradient of (3.3) with respect to θ and setting this equal to 0.

2.4.3 The full algorithm

We can now put everything together to get the algorithm to fit the penalized ME. Full details are given in Algorithm 1.

We obtain the expression in (2.12) by differentiating (2.3) with respect to σ, setting it to 0, and solving for σ.

2.4.4 Efficient way to initialize the parameters

The log-likelihood of ME is not convex, meaning that different starting values may affect the final solution. Based on our experiments, choosing small initial values (such as 0.1) for all elements of θ seems to converge to a good solution in most cases but the computation time may be long.

Algorithm 1 The algorithm to fit the penalized mixture of experts model.

Choose a value of lambda λ
Choose the maximum number of experts K
Initialize θ, σ and calculate $h, Y^*, X^*, p, W, z, L, \tilde{z}, \tilde{X}_2, Y^{**}, X^{**}, \widetilde{Y}, \widetilde{D}$
while *θ and σ are not converged (EM loop)* **do**
    while *γ is not converged (Newton-Raphson loop)* **do**
        Solve the dual problem using block coordinate descent:

$$\underset{u}{\text{minimize}} \; ||\widetilde{Y} - \widetilde{D}^T u||^2 \text{ subject to } ||u||_{2,\infty} \leq \lambda$$

        With the new u, recalculate $\theta, p, W, t, L, \tilde{t}, \tilde{X}_2, Y^{**}, X^{**}, \widetilde{Y}, \widetilde{D}$
    end
    Update σ

$$\sigma^{new} = \sqrt{\sum_{n=1}^{N}\sum_{k=1}^{K} h_k^{(n)} \left(Y^{(n)} - (\beta_k^{new})^T X^{(n)}\right)^2 \Big/ N} \tag{2.12}$$

    With the new θ and σ, recalculate $h, Y^*, X^*, t, \tilde{t}, Y^{**}, X^{**}, \widetilde{Y}, \widetilde{D}$
    end

One way to improve the choice of initial values is to initialize β by k-means clustering and linear regression with a ridge penalty. Specifically, first we apply k-means clustering to the covariate X, with k equal to the maximum number of experts we want to fit in ME. Then we fit a linear regression for data points in each cluster with a small ridge penalty (with ridge-penalty tuning parameter equal to, say 0.0001). As the result, we obtain k sets of coefficients from k ridge-penalized linear regressions and we can use those coefficients as initial values for experts in ME. Xing and Hu in (Xing and Hu, 2008) show that this strategy speeds up the convergence significantly for the unpenalized ME. The reason we choose to use ridge regression is that the number of observations in each cluster may be less than the number of covariates, making it impossible to fit least-squares linear regression. In the case when there is one or more clusters which only contain one class of a categorical covariate (for example: say we have a gender covariate which has 2 classes (male and female). It can happen that after doing clustering, observations in one particular cluster are

all males), we fit a ridge linear regression to the whole dataset again with a small penalization parameter. We then use the coefficients of that categorical covariate obtained from the ridge regression fitted on the whole dataset as the initial value for that particular covariate in those clusters.

2.4.5 High-dimensional situation

So far we have considered only the $N > P$ case. However, the case of $P > N$, or the high-dimensional setting, is becoming increasingly common. When $P > N$, as (Tibshirani, 2011) points out, there is a small complication for the dual problem of (2.8) since X^{**} is no longer full-column rank. To handle this situation, a quick fix is to modify the penalty term in (2.4) by the addition of a small ridge penalty so that the penalty becomes

$$\Omega_\lambda^h(\theta) = \lambda||D\theta||_{2,1} + \epsilon||\theta||_2^2, \tag{2.13}$$

where ϵ is a small positive constant we choose. With this modified term, the minimization problem in (2.8) becomes

$$\underset{\theta}{\text{minimize}}\ ||Y^{**} - X^{**}\theta||^2 + \lambda||D\theta||_{2,1} + \epsilon||\theta||_2^2.$$

This is equivalent to minimizing

$$||Y^{***} - X^{***}\theta||^2 + \lambda||D\theta||_{2,1},$$

where $Y^{***} = (Y^{**}, 0)^T$ and $X^{***} = [(X^{**})^T, \sqrt{\epsilon}I]^T$. Since X^{***} has full-column rank, we can proceed with the same strategy discussed in Section 4.2. Besides, we can choose ϵ to be extremely small so that the difference in solutions between using $\Omega_\lambda^h(\theta)$ and $\Omega_\lambda(\theta)$ is likely to be negligible.

2.5 Numerical results

2.5.1 Illustration of penalty on a simulated data set

In this section we give a brief illustration of how penalized ME works in a simple example. The data have a single covariate and 3 experts are enough to capture the relationship between the predictor and the response. Nevertheless, we will initialize the model with 6 experts. Then we will incrementally increase the value of λ (from 0 to 2.5) to make the coefficients of experts and gating network closer together. As a result, we can see that the regression curve representing the conditional mean of the response given the covariate becomes smoother and smoother. Eventually, it becomes a straight line when λ is big enough to make all experts the same. The fitted models are depicted in Figure 1.

Table 2.1: Models to compare with penalized ME, their tuning parameters and R packages.

Models	**Tuning parameters**	**R package**
Elastic net	λ and α	glmnet
Decision tree	Complexity	rpart
Random forest	Number of predictors used at each split	randomForest
Gradient boosting	Number of trees	gbm
Gaussian process	Kernel	kernlab

Table 2.2: Test MSE for different models in the original unit.

	Boston	**Galaxy**	**Air**	**Diabetes**	**Prostate**
Elastic net	18.05	810.76	391.74	3169.72	18.45
Decision tree	17.10	339.30	603.97	3592.72	43.46
Random forest	8.12	233.36	240.90	3343.69	19.91
Gradient boosting	11.21	328.38	247.33	3678.71	25.01
Gaussian process	10.45	225.37	322.18	3438.42	28.55
ME	10.11	326.57	304.06	3266.83	34.87
Penalized ME	10.11	324.18	303.83	3186.38	32.39

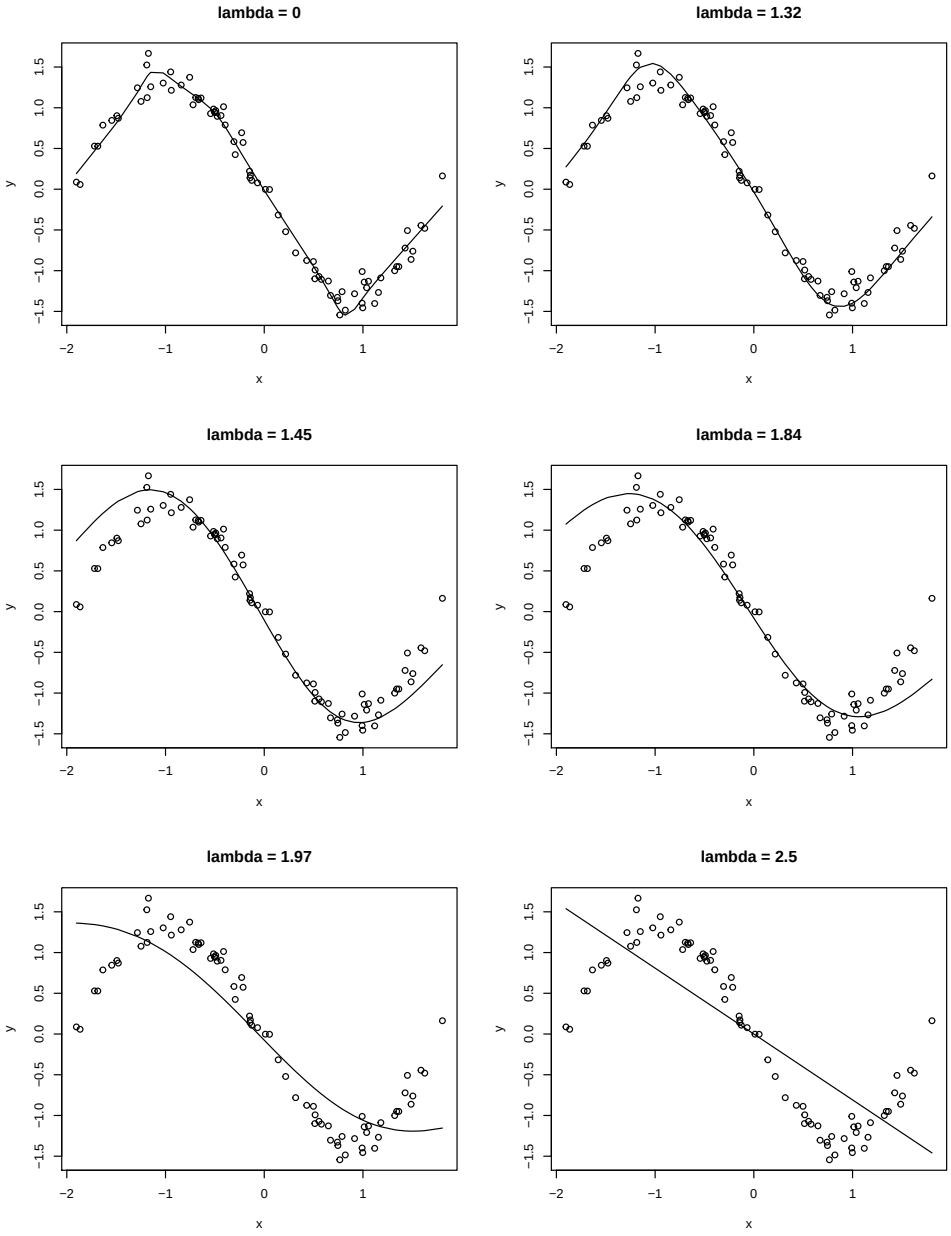

Figure 2.1: Penalized ME curves at different values of the penalty parameter λ.

2.5.2 Real-world applications

In this section we apply our model to the following 5 real-world regression datasets.

- Median housing price in Boston (dimension: 506x13) (Harrison Jr and Rubinfeld, 1978)
- Radial Velocity of Galaxy NGC7531 (dimension: 323x3) (Buta, 1987)
- Air quality (dimension: 111x5) (Chambers, 2017)
- Diabetes progression (dimension: 442x10) (Efron et al., 2004)
- Prostate (dimension: 97x8) (Stamey et al., 1989)

For the second dataset, we remove all incomplete observations. We split each dataset into: training set (70%), validation set (15%) and test set (15%), and we compare the performance of the penalized ME to six other commonly used machine learning models that are listed in Table 1. These methods are tuned using the validation set, except for the elastic net which uses 10-fold cross validation on the combined training and validation set. For the penalized ME, we tune the value of λ over the range 0 to 2.5. We then train the model on the combined training and validation set using the chosen tuning parameters and make predictions on the test set.

We will also fit an unpenalized ME with 6 experts to see whether adding a penalty term helps. Table 2 displays MSEs on the testing data sets for different models. As we can see, the concept of no free lunch still applies since there is no method that wins in all datasets. Nonetheless random forest performs particularly well in almost all cases. The ME and penalized ME also do well compared to other models. Comparing the ME to the penalized ME, we see that in all situations, adding the penalty term improved the prediction accuracy, as the test MSE of the penalized ME was in every case less than or equal to that of the unpenalized ME.

2.6 Conclusion

Mixture of experts is a powerful and flexible machine learning method. In this chapter, we have proposed adding a fusion penalty term to the likelihood function with the goal of penalizing the difference between the parameters of different experts. By doing this we can avoid overfitting and choose the best set of parameters at the same time. This has been illustrated above as penalized ME outperformed the unpenalized version in all data sets considered and also performed competitively when compared to other popular machine learning methods.

Chapter 3

Alternating Minimization and Dual Path algorithms for Group Fused Lasso

3.1 Introduction

The use of regularization to fight overfitting has been extremely popular in recent years. We can easily find regularized versions of almost all statistical and machine learning procedures. The two most commonly used penalty norms in these papers are L1 and L2 largely thanks to their convexity. The earliest and arguably the most famous use of L1 norm in statistics is the Lasso (Tibshirani, 1996) which has the form

$$\underset{\beta}{\text{minimize}}\ \frac{1}{2}||z - X\gamma||^2 + \lambda \sum_{i=1}^{P} |\gamma| \ ,$$

where z is the response vector of length N, X is a covariate matrix of dimension $N \times p$, γ is the parameter vector of length P and λ is the tuning parameter that controls the level of regularization. One appealing property of the Lasso solution is its sparsity. When λ gets large enough, some elements of β will become zeros, resulting in a sparse model. Therefore, unlike ridge regression, which uses L2 norm, Lasso can do model selection and shrinkage at the same time. To get the whole solution path of Lasso, one can use the modified LARS algorithm (Efron et al., 2004) or the coordinate descent algorithm (Friedman et al., 2010).

A noticeable extension of the Lasso is the Fused Lasso Signal Approximator (FLSA) (Tibshirani et al., 2005), which take the form

$$\underset{\beta}{\text{minimize}}\ \frac{1}{2}\sum_{i=1}^{N}(z_i-\gamma_i)^2+\lambda\sum_{i=2}^{n}|\gamma_i-\gamma_{i-1}|.$$

The penalty term in FLSA encourages identicalness between adjacent coefficients, resulting in a piecewise-constant solution. FLSA has wide applications in signal recovery, change-point detection, smoothing, genomic segmentation and more. There are many fast and efficient algorithms to solve FLSA in the literature. The two fastest ones that we are aware of are (Johnson, 2013) which uses dynamic programming and (Davies and Kovac, 2001) which applies a taut string principle. Other well-known methods include using coordinate descent (Friedman et al., 2007) and tracing out the whole solution path (Hoefling, 2010).

In this chapter, we will focus on a generalization of FLSA, called Group Fused Lasso (GFL). The loss function we minimize in GFL is

$$L(\beta)=\frac{1}{2}\sum_{i=1}^{N}||y_i-\beta_i||^2+\lambda\sum_{i=2}^{N}||\beta_i-\beta_{i-1}||. \tag{3.1}$$

Here we assume $y_i,\beta_i\in\mathbb{R}^m, i=1,2,...,N$. If we think of the observed response z in FLSA model as a time series or a profile in a certain time interval, then GFL handles cases in which we have multiple profiles in that particular time interval. The penalty term in GFL has the same function as the one in FLSA that is penalizing the differences between adjacent coefficients. Since β_i's are vectors, L2 norm is used to achieve group sparsity. Note that GFL reduces to FLSA when $m=1$. In this chapter we will propose two methods to solve the whole solution path of (3.1) efficiently, namely Alternating Minimization (AM) and Dual Path (DP).

The chapter is organized as follows: In Section 2, we give some key properties of GFL's solution that will be exploited in our algorithms. Section 3 talks about Alternating Minimization algorithm while the Dual Path algorithm is discussed in

the following section, Section 4. Lastly, we provide an empirical evaluation of our methods compared to other existing ones in terms of speed.

3.2 Properties of GFL's solution path

First we define two norms that will be used later. Given an arbitrary vector $b = (b_1^T, ..., b_k^T)^T$ where each block $b_i, i = 1, 2, ...k$, has length m, let

$$||b||_{2,1} = \sum_{j=1}^{k} ||b_j|| \quad \text{and} \quad ||b||_{2,\infty} = \max_{1 \le j \le k} ||b_j||.$$

Now we discuss some crucial properties of GFL's solution that will be used later in the chapter.

3.2.1 Minimum value of λ that makes all β_i's identical

In (3.1), when $\lambda = 0$, the solution is trivially $\beta_i = y_i, i = 1, 2, ..., N$. When λ gets large enough, all β_i's will be identical and their common value can easily be shown to be $\sum_{i=1}^{N} y_i/N$. The first step to compute the solution path is to know the range of λ, or essentially to know the minimum value of λ that makes all β_i's same (call it λ_{max}). We then just need to find the solution for GFL with λ ranging between 0 and λ_{max} since if we keep increasing λ past λ_{max}, the solution stays the same.

We rewrite (3.1) as

$$L(\beta) = \frac{1}{2} \sum_{i=1}^{N} ||y_i - \beta_i||^2 + \lambda ||D\beta||_{2,1}, \tag{3.2}$$

where β is $(\beta_1^T, \beta_2^T, ...\beta_N^T)^T$ and D is

$$D = \begin{bmatrix} 1 & -1 & 0 & \dots & 0 & 0 \\ 0 & 1 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 \end{bmatrix} \otimes I_m.$$

Following an argument in (Tibshirani, 2011), we reformulate (3.2) as

$$L(\beta, c) = \frac{1}{2}\sum_{i=1}^{N} ||y_i - \beta_i||^2 + \lambda||c||_{2,1} \qquad \text{subject to } c = D\beta.$$

The Lagrangian is then

$$\mathcal{L}(\beta, c, u) = \frac{1}{2}\sum_{i=1}^{N} ||y_i - \beta_i||^2 + \lambda||c||_{2,1} + u^T(c - D\beta). \tag{3.3}$$

The corresponding dual problem of (3.3) is

$$\underset{u}{\text{minimize}}\ \frac{1}{2}||y - D^T u||^2 \quad \text{subject to } ||u||_{2,\infty} \leq \lambda, \tag{3.4}$$

where $y = (y_1^T, y_2^T, ...y_N^T)^T$. We observe that (3.4) can be optimized using block coordinate descent. Specifically, we minimize over each block of length m of u, with all other elements of u fixed, until convergence, using the formula

$$u_j = T_\lambda\left[\left((D_{.j}^T)^T D_{.j}^T\right)^{-1}(D_{.j}^T)^T(y - D_{-.j}^T u_{-j})\right] \quad \text{for } j = 1, 2, ..., N-1. \tag{3.5}$$

where $D_{.j}^T$ represents the jth column block of D^T and T is a truncating function satisfying that

$$T_s(t) = \begin{cases} s * t/||t|| & \text{if} \quad ||t|| > s \\ t & \text{if} \quad ||t|| \leq s \end{cases}.$$

We denote the primal and dual solution as $\hat{\beta}$ and $\hat{u}$ respectively. By differentiating the Lagrangian with respect to β, setting this to 0, we have the primal-dual relationship

$$\hat{\beta}_\lambda = y - D^T\hat{u}_\lambda. \tag{3.6}$$

With (3.5) and (3.6), we are ready to compute λ_{max}. First, assume $\lambda = \lambda^*$ and the solution is $\hat{\beta}_{i,\lambda^*} = \sum_{j=1}^{N} y_j/N$ for $i = 1, 2, ...N$. This mean that $\lambda^* \geq \lambda_{max}$. We get the corresponding dual solution by using the formula

$$\hat{u}_{\lambda^*} = (D^T)^+(y - \hat{\beta}_{\lambda^*}).$$

where $(D^T)^+$ is the Moore−Penrose inverse of D^T. Since $\hat{u}_{\lambda^*}$ is the solution, if we keep updating blocks of $\hat{u}_{\lambda^*}$ using (3.5), $\hat{u}_{\lambda^*}$ should stay the same. This can only happen when $||\hat{u}_{i,\lambda^*}|| \leq \lambda^*$ for $i = 1, 2, ..., N-1$. Therefore, $\lambda_{max} = \max_{1,2,...,N-1} ||\hat{u}_{i,\lambda^*}||$. In the next section when we try to compute the solution path, we will construct an increasing sequence of L (typically 100) values of λ from 0 to λ_{max} and solve the GFL at these particular values of λ.

3.2.2 Permanent fusion of adjacent coefficients

We now present and prove a useful result that will facilitate the fast computation of the GFL's solution path in the next section. The similar result for FLSA (GFL when $m = 1$) is provided in (Friedman et al., 2007).

Theorem 1: If $\hat{\beta}_{i,\lambda'} = \hat{\beta}_{i+1,\lambda'}$, then for any $\lambda > \lambda'$, we have $\hat{\beta}_{i,\lambda} = \hat{\beta}_{i+1,\lambda}$

Proof: Our proof is largely based on the proof of Proposition 2 in (Friedman et al., 2007). The subgradient equations of (3.1) are

$$\beta_1 - y_1 - \lambda d_2 = 0,$$
$$\beta_i - y_i + \lambda(d_i - d_{i+1}) = 0, \qquad i = 2, ..., N,$$

where $d_i = (\beta_i - \beta_{i-1}) \big/ ||\beta_i - \beta_{i-1}||$ if $\beta_i \neq \beta_{i-1}$ and $d_i \in \{r; ||r|| \leq 1\}$ if $\beta_i = \beta_{i-1}$. These N equations fully characterize the solution of the GFL (Bertsekas, 1997). Now suppose we have $\hat{\beta}_{i-h-1} \neq \hat{\beta}_{i-h} = \hat{\beta}_{i-h+1} = ... = \hat{\beta}_i \neq \hat{\beta}_{i+1}$ for some i and h. This implies that $||d_{i-h}|| = ||d_{i+1}|| = 1$ and $||d_k|| \leq 1$ for $k \in \{i-h+1, ..., i\}$. Taking differences between adjacent equations involving $d_{i-h+1}, ..., d_i$ gives

$$Ad = \frac{1}{\lambda'}\delta y + E,$$

where $d = (d_{i-h+1}^T, ..., d_i^T)^T$, $E = (d_{i-h}^T, 0_{hm-2m}^T, d_{i+1}^T)^T$ (0_{hm-2m} is zero vector of length $hm - 2m$),

$$A = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 & 0 & 0 \\ -1 & 2 & -1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & 0 & \dots & 0 & -1 & 2 \end{bmatrix} \otimes I_m = B \otimes I_m,$$

and

$$\delta y = \begin{bmatrix} y_{i-h+1} - y_{i-h} \\ y_{i-h+2} - y_{i-h+1} \\ \vdots \\ y_{i-1} - y_{i-2} \\ y_i - y_{i-1} \end{bmatrix}.$$

We have $d = \left(\frac{1}{\lambda'} A^{-1}\delta y + A^{-1}E\right)$ and each m-length component of d has norm 2 less than or equal to 1. We focus on the second term $A^{-1}E = (B^{-1} \otimes I_m)E$. The explicit form of B^{-1} satisfies that the first and the last elements of any row are positive and they sum to 1 (Schlegel, 1970). Since all m-length components of E also have L2 norm less than or equal to 1, the weighted sum of any two of those components with positive and less than 1 weights will result in a vector with L2 norm less than or equal to 1. Therefore, all m-length components of $(A^{-1}E)$ also have L2 norm less than or equal to 1 as well. When we increase λ from λ', $\frac{1}{\lambda} A^{-1}\delta y$ becomes smaller in absolute value elementwise while $(A^{-1}E)$ are unaffected. As a result, each m-length component of d still has L2 norm less than or equal to 1, implying that identical coefficient vectors stay identical when λ increases. □

3.3 Alternating Minimization algorithm

In this section we discuss about the AM algorithm in general and how it can be applied to GFL.

3.3.1 General formula for AM algorithm

Alternating Minimization is a variant of augmented Lagrangian method. Suppose we are trying to optimize the following problem

$$\begin{aligned} &\text{minimize } f(x) + g(v) \\ &\text{subject to } Ax + Bv = k. \end{aligned}$$

The Lagrangian is

$$\mathcal{L}(x, v, p) = f(x) + g(v) + p^T(k - Ax - Bv).$$

where p is a Lagrangian multiplier. The update of x, v and p respectively are

$$\begin{aligned} x^{new} &= \arg\min_x \mathcal{L}(x, v, p) \\ v^{new} &= \arg\min_x \left[\mathcal{L}(x, v, p) + \frac{\nu}{2}||k - Ax - Bv||^2\right] \\ p^{new} &= p^{current} + \nu(k - Ax^{new} - Bv^{new}), \end{aligned}$$

where ν is an optimization parameter we choose. We keep iterating until convergence. A complete treatment of AM can be found in (Tseng, 1991).

3.3.2 Reformulation of GFL

In this subsection we work with a slight variant of GLF that takes the form

$$\underset{\beta}{\text{minimize}} \ \frac{1}{2}\sum_{i=1}^{N} w_i||y_i - \beta_i||^2 + \lambda\sum_{i=2}^{N}||\beta_i - \beta_{i-1}||, \tag{3.7}$$

where $w_i, i = 1, 2, ..., N$ are known scalars that represent the weight for each observed response vector y_i. The reason behind the use of such formula will be made clear in

the next subsection. We also denote w as $(w_1, w_2, ..., w_N)^T$. Clearly, when all w_i's are equal to 1, we have the GFL back.

First we recast (3.7) as a constrained optimization problem to fit in AM framework

$$\begin{aligned} &\underset{\beta}{\text{minimize}}\ \frac{1}{2}\sum_{i=1}^{N} w_i||y_i - \beta_i||^2 + \lambda\sum_{i=2}^{N}||v_i|| \\ &\text{subject to } \beta_i - \beta_{i-1} - v_i = 0. \end{aligned}$$

AM updates β by minimizing the Lagrangian, namely

$$\underset{\beta}{\text{minimize}}\ \frac{1}{2}\sum_{i=1}^{N} w_i||y_i - \beta_i||^2 + \sum_{i=2}^{N} p_i^T(\beta_i - \beta_{i-1} - v_i).$$

The solution for this minimization problem is

$$\beta_i^{new} = \begin{cases} y_i - p_i^{current}/w_i & \text{if} \quad i = 1 \\ y_i + (p_{i-1}^{current} - p_i^{current})/w_i & \text{if} \quad i = 2, ..., N-1 \\ y_i + p_{i-1}^{current}/w_i & \text{if} \quad i = N \end{cases} .$$

The update for $v_i, i = 2, ..., N$ is

$$\begin{aligned} v_i^{new} &= \arg\min_{v_i}\left[\lambda||v_i|| + \frac{\nu}{2}||\beta_i - \beta_{i-1} - v_i||^2 - p_i^T(\beta_i - \beta_{i-1} - v_i)\right] \\ &= \arg\min_{v_i}\left[\frac{1}{2}||v_i - \beta_i^{new} + \beta_{i-1}^{new} - \frac{p_i^{current}}{\nu}||^2 + \frac{\lambda}{\nu}||v_i||\right] \\ &= \left(1 - \frac{\lambda}{\nu}\Big/||\beta_i^{new} - \beta_{i-1}^{new} + \frac{p_i^{current}}{\nu}||\right)_+ \Big(\beta_i^{new} - \beta_{i-1}^{new} + \frac{p_i^{current}}{\nu}\Big). \end{aligned}$$

The last equality is the proximal map of L2 norm (Chi and Lange, 2015). Lastly, the update of $p_i, i = 2, ..., N$ is obtained directly from the general AM formula, which is

$$p_i^{new} = p_i^{current} + \nu(v_i^{new} - \beta_i^{new} + \beta_{i-1}^{new}).$$

We keep iterating between β_i's, v_i's and p_i's until convergence.

3.3.3 Fusion of neighboring coefficients along the solution path

Theorem 1 says that once two or more neighboring coefficient vectors are fused (identical), they will never become different again as we move forward along the path (λ increases). With this nice property, we can discard all but one of those identical coefficient vectors and collapse the problem into one with fewer parameters.

More specifically, assume we are currently at $\lambda = \lambda'$ along the solution path and we have removed k observations before we get to λ'. Our original GFL problem then becomes

$$\underset{\beta}{\text{minimize}} \ \frac{1}{2} \sum_{i=1}^{N-k} w_i ||y_i - \beta_i||^2 + \lambda' \sum_{i=2}^{N-k} ||\beta_i - \beta_{i-1}||.$$

Note that we start at $\lambda = 0$ with all w_i's equal 1 and $k = 0$. Now if after solving the above optimization we have $\hat{\beta}_i = \hat{\beta}_{i-1}$, we modify the problem as follows:

- $\bar{y} \leftarrow (w_{i-1}y_{i-1} + w_i y_i)/(w_{i-1} + w_i)$.
- $\bar{w} \leftarrow w_{i-1} + w_i$.
- $y_{i-1} \leftarrow \bar{y}$.
- $w_{i-1} \leftarrow \bar{w}$.
- Remove y_i, β_i, w_i and shift all indices greater than i to the left by 1.

Since we merge the weights of coefficient vectors that are the same, some weights will be bigger than 1. Therefore, in the previous subsection, we have to deal with the formula (3.7) instead of the original GLF's loss function. Also, by removing observations and their corresponding coefficients, we have fewer and fewer parameters to solve as we move towards the end of the path. This fact greatly speeds up the computation of the solution path of GFL.

3.3.4 Other details

We implement acceleration for our AM algorithm using Nesterov technique described in (Goldstein et al., 2014). We also choose ν using the Anderson-Morely upper bound (Anderson Jr and Morley, 1985). One practical trick to speed up convergence is to initialize β_i at a particular value of λ to be $2a - b$ where a and b are the values of β_i at the last λ and the second last λ respectively.

3.4 Dual Path algorithm

In this section, we approach the GFL problem in a different direction, namely its dual problem.

3.4.1 Solving the dual problem

We also work with (3.7) in this section, for the same reason mentioned in the previous section. The approach to obtain the dual problem for (3.7) is very similar to the one described in Subsection 2.1. The dual problem of (3.7) is

$$\underset{u}{\text{minimize}}\ \frac{1}{2}||Wy - W^{-1}D^T u||^2 \quad \text{subject to } ||u||_{2,\infty} \leq \lambda,$$

where $W = W' \otimes I_m$ and W' is a diagonal matrix with the diagonal being the vector w. To optimize this problem, we minimize over each block of length m of u, with all other elements of u fixed, until convergence, using the formula

$$u_j^{new} = T_\lambda\left[\left((\widetilde{D}_{.j}^T)^T\widetilde{D}_{.j}^T\right)^{-1}(\widetilde{D}_{.j}^T)^T(\widetilde{y} - \widetilde{D}_{-.j}^T u_{-j}^{current})\right] \quad \text{for } j = 1, 2, ..., N-1.$$

where $\widetilde{D}_{.j}^T = W^{-1}D^T$ and $\widetilde{y} = Wy$. The relationship between primal-dual solutions is

$$\hat{\beta}_\lambda = y - W^{-1}D^T\hat{u}_\lambda. \tag{3.8}$$

3.4.2 Fusion of neighboring primal coefficients along the solution path

As discussed in Subsection 3.3.3, as we move forward along the path, some coefficient vectors will become fused. This leads to a different form of the GFL's primal problem and consequently a different form of the dual problem as well. Specifically, in addition to the modifications in Subsection 3.3.3, we also need to:

- Remove the ith row and ith column of W'.
- Recalculate W.
- Remove the $(i-1)$th block of u
- Remove the last m rows and columns of D

Here we also have fewer parameters (fewer blocks of u) to solve as we move further along the path, reducing the computational burden.

3.4.3 Other details

Similar to the AM algorithm, significant speedup can be achieved by initializing u_i at a particular value of λ to be $2a - b$ where a and b are the values of u_i at the last λ and the second last λ respectively.

3.5 Numerical results

In this section we evaluate our two algorithms with other existing methods in terms of speed. Both simulated and real datasets are used. The two competitors we will consider are the algorithms in (Bleakley and Vert, 2011). More specifically, they convert (3.1) into a group lasso problem and either solve it directly (GL) or approximate the solution path using group-LARS (GLARS). There are other proposed solvers for (3.1), namely (Alaíz et al., 2013) and (Wytock et al., 2014). However, we

cannot find the implementations of their methods and their focus are mainly to solve (3.1) at a certain value of λ, not the entire solution path.

We also want to note that our two algorithms are coded entirely in C++ with Armadillo library (Sanderson and Curtin, 2016) while the implementation of (Bleakley and Vert, 2011) is in Matlab. All numerical computations were carried out on a Intel Core i5 Macbook Air.

3.5.1 Simulated datasets

We simulate datasets with different sample sizes N and different numbers of profiles m. For the first profile, the first 30% of observations have values 0, the next 30% are 3 and the remaining 40% are -1. The values of datapoints in each additional profile will be the values of the last profile plus 0.5. All datapoints are corrupted by white noise from standard normal distribution. Figure 1 shows an example where $N = 100$ and $m = 3$. We limit the maximum sample size n to be 1000 solely because the time it takes GL to run becomes unmanageable at bigger sample sizes. The timings for different methods under different setups are presented in Table 3.1.

Table 3.1: Timings (sec) for different optimization methods on different simulation settings

m=1						m=2				
N	10	50	100	500	1000	10	50	100	500	1000
GL	0.04	0.42	1.37	47.63	227	0.03	0.44	1.48	59.67	387
GLARS	0.005	0.02	0.03	0.16	0.41	0.005	0.02	0.03	0.16	0.42
AM	0.003	0.013	0.023	0.60	2.88	0.009	0.04	0.07	1.20	5.93
DP	0.021	0.08	0.33	16.08	95.2	0.025	0.12	0.53	21.36	127

m=3						m=4				
N	10	50	100	500	1000	10	50	100	500	1000
GL	0.04	0.42	1.46	74.63	508	0.05	0.48	1.52	96.57	712
GLARS	0.006	0.02	0.04	0.17	0.45	0.007	0.03	0.03	0.18	0.48
AM	0.015	0.12	0.14	2.31	10.3	0.015	0.13	0.18	3.18	17.2
DP	0.027	0.17	0.57	26.29	150	0.027	0.17	0.66	33.52	178

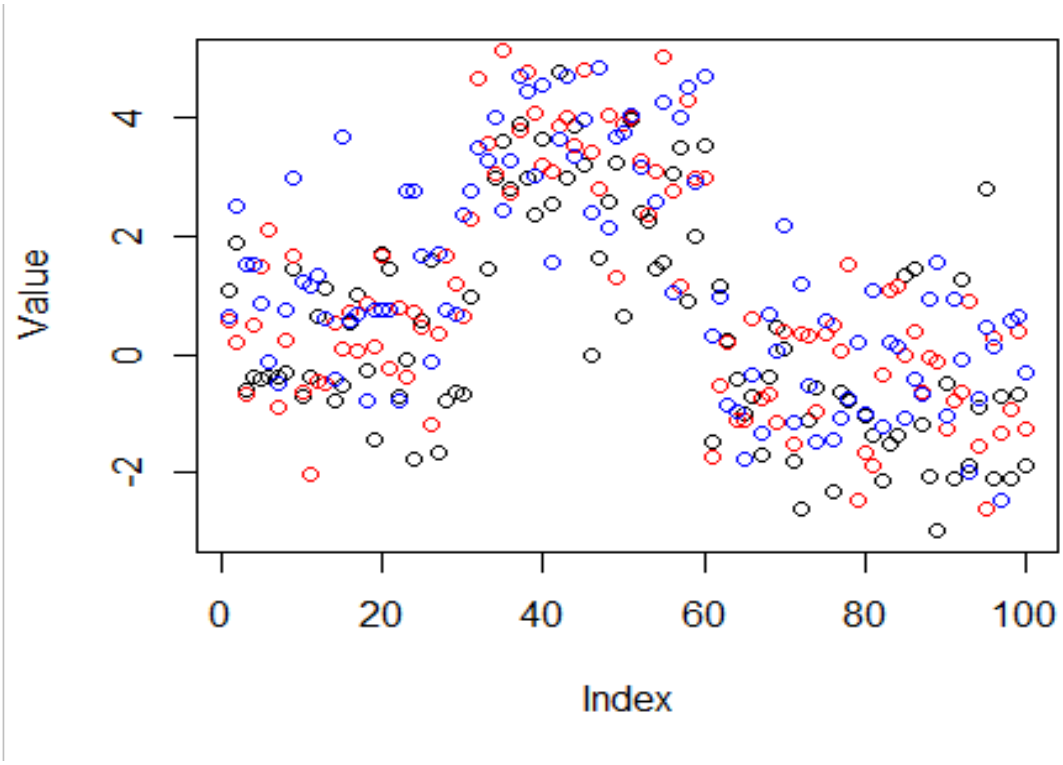

Figure 3.1: Example of a simulated dataset with $N = 100$ and $m = 3$. Points with the same color belong to the same profile.

We can see that the AM algorithm totally outperforms the exact solver GL. On the simulation settings with the largest sample size, the AM is faster than GL by at least 50 times. DP is slower than AM but the former is still significantly faster than GL, especially as we increase m. We want to note that even though GLARS seems to be incredibly fast, this method only solves an approximation of (3.1). A property of this approximation that allows fast computation is the piecewise linear property of the solution path. In general, the solution path for group lasso (the problem that GL solves) is not piecewise linear (Yuan and Lin, 2006).

3.5.2 Real-world datasets

The main application of Group Fused Lasso and Fused Lasso in general is smoothing and change-point detection in mean. Therefore we choose two datasets in which it is of interest to detect change-points accurately and efficiently. The first dataset is

the microarray aCGH (James and Matteson, 2013). This dataset contains microarray data of 43 patients who suffer from bladder tumor. Since patients share the same disease, we can use the aggregated data to detect the change-points for all of them. A change-point is defined to be a change in mean of aCGH number of copies. Figure 3.2 shows the aCGH data for the first three patients. The second dataset includes the weekly log returns of the top 30 companies (excluding Kraft Foods Inc.) whose stock prices are used to calculate the Dow Jones Industrial Average (DJIA) (James and Matteson, 2013). Each stock has 1139 observations, representing weekly data from 04/1990 to 01/2012. Just like in the first dataset, we expect the price of these stocks to be influenced by the same macro environment so we can use all data to detect the change-points in price jointly. Figure 3.3 shows the weekly log returns for the first 3 companies. We consider a maximum of 20 individuals at a time in the first dataset and 20 stocks at the time in the second dataset. Table 3.2 contains the timing results for all 4 optimization methods on the 2 datasets. All results in the subsection are obtained on a c5d.xlarge instance on Amazon Web Services. Again we can see that the AM and DP algorithms perform much better than GL in terms of speed.

Table 3.2: Timings (sec) for different optimization methods on 2 real-world datasets. (*GLARS only provides the approximated solution.)

Dataset	Method	m=1	m=2	m=3	m=5	m=10	m=20
aCGH	GL	83	148	208	330	1209	5374
	GLARS	0.007	0.009	0.013	0.21	0.91	3.17
	AM	0.21	0.74	2.85	5.23	25.6	113.8
	DP	14.9	26.7	32.3	52.1	194	880
DJIA	GL	42	76	101	134	623	3081
	GLARS	≈ 0	0.008	0.026	0.074	0.23	1.62
	AM	0.09	0.73	1.91	3.50	17.2	72.9
	DP	8.49	12.7	19.5	31.4	148	572

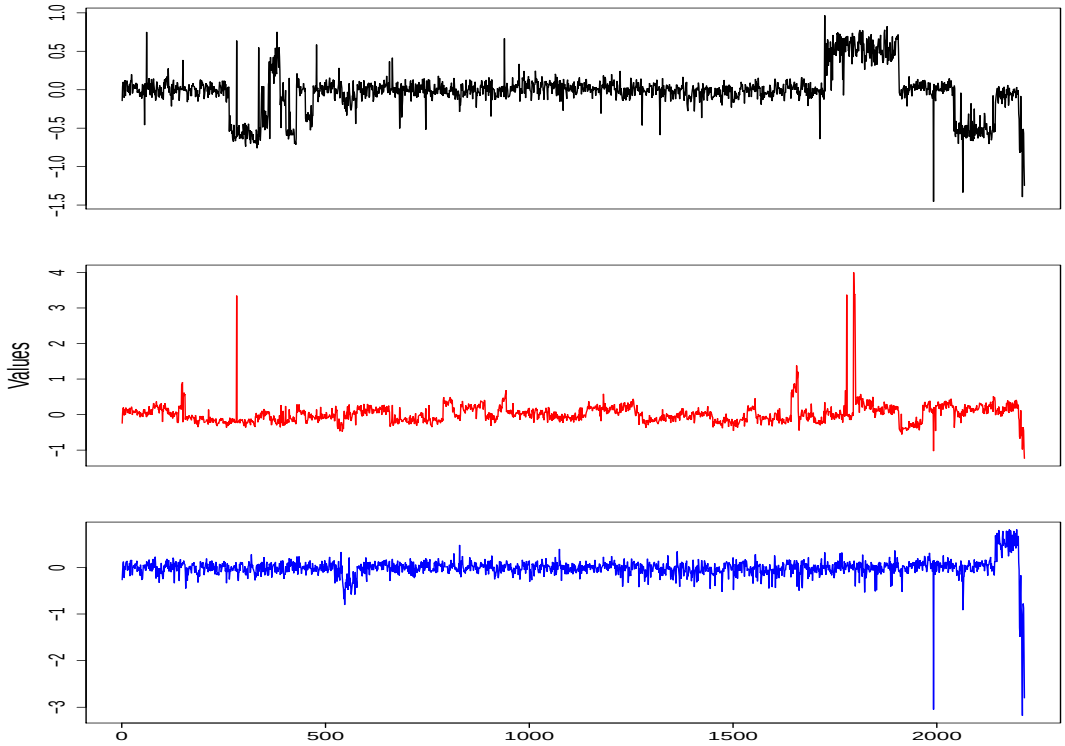

Figure 3.2: The aCGH data for the first 3 individuals

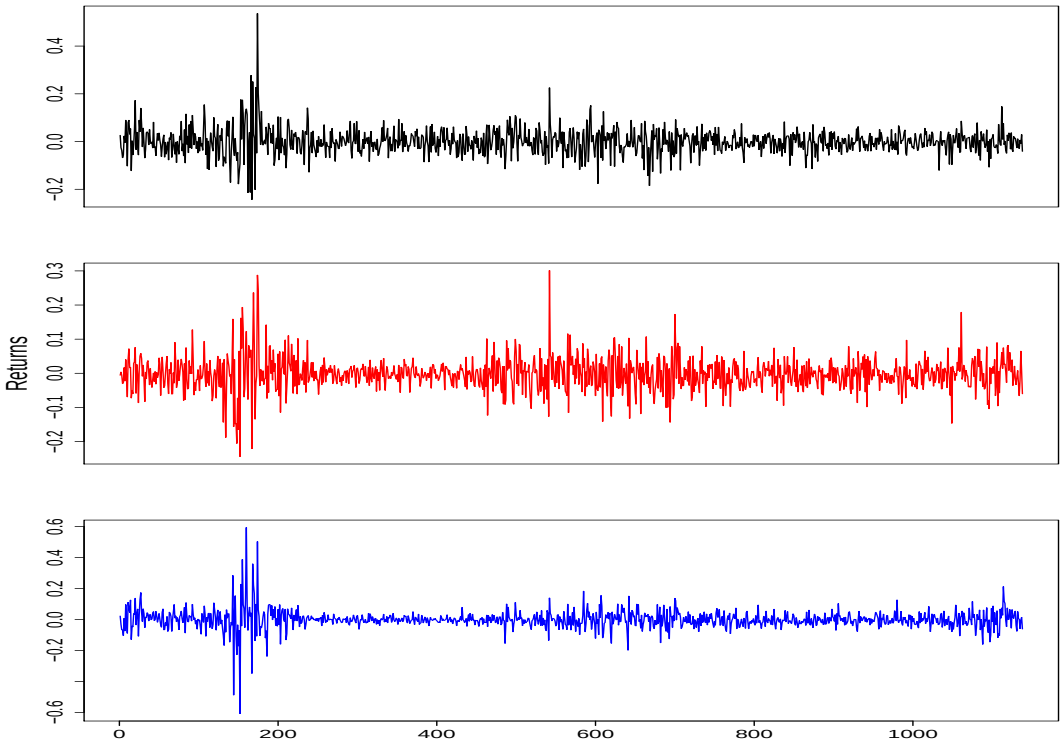

Figure 3.3: Log returns for DJIAindex from 04/1990 to 01/2012 of 3 companies

3.6 Conclusion

Alternating Minimization and Dual Path algorithms seem to be promising methods to solve the Group Fused Lasso problem. While the Dual Path algorithm is quite problem-specific, Alternating Minimization is a very generic method that can be applied to a broad class of problem, especially when there are some types of norm in the constraint since each update has closed form. Again we also want to note that our methods are coded in C++ which is usually faster than Matlab, the software in which our competing methods are coded in. Nevertheless, the reduction in computation time from using a powerful optimization method (AM) and from strategically merging parameters to simplify the problem is quite remarkable.

Chapter 4

Semiparametric Transformation Models for Complete and Survival data

4.1 Introduction

Linear regression and analysis of variance (ANOVA) are widely used for studying the relationship between a response and multiple covariates. One important requirement of such models is that the response variable needs to have a normal distribution with a constant variance given the covariates. However, this assumption may not hold in many real life examples, and this violation can be detected by some diagnostic methods based on residuals. A common remedy is to apply some transformation of the response variable so that the transformed response variable meets the normality requirement and then one can refit the model with the transformed response.

One popular transformation model for this strategy is the so-called Box-Cox transformation in Box and Cox, 1964, which takes the following form

$$y^{(\lambda)} = \begin{cases} \frac{y^{\lambda}-1}{\lambda} & \text{if} \quad \lambda \neq 0 \\ \log(y) & \text{if} \quad \lambda = 0, \end{cases}$$

where y is a original positive response and $y^{(\lambda)}$ is the transformed response. As a member of the power transformation family, the Box-Cox transformation is easy to understand and widely used.

There are also various extensions of Box-Cox transformation models that allow negative values for the response variable. For example, Box and Cox, 1964 proposed a modified transformation model with an additional shift parameter. Manly, 1976

proposed an exponential transformation that can transform skewed unimodal distributions to approximately normal ones. Schlesselman, 1971 proposes a modified version of the Box-Cox transformation that preserves scale invariance in likelihood procedures. Another notable and more recent contribution to this topic was made by Yeo and Johnson, 2000, which can handle negative values of the response and also enjoys many of the desirable features of the Box-Cox transformation.

Although these Box-Cox transformation models are widely adopted in the literature when the transformation of the response is needed, these models may not always provide adequate fit for some specific data sets. This is not surprising because these transformation models are parametric and can be overly restrictive in many cases. In fact, the Box-Cox transformation is not recommended if the estimated λ from the Box-Cox transformation has absolute value close to 2 or bigger.

In this chapter, we aim to propose a more flexible regression model that allows the response variable to have an unknown continuous distribution. The proposed model is based on the fact that if the response variable Y has a continuous distribution with a cumulative distribution function (CDF) F, then $F(Y)$ has a uniform distribution on $[0, 1]$. It is also known that $\Phi^{-1}(U)$ has a standard normal distribution if U has a uniform distribution on $[0, 1]$, where Φ^{-1} is the inverse CDF of the standard normal distribution. Thus, the transformation $\alpha = \Phi^{-1} \cdot F$ can convert the original response variable Y that has an arbitrary distribution to a normal random variable. Since both Φ^{-1} and F are increasing functions, the transformation α is also increasing.

This idea can be extended to regression settings by incorporating covariates. Let X denote the vector of covariates of length p. We propose the following transformation model

$$\alpha(Y) = X^T\beta + \epsilon, \tag{4.1}$$

where $\epsilon \sim N(0, 1)$, and β is the vector of regression coefficients, and α is an unspecified monotone transformation function. This model assumes that there is a linear

relationship between the transformed response and the covariates with an additional standard normal random error.

Model (4.1) is referred to as the semiparametric probit model and has been studied primarily in the literature of survival analysis as a member of a general class of linear transformation models. Instead of the standard normal distribution, taking the extreme value distribution and the standard logistic distribution for the distribution of ϵ leads to two most popular survival models: the proportional hazards (PH) model and the proportional odds (PO) model, respectively. The linear transformation models have been studied extensively, and existing work include Zhang et al., 2013, Cheng and Wang, 2011, Xu et al., 2019 for current status data, Zhang, 2009, Zeng et al., 2016 for general interval-censored data, and Li et al., 2010 for panel count data among others. Lin and Wang, 2010 developed an efficient Bayesian estimation approach for regression analysis of interval-censored data specifically under the probit model.

Although the semiparametric probit model has been essentially studied through the linear transformation models in survival literature, direct study of the semiparametric probit model for complete and right-censored data is rare and clearly understudied. In this chapter, we first propose a novel EM algorithm to fit the semiparametric probit model under complete data and then generalize it to right-censored data. Our method allows estimations the transformation function and the regression parameters simultaneously. The proposed approaches enjoy several appealing properties such as being easy to implement, robust to initial values, fast to converge and providing variance estimation in closed forms.

The remainder of this chapter is organized as follows. Section 2 gives an overview about our nonparametric transformation model. Sections 3 and 4 present the details of the proposed methods for complete data and right-censored data, respectively. Sections 5 provides details about model diagnostics. Section 6 shows the results from

comprehensive simulation studies and Section 7 provides two real life applications of our methods. Some concluding remarks are given in Section 8.

4.2 Properties of the semiparametric probit model

Model (4.1) is a semiparametric model since the transformation function is nonparametric and the regression part has a parametric form. Unlike the usual linear regression model, the variance of random error is fixed as 1 in the proposed model for identifiability since both the function α and β are unknown in this model. Also, there is no intercept in the linear regressor in the right side of model (1) for identifiablity since an intercept can be absorbed into the transformation.

Since α is nondecreasing, the cumulative distribution function (CDF) of Y given X under the probit model (4.1) is

$$\begin{aligned} F(y) &= P\Big(Y \le y|X\Big) = P\Big\{\alpha(Y) \le \alpha(y)|X\Big\} = P\Big\{X^T\beta + \epsilon \le \alpha(y)|X\Big\} \\ &= P\Big\{\epsilon \le \alpha(y) - X^T\beta|X\Big\} = \Phi\Big\{\alpha(y) - X^T\beta\Big\}, \end{aligned}$$

where Φ is the CDF of the standard normal distribution. Differentiating this CDF with respect to y yields the probability density function (pdf) of Y in the form of

$$f(y) = \phi\{\alpha(y) - X^T\beta\}\alpha'(y),$$

where $\phi(.)$ is the pdf of the standard normal distribution, and α' is the first derivative of α. The hazard function of Y takes the form

$$\lambda(y|X) = \alpha'(y)\phi\{\alpha(y) - X^T\beta\}[1 - \Phi\{\alpha(y) - X^T\beta\}]^{-1}.$$

This hazard function allows the hazard curves to cross for different predictor values, which is not maintained by the Proportional Hazards (PH) model. The transformation function α can be also interpreted as the inverse-probit transformed baseline CDF under the probit model, while β_l, the lth element of β, can be interpreted as

the expected change in the inverse-probit transformed failure time due to a 1 unit increase of the lth covariate while keeping all other covariates fixed.

Since the transformation function α is an unspecified nondecreasing function with infinite dimensions, estimation under the probit model is challenging. There does not seem to exist a partial likelihood, as in the case of the PH model for right-censored data, which allows one to estimate the regression parameters directly without the need to estimate the unknown baseline function.

Following the strategy of Lin and Wang (2011), we propose to model α with the monotone splines of Ramsay, 1988. Specifically, it is assumed that α can be written as a linear combination of monotone splines as

$$\alpha(\cdot) = \gamma_0 + \sum_{k=1}^{K} \gamma_k b_k(\cdot), \tag{4.2}$$

where the b_k's are the monotone spline (I-spline) basis functions and $\{\gamma_k\}_{k=0}^{K}$ are nonnegative unknown spline coefficients. All b_k's are constructed before the analysis and are non-decreasing from 0 to 1. The constraints of these spline coefficients ensure that α is nondecreasing.

The monotone splines are very flexible for approximating unknown non-decreasing functions with only a finite number of coefficients. This essentially coverts a semiparametric problem into a parametric problem, but does not requires any specific assumed form for the target function. Since the spline basis functions are completely determined once the knots and degree are specified, they do not need to be re-calculated during the estimation process.

An appealing byproduct of the use of the monotone spline representation of α in (4.2) is that the derivative of α takes the following explicit form,

$$\alpha'(\cdot) = \sum_{k=1}^{K} \gamma_k m_k(\cdot), \tag{4.3}$$

where $m_k(\cdot)$ is the first derivative of $b_k(\cdot)$ for $k = 1, \ldots, K$. These $m_k(\cdot)$'s are often referred to as M splines in the literature (Ramsay, 1988).

We consider complete data under the semiparametric probit model. Let $\{y_i, X_i\}, i = 1, 2, \ldots, N$ be an iid sample from the joint distribution of (Y, X). Treating covariates X_i as fixed, the observed likelihood under the probit model takes the form

$$\begin{aligned}\mathbf{L}_o &= \prod_{i=1}^{N} f(y_i|X_i) = \prod_{i=1}^{N} \phi\{\alpha(y_i) - X_i^T\beta\}\alpha'(y_i) \\ &\propto \prod_{i=1}^{N}\Bigg(\exp\Big[-\frac{1}{2}\Big\{\gamma_0 + \sum_{k=1}^{K}\gamma_k b_k(y_i) - X_i^T\beta\Big\}^2\Big]\Big\{\sum_{k=1}^{K}\gamma_k m_k(y_i)\Big\}\Bigg).\end{aligned}$$

The unknown parameters to be estimated are the regression parameters β and spline coefficients γ_k's.

Even though there are only a finite number of unknown parameters, direct optimization of this likelihood encounters many numerical problems such as non-convergence. To overcome this problem, we explore an EM algorithm by Dempster et al., 1977 below. First we consider the following data augmentation to get rid of the summation inside of the multiplicative terms of the likelihood. Introduce a latent multinomial vector $\mathbf{u_i} = (u_{i1}, ..., u_{iK}) \sim \mathcal{M}(1, \mathbf{p}_i)$, a multinomial distribution with the total count 1 and $\mathbf{p}_i = \left(\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}\right)$, for $i = 1, 2, ..., N$. The augmented likelihood treating all u_i's as missing data is

$$\mathbf{L}_c = \prod_{i=1}^{N}\Bigg(\exp\Big[-\frac{1}{2}\Big\{\gamma_0 + \sum_{k=1}^{K}\gamma_k b_k(y_i) - X_i^T\beta\Big\}^2\Big]\prod_{k=1}^{K}\Big\{\gamma_k m_k(y_i)\Big\}^{u_{ik}}\Bigg)$$

up to a multiplicative constant. Integrating u_i's out of $\mathbf{L}_c$ leads to the observed likelihood. This augmented likelihood will be used as the complete data likelihood for the derivation of our EM algorithm.

Let θ be the vector of all unknown parameters including β and γ_l's and let $\mathcal{D}$ denote the observed data. The E-step of the EM algorithm requires taking the conditional expectation of the logarithm of the complete likelihood with respect to the latent vectors u_i's given the observed data $\mathcal{D}$. This yields

$$Q(\theta, \theta^{(d)}) = \sum_{i=1}^{N}\Bigg[-\frac{1}{2}\Big\{\gamma_0 + \sum_{k=1}^{K}\gamma_k b_k(y_i) - X_i^T\beta\Big\}^2 + \sum_{k=1}^{K} h_{ik}^{(d)}\Big\{\log(\gamma_k)\Big\}\Bigg],$$

where $\theta^{(d)}$ is the current value of θ at the d-th step of the EM algorithm and

$$h_{ik}^{(d)} = \mathbb{E}(u_{ik}|\theta^{(d)}, \mathcal{D}) = \frac{\gamma_k^{(d)} m_k(y_i)}{\sum_{l=1}^{K} \gamma_l^{(d)} m_l(y_i)}$$

for $k = 1, \cdots, K$ and $i = 1, \cdots, n$.

The M-step of the EM algorithm is to maximize $Q(\theta, \theta^{(d)})$ with respect to θ. It turns out that there is no closed form expression for the global maximizer of $Q(\theta, \theta^{(d)})$. To overcome this problem, we update the unknown parameters sequentially in the order of β and γ_l for $l = 0, 1, \cdots, K$. That is, we first maximize the Q function with respect to β given $\gamma_l = \gamma_l^{(d)}$ for all l. This leads to a least squares solution

$$\beta^{(d+1)} = (X^T X)^{-1} X^T (\gamma_0^{(d)} 1_N + \mathbf{B}\gamma^{(d)}),$$

where 1_N is a vector of length N with all entries equal to 1, $\mathbf{B}$ is a $N \times K$ matrix with (i, k)-th entry equal to $b_k(y_i)$, and $\gamma = (\gamma_1, \gamma_2, ..., \gamma_K)^T$.

Then we maximize the Q function with respect to γ_0 given $\beta = \beta^{(d+1)}$ and $\gamma_l = \gamma_l^{(d)}$ for $l > 0$. This leads to closed expression for $\gamma_0^{(d+1)}$, which is

$$\gamma_0^{(d+1)} = \frac{1}{N} \sum_{i=1}^{N} \Big\{ X_i^T \beta^{(d+1)} - \sum_{k=1}^{K} \gamma_k^{(d)} b_k(y_i) \Big\}.$$

Now we present the details of updating the γ_k's sequentially. For any $k \geq 1$, suppose that the $\gamma_l^{(d+1)}$'s have been obtained for all $l < k$ and let $Q_k^{(d)}(\gamma_k)$ be the updated Q function with $\beta = \beta^{(d+1)}$ and $\gamma_l = \gamma_l^{(d+1)}$ for all $l < k$ and $\gamma_l = \gamma_l^{(d)}$ for all $l > k$. Differentiating $Q_k^{(d)}(\gamma_k)$ with respect to γ_k yields

$$\frac{\partial Q_k^{(d)}(\gamma_k)}{\partial \gamma_k} = \sum_{i=1}^{N} \Bigg[- b_k(y_i) \Big\{ \gamma_0^{(d+1)} + \sum_{l<k} \gamma_l^{(d+1)} b_l(y_i) + \sum_{l>k} \gamma_l^{(d)} b_l(y_i) - X_i^T \beta \Big\} + \frac{h_{ik}^{(d)}}{\gamma_k} \Bigg].$$

Setting $\partial Q_k^{(d)}(\gamma_k)/\partial \gamma_k = 0$ leads to the following quadratic equation for γ_k

$$\gamma_k^2 \sum_{i=1}^{N} b_k^2(y_i) + \gamma_k \sum_{i=1}^{N} \Bigg[b_k(y_i) \Big\{ \gamma_0 + \sum_{l \neq k} \gamma_l b_l(y_i) - X_i^T \beta \Big\} \Bigg] - \sum_{i=1}^{N} h_{ik}^{(d)} = 0. \tag{4.4}$$

Define $a_k = \sum_{i=1}^{N} b_k^2(y_i)$, $e_k^{(d)} = \sum_{i=1}^{N} h_{ik}^{(d)}$, and

$$c_k^{(d)} = \sum_{i=1}^{N} \Bigg[b_k(y_i) \Big\{ \gamma_0^{(d+1)} + \sum_{l<k} \gamma_l^{(d+1)} b_l(y_i) + \sum_{l>k} \gamma_l^{(d)} b_l(y_i) - X_i^T \beta^{(d+1)} \Big\} \Bigg].$$

Since $a_k > 0$ and $e_k^{(d)} > 0$, there is a unique positive solution to (4.4). This together with the fact $\partial^2 Q_k^{(d)}(\gamma_k)/\partial\gamma_k^2 = -\gamma_k^{-2} e_k^{(d)} < 0$ yields a unique maximizer $\gamma_k^{(d+1)}$ of $Q_k^{(d)}(\gamma_k)$ given by

$$\gamma_k^{(d+1)} = \frac{-c_k^{(d)} + \sqrt{c_k^{(d)2} + 4a_k e_k^{(d)}}}{2a_k}.$$

Thus, we update the $\gamma_k^{(d+1)}$'s sequentially for $k = 1, \cdots, K$. This algorithm is essentially an Expectation/Conditional Maximization (ECM) algorithm, which was developed by Meng and Rubin, 1993.

Algorithm 2 The ECM algorithm for complete data.

1: Let $d = 0$ and initialize $\beta^{(d)}$,$\gamma_0^{(d)}$, and $\gamma^{(d)}$.
2: Calculate $h_{ik}^{(d)} = \frac{\gamma_k^{(d)} m_k(y_i)}{\sum_{l=1}^K \gamma_l^{(d)} m_l(y_i)}$ for $i = 1, 2, ..., N$ and $k = 1, 2, ..., K$.
3: Calculate $\beta^{(d+1)} = (X^T X)^{-1} X^T (\gamma_0^{(d)} 1_N + \mathbf{B}\gamma^{(d)})$.
4: Calculate $\gamma_0^{(d+1)} = \sum_{i=1}^N \left\{ X_i^T \beta^{(d+1)} - \sum_{k=1}^K \gamma_k^{(d)} b_k(y_i) \right\} / N$.
5: Calculate $\gamma_k^{(d+1)} = \frac{-c_k^{(d)} + \sqrt{c_k^{(d)2} + 4a_k e_k^{(d)}}}{2a_k}$ sequentially for $k = 1, 2, ..., K$. Then let $d = d + 1$.
6: Repeat steps $2 \sim 5$ until convergence.

We summarize our proposed ECM algorithm in succinct manner in Algorithm 2. This algorithm is very appealing because all parameters are updated in simple closed-form at each iteration. Let $\hat{\beta}, \hat{\gamma_0}$ and $\hat{\gamma}$ denote the converged coefficients of the EM algorithm. The variance estimates of $\hat{\beta}, \hat{\gamma_0}$ and $\hat{\gamma}$ can be obtained in closed form by using the Louis method. We defer this part to next section, which provides a unified formula of the variance estimates for both cases of complete and right-censored data.

4.4 The proposed approach for right-censored data

In this section, we consider the right-censored failure data. Right-censored data are very common in many fields such as real life epidemiological, social behavioral, and medical studies. Subjects are usually under continuous monitoring and the failure time of interest would be exactly known if the failure occurred during the study.

However, many subjects do not experience the failure events throughout the study, and in this case we say their failure times are right censored at the study end time or at the censoring time. The censoring time can be the real end time of the study or the dropout time or death time for a specific subject. Right-censored data are a mixture of exactly observed observations and right-censored observations for the failure time.

Let T be the failure time of interest, C be a random censoring time, and X be a vector of covariates. For right-censored data, the observed time is the minimum of the failure time and the censoring time, i.e., $Y = min(T, C)$. Define $\Delta = I(T \leq C)$ be the censoring variable, taking the value 1 for exactly observed and 0 for right-censored observation. In this section, we assume that T follows a semiparametric probit model as in model (1), and the primary interests are to estimate the covariate effects β on the failure time T and the associated survival functions.

Let $\mathcal{D}_i = \{y_i, X_i, \delta_i\}_{i=1}^N$ be an iid copy of $\mathcal{D} = \{Y, X, \Delta\}$ for subjects $i = 1, \cdots, N$. Under the non-informative assumption that the failure time and the censoring time are conditionally independent given the covariates, the observed likelihood can be written as

$$\begin{aligned} L_{obs} &= \prod_{i=1}^{N} f(y_i|X_i)^{\delta_i}\{1 - F(y_i|X_i)\}^{1-\delta_i} \\ &= \prod_{i=1}^{N} \Big[\phi\Big\{\alpha(y_i) - X_i^T\beta\Big\}\alpha'(y_i)\Big]^{\delta_i}\Big[1 - \Phi\Big\{\alpha(y_i) - X_i^T\beta\Big\}\Big]^{1-\delta_i}, \end{aligned} \tag{4.5}$$

where α and α' are modeled with splines as in Section 2. As in Section 3, we seek an EM/ECM to estimate the parameters $\theta = (\beta', \gamma_0, \gamma')'$ jointly. Motivated by Lin and Wang, 2010, we introduce the normal latent variables

$$Z_i = N(\alpha(y_i) - X_i^T\beta, 1), \qquad i = 1, 2, \ldots, N,$$

with constraints $Z_i < 0$ for all censored subjects with $\delta_i = 0$ based on the fact that

$$1 - F(y_i) = 1 - \Phi\{\alpha(y_i) - X_i^T\beta\} = \int_{-\infty}^{0} \phi\{z_i - \alpha(y_i) + X_i^T\beta\}dz_i.$$

For the purpose of notational convenience, we define $Z_i = 0$ for all exactly observed subjects with $\delta_i = 1$. Conditional on the latent variables Z_i's, the augmented data likelihood takes the form

$$\mathbf{L}_{a1} = \prod_{i=1}^{N} \phi\Big\{Z_i - \alpha(y_i) + X_i^T\beta\Big\}\{\alpha'(y_i)\}^{\delta_i}$$

with the constraints $Z_i = 0$ if $\delta_i = 1$ and $Z_i < 0$ if $\delta_i = 0$ for $i = 1, \cdots, N$. Integrating this augmented likelihood with respect to all Z_i's leads back to the observed likelihood (4).

We use the same strategy to handle the summation of M splines in α', and introduce multinomial latent vectors u_i's with $u_i = (u_{i1}, \cdots, u_{iK}) \sim \mathcal{M}(1, \mathbf{p}_i)$, a multinomial distribution with the total count 1 and $\mathbf{p}_i = \Big(\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}\Big)$, for all censored subjects with $\delta_i = 0$. The new augmented likelihood is now

$$\mathbf{L}_c = \prod_{i=1}^{N} \exp\Big[-\frac{1}{2}\Big\{Z_i - \gamma_0 - \sum_{k=1}^{K}\gamma_k b_k(y_i) + X_i^T\beta\Big\}^2\Big] \prod_{k=1}^{K}\Big\{\gamma_k m_k(y_i)\Big\}^{\delta_i u_{ik}},$$

up to a multiplicative constant. This augmented likelihood will be used as the complete data likelihood for the derivation of our ECM algorithm.

The conditional expectation of the log of the complete likelihood with respect to the latent variables Z_i's and u_{ik}'s given the observed data $\mathcal{D}$ and the current parameters $\theta^{(d)}$ takes the form

$$\begin{aligned} Q(\theta, \theta^{(d)}) = -\frac{1}{2}\sum_{i=1}^{N}\Big\{\gamma_0 + \sum_{k=1}^{K}\gamma_k b_k(y_i) - X_i^T\beta\Big\}^2 + \sum_{i=1}^{N}\sum_{k=1}^{K} h_{ik}^{(d)} \log(\gamma_k)\delta_i \\ + \sum_{i=1}^{N} E(Z_i|\mathcal{D}_i, \theta^{(d)})\Big\{\gamma_0 + \sum_{k=1}^{K}\gamma_k b_k(y_i) - X_i^T\beta\Big\}, \end{aligned}$$

up to an additive constant, where $h_{ik}^{(d)} = \mathbb{E}(u_{ik}|\theta^{(d)}, \mathcal{D})$ and $E(Z_i|\mathcal{D}, \theta^{(d)})$ are the conditional expectations of u_{ik} and Z_i, respectively. All these conditional expectations have explicit forms, such that

$$h_{ik}^{(d)} = \frac{\gamma_k^{(d)} m_k(y_i)}{\sum_{l=1}^{K}\gamma_l^{(d)} m_l(y_i)},$$

and

$$E(Z_i|\mathcal{D},\theta^{(d)}) = \begin{cases} \mu_i^{(d)} - [\Phi\{\mu_i^{(d)}\}]^{-1}\phi\{\mu_i^{(d)}\}, & \text{if } \delta_i = 0 \\ 0, & \text{if } \delta_i = 1, \end{cases}$$

where $\mu_i^{(d)} = \gamma_0^{(d)} + \sum_{k=1}^{K} \gamma_k^{(d)} b_k(y_i) - X_i^T\beta^{(d)}$, for $k = 1, \cdots, K$ and $i = 1, \cdots, N$. Similar to the ECM for the case of complete data in Section 3, we seek to optimize the Q function with respect to β, γ_0, and all the elements of γ sequentially and based on the newly updated values for the other parameters. Similar to the derivation process in Section 3, the strategy leads to the following updating of all parameters in closed form,

$$\beta^{(d+1)} = \Big\{\sum_{i=1}^{N} X_i X_i^T\Big\}^{-1} \sum_{i=1}^{N} X_i \Big\{\gamma_0^{(d)} + \sum_{k=1}^{K} \gamma_k^{(d)} b_k(y_i) - E(Z_i|\mathcal{D}_i,\theta^{(d)})\Big\},$$

$$\gamma_0^{(d+1)} = \frac{1}{N}\sum_{i=1}^{N}\left[X_i^T\beta^{(d+1)} + E(Z_i|\mathcal{D}_i,\theta^{(d)}) - \Big\{\sum_{k=1}^{K}\gamma_k^{(d)} b_k(y_i)\Big\}\right],$$

and

$$\gamma_k^{(d+1)} = \frac{-c_k^{(d)} + \sqrt{c_k^{(d)2} + 4a_k e_k^{(d)}}}{2a_k},$$

where $a_k = \sum_{i=1}^{N} b_k^2(y_i)$, $e_k^{(d)} = \sum_{i=1}^{N} \delta_i h_{ik}^{(d)}$, and

$$c_k^{(d)} = \sum_{i=1}^{N} b_k(y_i)\Big\{\gamma_0^{(d+1)} + \sum_{l<k}\gamma_l^{(d+1)} b_l(y_i) + \sum_{l>k}\gamma_l^{(d)} b_l(y_i) - X_i^T\beta^{(d+1)} - E(Z_i|\mathcal{D}_i,\theta^{(d)})\Big\}$$

for $k = 1, \cdots, K$.

Let $\hat{\theta} = (\hat{\beta}', \hat{\gamma}_0, \hat{\gamma}')'$ denote the converged values of the ECM sequence of $\theta^{(d)}$'s. The variance estimates $var(\hat{\theta})$ can be calculated as the inverse of the observed information matrix $\{\mathbf{I}(\hat{\theta})\}^{-1}$, where $\mathbf{I}(\theta)$ can be obtained using the missing data principle (Louis, 1982),

$$\mathbf{I}(\theta) = -\frac{\partial^2 Q(\theta,\hat{\theta})}{\partial\theta\partial\theta'} - var\left(\frac{\partial log L_c}{\partial\theta}\right).$$

The detailed formulas are displayed in the Appendix. It is clear that all quantities involved have closed-form expressions, which lead to exact and fast calculations. Note that these formulas also apply for the variance estimates for the case of complete data, which are a special case of right-censored data taking δ_i to be 1 for all subjects.

Algorithm 3 The ECM algorithm for right-censored data.

1: Let $d = 0$ and initialize $\beta^{(d)}$,$\gamma_0^{(d)}$ and $\gamma^{(d)}$.
2: Calculate $h_{ik}^{(d)} = \frac{\gamma_k^{(d)} m_k(y_i)}{\sum_{l=1}^{K} \gamma_l^{(d)} m_l(y_i)}$ for $i = 1, 2, ..., N$ and $k = 1, 2, ..., K$.
3: Calculate $\beta^{(d+1)} = \Big\{ \sum_{i=1}^{N} X_i X_i^T \Big\}^{-1} \sum_{i=1}^{N} X_i \Big\{ \gamma_0^{(d)} + \sum_{k=1}^{K} \gamma_k^{(d)} b_k(y_i) - E(Z_i|\mathcal{D}_i, \theta^{(d)}) \Big\}$.
4: Calculate $\gamma_0^{(d+1)} = \sum_{i=1}^{N} \Big[X_i^T \beta^{(d+1)} + E(Z_i|\mathcal{D}_i, \theta^{(d)}) - \Big\{ \sum_{k=1}^{K} \gamma_k^{(d)} b_k(y_i) \Big\} \Big] / N$.
5: Calculate $\gamma_k^{(d+1)} = \frac{-c_k^{(d)} + \sqrt{c_k^{(d)2} + 4 a_k e_k^{(d)}}}{2 a_k}$ for $k = 1, 2, ..., K$ and let $d = d + 1$.
6: Repeat steps $2 \sim 5$ until all convergence.

4.5 Model diagnosis

4.5.1 For complete data

Model diagnosis is important to address whether the analysis based on a specific model is valid for a specific data set. To evaluate the validity of the probit model assumption for complete data, first we define the following residuals

$$r_i = \widehat{\alpha}(y_i) - X_i^T \widehat{\beta}, \tag{4.6}$$

for $i = 1, \cdots, n$. If the probit model assumption is valid, these residuals r_i's can be considered as a random sample from a standard normal distribution. Based on this fact, one can view quantile-quantile (Q-Q) plot by Gnanadesikan and Wilk, 1968 to check the validity of the model assumption. One can conclude a violation of the probit model assumption if there is a serious deviation of the observations from a straight line in the Q-Q plot. Alternatively one can calculate the empirical CDF $\widehat{F}_n(t) = \frac{1}{n} \sum_{i=1}^{n} 1_{(r_i \le t)}$ based on the sample of residuals r_i's and compare $\widehat{F}_n$ with the standard normal CDF Φ. The probit model assumption is considered problematic if there is a clear difference between $\widehat{F}_n$ and Φ. In addition to the visual plots, formal tests such as Shapiro-Wilk test by Shapiro and Wilk, 1965 can also be used to check whether the residuals follow a normal distribution. There is a built-in function in R by Team, 2013 called "shapiro.test" that performs the Shapiro-Wilk test.

4.5.2 For censored data

For censored data, we can define scaled residuals r_i's as in (4.6). Notice that r_i is right-censored observation of ϵ_i for subject i with $\delta_i = 0$. Thus, the r_i's form a right-censored random sample from a standard normal distribution if the probit model assumption is valid. Thus, one can obtain the Kaplan-Meier (Kaplan and Meier, 1958) estimate $S_n(t)$ of the true survival function based on the sample (r_i, δ_i)'s and compare it with the survival function $1 - \Phi(t)$ under the probit model. The probit model assumption is considered problematic if there is a clear difference between the empirical survival function $\widehat{S}_n$ and the true survival function $1-\Phi$. Formal tests such as the one sample log-rank test by Breslow, 1975 can be used to compare $\widehat{S}_n$ and $1 - \Phi$.

4.6 Simulation results

Extensive simulation studies were conducted to evaluate the performance of the proposed methods in different settings. The failure time T was generated from the following probit model

$$F(t|x) = \Phi(\alpha(t) + \beta_1 X_1 + \beta_2 X_2),$$

with two continuous and discrete covariates $X_1 \sim U(0,1)$ and $X_2 \sim Bernoulli(0.5)$. Three different functions were considered for the true function $\alpha(t)$, with two Box-Cox transformations $\log(t)$ and $(t^{0.1} - 1)/0.1$ and a non-Box-Cox transformation $t^3 + t$. The true values of $\beta_1, \beta_2)$ were taken to be $(0,0)$, $(0,1)$, $(1,0)$, and $(1,-1)\}$. The censoring time C was generated from an exponential distribution $\exp(\lambda)$ with mean parameter λ. Four scenarios were considered with different right-censoring rates of 0%, roughly 20%, 40% and 60% by taking appropriate values of λ for different parameter configurations. The 0% censoring rate corresponds to the case of complete

data. For each setup, 500 data sets were generated with 200 observations for each data set.

To implement our methods, the monotone splines were specified with degree 2 and 15 knots for all the simulation. The knots were set to based on the quantiles of the observed response values. Tables 4.1 $\sim$ 4.3 present the characteristics of the estimates of the regression coefficients β_1 and β_2 in terms of bias, the difference between the average of 500 point estimators and the true value, SSD, the standard deviation of the 500 point estimators, ASE, the average of the standard errors, and CP95, the 95% coverage probability for each parameter estimate across all simulation setups. The results in Tables 4.1 $\sim$ 4.3 suggest that our methods perform very well with very small bias, close values between SSD and ASE, and CP95 being close to the nominal value 0.95 for each parameter estimate across all parameter configurations in the simulation. It is observed that the estimation performance diminishes with larger bias and larger variance estimate for each parameter estimation in general as the censoring rate increases. This is expected as the information about the failure time contained in the data decreases as the censoring rate increases.

In order to assess the estimation performance of our methods on the baseline CDF, we consider local mean squared errors (MSEs) on a set of pre-specified grid points. For any grade point t, the local MSE of the baseline CDF estimate $\hat{F}_0(t)$ is defined as

$$\text{MSE}(\hat{F}_0(t)) = \frac{1}{500}\sum_{j=1}^{500}(F_0(t) - \hat{F}_0^{(j)}(t))^2,$$

where $\hat{F}_0^{(j)}(t)$ is the estimator of $F_0(t)$ based on the j-th dataset. The mean MSE of $\hat{F}_0$, defined as the average of these local MSEs over the set of grid points, provides a global measure of MSE of $\hat{F}_0$ and can be used to evaluate how accurate the estimation of F_0 is. Table 4.4 present the results of MSEs for $\hat{F}_0$ as well as for $\hat{\beta}_1$ and $\hat{\beta}_2$. The small values of mean MSEs of $\hat{F}_0$ indicate that the proposed methods can estimate the baseline CDF accurately.

For the comparison purposes, the Box-Cox transformation method was applied to the the complete data, i.e., the simulation data with 0% right censoring rate. Table 4.5 summarizes the estimation results of the regression parameter estimates in terms of bias, SSD, ASE, and CP95, and Table 4.6 summarizes the results of mean squared errors of $\hat{\beta}_1$, $\hat{\beta}_2$, and $\hat{F}_0$. The results in these two tables suggest that the Box-Cox method works well when the true transformation $\alpha(t) = \log(t)$ and $\alpha(t) = (t^{0.1} - 1)/0.1$, both falling with the Box-Cox transformation family, but fails to work when the true transformation is $\alpha(t) = t^3 + t$, a transformation not within the the Box-Cox transformation family. In contrast, the proposed method works well for all these cases as seen in Tables 4.1∼4.4.

Table 4.1: Simulation results from the proposed methods based on 500 simulated data sets with sample size 200 when the true transformation function is $\alpha(t) = \log(t)$. CR stands for censored rate. Bias is the difference between the average of 500 point estimators and the true value. SSD is the standard deviation of the 500 point estimators. ASE is the average of the standard errors of each point estimator. CP95, the coverage probability, is calculated as the proportion of the 95% confidence intervals that covers the true value of the coefficient.

$\alpha(t) = \log(t)$		Results for $\beta 1$				Results for $\beta 2$			
CR	$(\beta 1, \beta 2)$	Bias	SSD	ASE	CP95	Bias	SSD	ASE	CP95
0%	$(0, 0)$	0.001	0.252	0.248	0.942	-0.002	0.146	0.143	0.946
	$(0, 1)$	0.002	0.251	0.248	0.942	-0.015	0.155	0.152	0.946
	$(1, 0)$	-0.013	0.260	0.253	0.944	-0.003	0.145	0.143	0.954
	$(1, -1)$	-0.017	0.253	0.253	0.942	-0.016	0.155	0.152	0.942
20%	$(0, 0)$	-0.004	0.267	0.249	0.941	-0.012	0.145	0.144	0.946
	$(0, 1)$	0.010	0.284	0.256	0.952	-0.017	0.163	0.170	0.938
	$(1, 0)$	0.018	0.278	0.282	0.947	0.011	0.165	0.178	0.944
	$(1, -1)$	0.011	0.280	0.272	0.934	-0.017	0.161	0.163	0.942
40%	$(0, 0)$	0.012	0.317	0.289	0.959	-0.019	0.171	0.178	0.939
	$(0, 1)$	-0.025	0.295	0.297	0.941	0.023	0.181	0.178	0.941
	$(1, 0)$	-0.029	0.327	0.301	0.932	0.027	0.184	0.181	0.961
	$(1, -1)$	-0.029	0.323	0.291	0.942	-0.029	0.180	0.178	0.938
60%	$(0, 0)$	0.028	0.361	0.341	0.928	0.029	0.207	0.209	0.959
	$(0, 1)$	0.032	0.358	0.334	0.965	0.031	0.213	0.211	0.967
	$(1, 0)$	0.035	0.371	0.341	0.967	-0.037	0.218	0.201	0.931
	$(1, -1)$	-0.031	0.362	0.334	0.926	-0.029	0.211	0.198	0.965

Table 4.2: Simulation results from the proposed methods based on 500 simulated data sets with sample size 200 when the true transformation function is $\alpha(t) = (t^{0.1} - 1)/0.1$. CR stands for censored rate. Bias is the difference between the average of 500 point estimators and the true value. SSD is the standard deviation of the 500 point estimators. ASE is the average of the standard errors of each point estimator. CP95, the coverage probability, is calculated as the proportion of the 95% confidence intervals that covers the true value of the coefficient.

$\alpha(t) = (t^{0.1} - 1)/0.1$		Results for $\beta 1$				Results for $\beta 2$			
CR	$(\beta 1, \beta 2)$	Bias	SSD	ASE	CP95	Bias	SSD	ASE	CP95
0%	$(0, 0)$	0.001	0.252	0.248	0.942	-0.002	0.146	0.143	0.946
	$(0, 1)$	0.002	0.251	0.248	0.942	-0.016	0.155	0.152	0.948
	$(1, 0)$	-0.013	0.260	0.253	0.944	-0.003	0.146	0.143	0.952
	$(1, -1)$	-0.017	0.253	0.253	0.944	-0.016	0.155	0.152	0.942
20%	$(0, 0)$	-0.004	0.267	0.249	0.946	-0.012	0.145	0.145	0.946
	$(0, 1)$	0.012	0.284	0.257	0.955	-0.017	0.163	0.172	0.936
	$(1, 0)$	0.018	0.270	0.282	0.946	0.011	0.157	0.178	0.947
	$(1, -1)$	0.011	0.280	0.275	0.934	-0.018	0.161	0.163	0.941
40%	$(0, 0)$	0.012	0.318	0.287	0.959	-0.019	0.171	0.178	0.939
	$(0, 1)$	-0.028	0.297	0.297	0.944	0.023	0.183	0.178	0.941
	$(1, 0)$	0.029	0.317	0.303	0.932	-0.027	0.185	0.181	0.961
	$(1, -1)$	-0.029	0.323	0.291	0.944	0.029	0.182	0.176	0.938
60%	$(0, 0)$	0.028	0.361	0.341	0.928	0.029	0.207	0.209	0.959
	$(0, 1)$	0.032	0.361	0.336	0.965	0.031	0.216	0.214	0.966
	$(1, 0)$	0.035	0.371	0.338	0.966	-0.038	0.216	0.201	0.934
	$(1, -1)$	-0.031	0.362	0.334	0.928	-0.029	0.211	0.198	0.966

Table 4.3: Simulation results from the proposed methods based on 500 simulated data sets with sample size 200 when the true transformation function is $\alpha(t) = t^3 + t$. CR stands for censored rate. Bias is the difference between the average of 500 point estimators and the true value. SSD is the standard deviation of the 500 point estimators. ASE is the average of the standard errors of each point estimator. CP95, the coverage probability, is calculated as the proportion of the 95% confidence intervals that covers the true value of the coefficient.

$\alpha(t) = t^3 + t$		Results for $\beta 1$				Results for $\beta 2$			
CR	$(\beta 1, \beta 2)$	Bias	SSD	ASE	CP95	Bias	SSD	ASE	CP95
0%	$(0, 0)$	0.001	0.252	0.248	0.946	-0.002	0.146	0.143	0.944
	$(0, 1)$	0.002	0.251	0.248	0.942	-0.016	0.155	0.152	0.948
	$(1, 0)$	-0.013	0.260	0.253	0.944	-0.003	0.145	0.143	0.950
	$(1, -1)$	-0.017	0.253	0.253	0.946	-0.016	0.155	0.152	0.944
20%	$(0, 0)$	0.004	0.267	0.249	0.946	-0.012	0.145	0.144	0.946
	$(0, 1)$	0.012	0.284	0.257	0.952	-0.017	0.163	0.170	0.936
	$(1, 0)$	0.018	0.270	0.282	0.946	0.011	0.167	0.178	0.947
	$(1, -1)$	0.011	0.280	0.275	0.934	-0.018	0.161	0.163	0.944
40%	$(0, 0)$	0.012	0.318	0.289	0.959	-0.019	0.171	0.178	0.939
	$(0, 1)$	-0.028	0.295	0.297	0.941	-0.023	0.183	0.178	0.941
	$(1, 0)$	0.029	0.317	0.303	0.932	-0.027	0.184	0.181	0.961
	$(1, -1)$	-0.029	0.323	0.291	0.944	-0.029	0.182	0.178	0.938
60%	$(0, 0)$	0.028	0.361	0.341	0.928	0.029	0.207	0.209	0.959
	$(0, 1)$	-0.032	0.361	0.334	0.965	0.031	0.216	0.214	0.967
	$(1, 0)$	0.035	0.371	0.342	0.966	-0.038	0.218	0.201	0.934
	$(1, -1)$	-0.031	0.362	0.334	0.926	-0.029	0.211	0.198	0.966

Table 4.4: Mean Squared Errors of $\hat{\beta}_1, \hat{\beta}_2, \hat{F}_0$ for our proposed methods with $\alpha(t) = \log(t)$ and $\alpha(t) = t^3 + t$. Here $\overline{\text{MSE}}(\hat{F}_0)$ is the average of the local mean squared errors of $\hat{F}(t)$ over the a set of grid points.

		$\alpha(t) = \log(t)$			$\alpha(t) = t^3 + t$		
CR	(β_1, β_2)	MSE($\hat{\beta}_1$)	MSE($\hat{\beta}_2$)	$\overline{MSE}(\hat{F}_0)$	MSE(β_1)	MSE(β_2)	$\overline{MSE}(\hat{F}_0)$
0%	$(0, 0)$	0.0147	0.0457	0.0009	0.0194	0.0638	0.0010
	$(0, 1)$	0.0183	0.0632	0.0011	0.0327	0.0917	0.0009
	$(1, 0)$	0.0247	0.0672	0.0009	0.0284	0.0842	0.0012
	$(1, -1)$	0.0267	0.0798	0.0009	0.0281	0.0831	0.0009
20%	$(0, 0)$	0.0274	0.0683	0.0009	0.0318	0.0901	0.0011
	$(0, 1)$	0.0389	0.0801	0.0009	0.0356	0.0867	0.0009
	$(1, 0)$	0.0295	0.0783	0.0011	0.0295	0.0912	0.0009
	$(1, -1)$	0.0232	0.0925	0.0010	0.0328	0.0794	0.0010
40%	$(0, 0)$	0.0386	0.1045	0.0012	0.0401	0.0873	0.0011
	$(0, 1)$	0.0396	0.0948	0.0009	0.0381	0.0920	0.0010
	$(1, 0)$	0.0289	0.0892	0.0011	0.0314	0.1104	0.0009
	$(1, -1)$	0.0324	0.1148	0.0009	0.0334	0.1008	0.0009
60%	$(0, 0)$	0.0294	0.1138	0.0012	0.0392	0.1193	0.0011
	$(0, 1)$	0.0327	0.0917	0.0013	0.0411	0.1301	0.0013
	$(1, 0)$	0.0384	0.1242	0.0012	0.0312	0.0109	0.0012
	$(1, -1)$	0.0436	0.1288	0.0011	0.0438	0.1246	0.0011

Table 4.5: Simulation results from the Box-Cox transformation method based on 500 simulated data sets with sample size 200 for three true transformation function α with no censoring. Bias is the difference between the average of 500 point estimators and the true value. SSD is the standard deviation of the 500 point estimators. ASE is the average of the standard errors of each point estimator. CP95, the coverage probability, is calculated as the proportion of the 95% confidence intervals that covers the true value of the coefficient.

		Results for $\beta 1$				Results for $\beta 2$			
$\alpha(t)$	$(\beta 1, \beta 2)$	Bias	SSD	ASE	CP95	Bias	SSD	ASE	CP95
$log(t)$	$(0,0)$	-0.005	0.271	0.244	0.962	-0.002	0.160	0.143	0.946
	$(0,1)$	-0.004	0.271	0.244	0.961	0.001	0.166	0.142	0.902
	$(1,0)$	-0.005	0.275	0.244	0.964	0.002	0.160	0.143	0.944
	$(1,-1)$	-0.004	0.271	0.244	0.962	0.000	0.160	0.142	0.942
$\frac{t^{0.1}-1}{0.1}$	$(0,0)$	0.004	0.267	0.249	0.946	-0.012	0.145	0.145	0.946
	$(0,1)$	-0.004	0.271	0.244	0.955	0.001	0.166	0.142	0.906
	$(1,0)$	-0.005	0.274	0.244	0.966	0.001	0.163	0.142	0.947
	$(1,-1)$	-0.004	0.271	0.244	0.964	0.000	0.160	0.163	0.941
t^3+t	$(0,0)$	-0.001	0.008	0.008	0.940	-0.000	0.005	0.004	0.960
	$(0,1)$	-0.001	0.008	0.008	0.944	-0.967	0.007	0.005	0
	$(1,0)$	-0.968	0.007	0.008	0	0.000	0.005	0.005	0.941
	$(1,-1)$	-0.968	0.011	0.008	0	0.969	0.008	0.004	0

Table 4.6: Mean Squared Errors of $\hat{\beta}_1, \hat{\beta}_2, \hat{F}_0$ for Box-Cox transformation method with 3 different $\alpha(t)$ with no censoring.

$\alpha(t)$	True (β_1, β_2)	MSE$(\hat{\beta}_1)$	MSE$(\hat{\beta}_2)$	$\overline{MSE}(\hat{F}_0)$
$\log(t)$	$(0,0)$	0.0722	0.0250	0.0000
	$(0,1)$	0.0722	0.0271	0.0000
	$(1,0)$	0.0743	0.0251	0.0000
	$(1,-1)$	0.0721	0.0251	0.0000
$\frac{t^{0.1}-1}{0.1}$	$(0,0)$	0.0722	0.0251	0.0001
	$(0,1)$	0.0722	0.0271	0.0001
	$(1,0)$	0.0738	0.0251	0.0001
	$(1,-1)$	0.0721	0.0251	0.0001
t^3+t	$(0,0)$	0.0000	0.0000	0.1151
	$(0,1)$	0.0000	0.9348	0.1464
	$(1,0)$	0.9371	0.0000	0.1430
	$(1,-1)$	0.9384	0.9393	0.1254

4.7 Real-data applications

4.7.1 Boston housing data analysis

The Boston housing dataset by Harrison Jr and Rubinfeld, 1978 has been widely used to as a benchmark to compare different machine learning methods and regression models. The data were collected in 1978 by the U.S Census Service, and each of the 506 entries represented the aggregated data about 14 features for homes from various suburbs in Boston, Massachusetts. The response variable is the median value of owner-occupied homes in thousands of dollars. This is a complete dataset to illustrate our proposed method.

We apply the proposed method to this data set by taking degree 2 and using different numbers of interior knots based on the quantiles for the monotone splines. Akaike information criterion (AIC) by Akaike, 1998 is used for model selection. Table 4.7 reports the calculated AIC from the proposed method when using different number of knots. As seen in Table 4.7, the model with 15 interior knots produces the smallest AIC value and is chosen as our final model.

For comparison, we also fit a linear regression model for the Box-Cox transformed response. It is worth noting that the probit model has a standard normal random error and the linear model based on Box-Cox transformation assumes a normal random error with an unknown variance. To make the estimated coefficient comparable for the two methods, we scaled the coefficient estimators from Box-Cox transformation by $1/\hat{\sigma}$. Table 4.8 shows the estimation results of the covariate effects from the two competing methods. Our propose method has no intercept since it is absorbed in the transformation part. As seen in Table 4.8, the two models produce very similar estimation results for all these covariate effects, in terms of the point estimates and their 95% confidence intervals.

To further examine the model fit, we investigated the residuals from the two competing methods. The residuls from the proposed probit model were calculated as in equation 4.6 of Section 5.1, and the residuals of the Box-Cox transformation model were the scaled residuals, the usual residuals from the usual linear regression model multiplied by the estimated standard deviation $\hat{\sigma}$ of the random errors, in order to provide a fair comparison. From Figure 2, the residual plot from the proposed probit model show a good of fit with essentially no particular pattern and with equal variance across the predicted values, while the residual plot from the Box-Cox transformation model shows a lack of fit with a decreasing variance across the predicted values. Figure 3 shows the Quantile-Quantile (Q-Q) plot for each set of residuals from the two methods. As seen from Figure 3, the Q-Q plot based on the Box-Cox transformation shows a serious violation of the normality assumption, while the Q-Q plot from the probit model does not suggest so. To provide more formal normality test results, Shapiro-Wilk tests of normality were conducted on the two sets of residuals using R function "shapiro.test". These tests report p-values of 0.2283 and 0 for the probit model and the Box-Cox transformation model, respectively. These results again suggest that the residuals from the probit model pass the normality test while the residuals from the Box-Cox tranformation model do not. Altogether these results of residual analysis suggest that the probit model provides a good fit for this data set while the regression model based on the Box-Cox tranformation has a lack of fit.

Table 4.7: The calculated Kaike Information Criterion (AIC) values from the probit models with different numbers of interior knots for the Boston Housing data analysis.

#	10	12	14	15	16	18	20
AIC	1833.085	1830.918	1824.239	1813.388	1816.318	1820.449	1822.982

Table 4.8: Coefficient estimates from the probit model and Box-Cox transformation model.

	Proposed model			Box-Cox transformation		
	$\hat{\beta}$	Lower	Upper	$\hat{\beta}$	Lower	Upper
(Intercept)	-	-	-	19.263	17.151	21.375
xcrim	-0.043	-0.048	-0.037	-0.051	-0.066	-0.039
xzn	0.006	-0.020	0.032	0.008	0.000	0.012
xindus	0.013	-0.346	0.371	0.012	-0.012	0.039
xchas	0.582	-1.028	2.192	0.541	0.187	0.898
xnox	-4.494	-4.679	-4.309	-4.130	-5.713	-2.547
xrm	0.503	0.498	0.509	0.525	0.350	0.696
xage	-0.004	-0.089	0.082	0.000	-0.004	0.008
xdis	-0.289	-0.317	-0.261	-0.268	-0.350	-0.187
xrad	0.080	0.079	0.082	0.074	0.047	0.101
xtax	-0.004	-0.060	0.053	-0.004	-0.004	0.000
xptratio	-0.222	-0.224	-0.221	-0.206	-0.261	-0.152
xblack	0.003	-0.021	0.026	0.004	0.000	0.004
xlstat	-0.159	-0.173	-0.144	-0.152	-0.171	-0.128

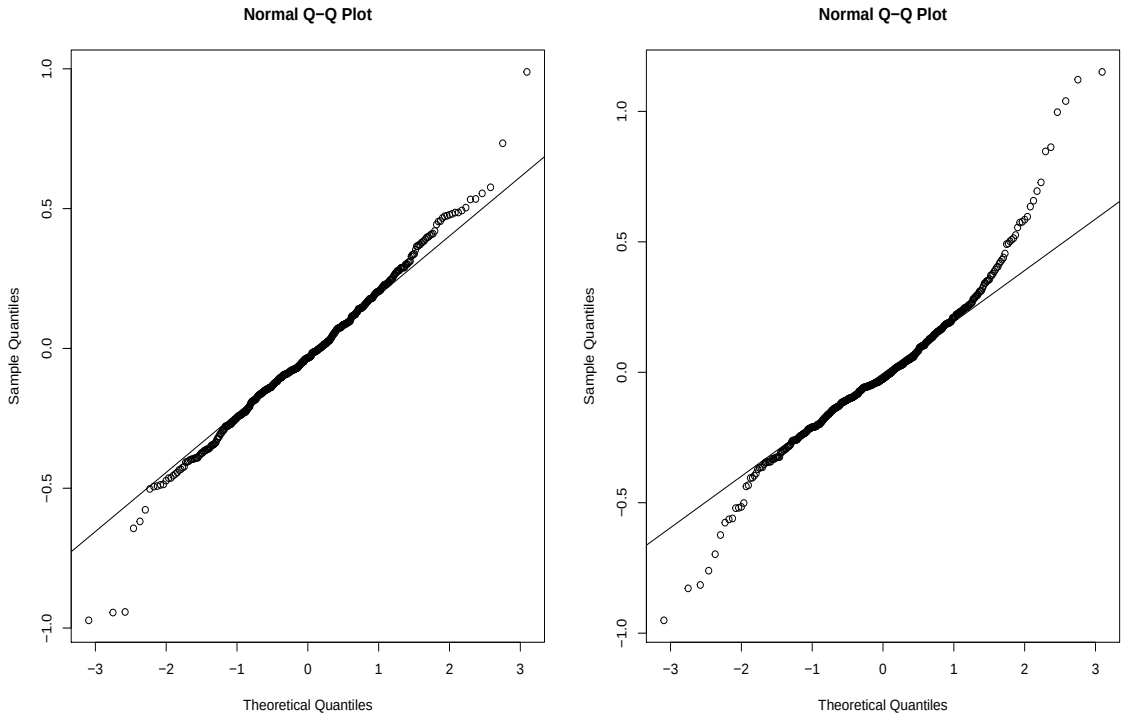

Figure 4.2: Quantile-Quantile plots of the residuals from the proposed probit model (left) and the Box-Cox transformation model (right) for Boston Housing data analysis.

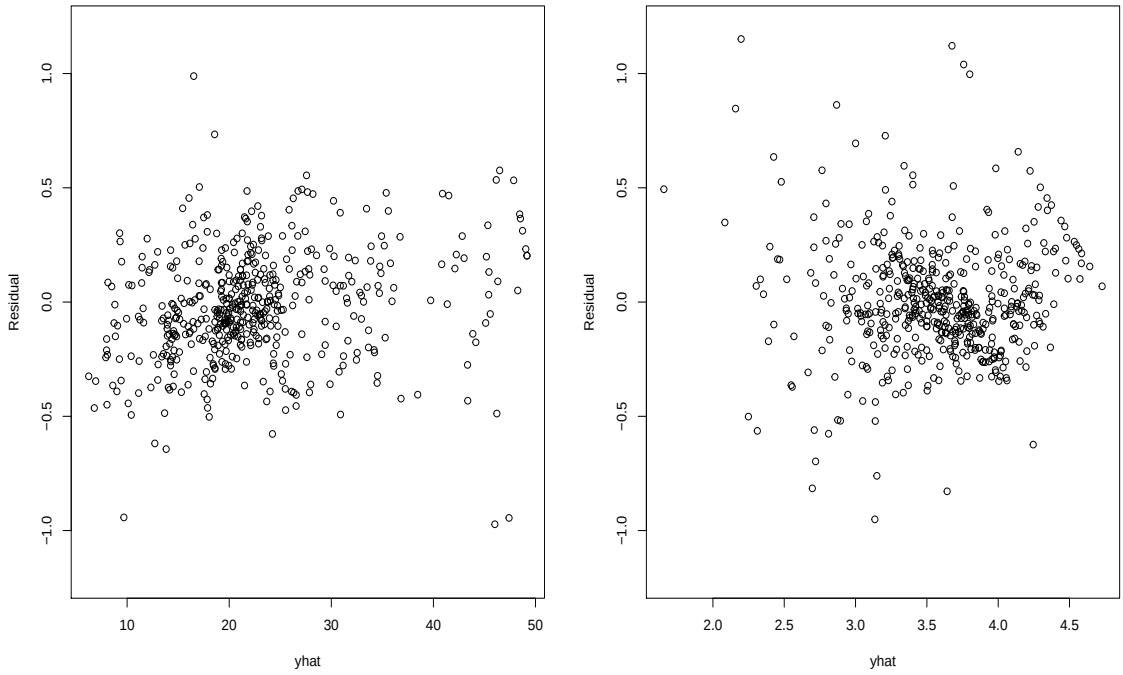

Figure 4.1: Residual plots from the proposed probit model (left) and the Box-Cox transformation model (right) for Boston Housing data analysis.

4.7.2 Prostate cancer data analysis

The Prostate, Lung, Colorectal, and Ovarian (PLCO) Cancer Screening Trial is a large randomized trial designed and sponsored by the National Cancer Institute (NCI) to determine the effects of screening on cancer diagnosis and to reduce the cancer mortality. The study was initiated in 1993 and recruited participants who aged between 55 and older, had no previous history of any PLCO cancer, and were not participating in any other cancer screening and/or primary prevention trials. Details about the PLCO study can be found in Andriole et al. (2012).

The data considered here were withdrawn from the prostate cancer intervention arm, which contained 34,175 male participants. As an eligibility requirement, all patients had no history of any PLCO cancer. The response variable of interest was taken to be the age of prostate cancer diagnosis since the aim of this analysis was to

study the effects of some potential risk factors on the age of diagnosis. The diagnosis times were exactly observed for those participants who had prostate cancer during the study and were right-censored for the participants who did not contract prostate cancer by their last examination times. Ten binary covariates on participants' health status at the enrollment were considered in our analysis: education, with 1 for having at least college education; obese, with 1 for having a body mass index at least 30; heartd, with 1 for having history of any heart diseases; stroke, with 1 for having a stroke in the past; diabetes, with 1 for having diabetes; hepatitis, with 1 for having hepatitis; family history; with 1 for having at least one close relative who had prostate cancer before; psa history, with 1 suggesting having prostate-specific antigen testing in the past. In addition, race was also included in our analysis with four categories: Asian, African American, Caucasian, and other. Three dummy variables was created with Caucasian being the baseline category in our data analysis. After deleting the observations with missing values in the covariates, the final data set contained 20,553 observations in total, with a 90.6% right-censoring rate.

We applied our proposed method to this data set taking 2 for the degree of monotone splines but different number of knots based on the quantiles of the observed response values. Table 9 shows the AIC values from our methods using different number of knots. It turns out that the model with 18 interior knots had the smallest AIC value and then was selected as the final model.

For comparison, we also implemented the partial likelihood method from Cox model (Cox, 1972), the most popular semiparametric regression model in survival analysis. In general, a positive coefficient in the conventional Cox model implies a positive covariate effect on the hazard and thus leads to a shortened survival time. However, a positive regression coefficient in probit model (1) implies a prolonged survival time for the corresponding covariate. In order to provide easy comparison, we intentionally used $-x$ to replace x when fitting the Cox model so that the covariate

effects on the survival times have the same direction for the two models. Table 4.10 presents the he estimated covariate effects and their 95% confident intervals from these two models. Although the coefficients from the two models have different interpretations and are directly comparable, all of the estimated coefficients have the same signs and their 95% confidence intervals show the same status of whether they contain 0, indicating that the two methods identifies the same set of risk factors.

To conduct model diagnosis, we calculated the residuals r_i's as described in Section 5. Notice that a Q-Q plot is not appropriate as the residuals are also subject to right-censoring. Figure 4.6 plots the estimated Kaplan-Meier curve and its its 95% confidence band based on the residuals r_i's and the censoring indicators δ_i's as well as the true survival function $1 - \Phi(\cdot)$ under the probit model assumption. As seen clearly in Figure 4.6, the true survival curve is very close to the estimated Kaplan-Meier curve, which suggests the validity of the probit model for this analysis. We also implemented a formal one-sample log-rank test using the R function "LogRank1" implemented by Professor Mai Zhou at University of Kentucky. The log-rank test produced a p-value 0.9379, also indicating no significant difference between the estimated Kaplan-Meier curve of the residual and the true survival function of a standard normal distribution. All these results suggest that the probit model is valid for this analysis.

Table 4.9: The calculated Kaike Information Criterion (AIC) values from the probit models with different numbers of interior knots for the prostate cancer data analysis.

#	10	12	15	17	18	19	20
AIC	16799	16798	16796	16791	16789	16790	16792

Table 4.10: Estimated covariate effects from the probit model and Cox model.

	Proposed model			Cox model		
Covariate	$\hat{\beta}$	Lower	Upper	$\hat{\beta}$	Lower	Upper
educ	-0.094	-0.146	-0.042	-0.151	-0.242	-0.060
obese	-0.069	-0.130	-0.008	-0.099	-0.208	0.011
heartd	0.226	0.151	0.300	0.389	0.254	0.524
stroke	0.223	0.062	0.383	0.341	0.050	0.632
smoker	-0.123	-0.192	-0.053	-0.223	-0.346	-0.100
diabetes	0.252	0.157	0.348	0.427	0.253	0.602
hepatitis	0.122	-0.021	0.262	0.211	-0.047	0.468
fam_hist	-0.332	-0.417	-0.247	-0.532	-0.671	-0.394
psa_hist	0.048	-0.004	0.103	0.077	-0.017	0.172
raceb	-0.321	-0.428	-0.215	-0.557	-0.733	-0.381
racea	0.456	0.284	0.628	0.700	0.378	1.022
raceo	0.030	-0.143	0.203	0.089	-0.246	0.425

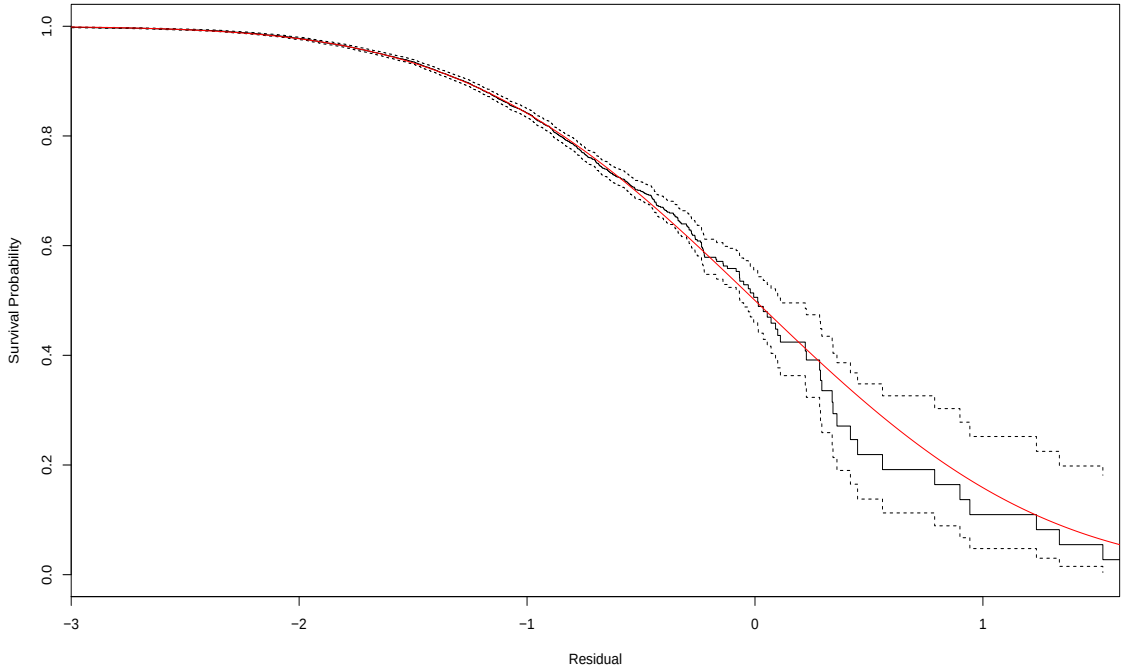

Figure 4.3: The true survival curve (red solid), the estimated Kaplan-Meier curve, and its 95% pointwise confidence band based on the residuals of the proposed probit model for the analysis of prostate cancer data.

4.8 Concluding remarks

This chapter proposes a flexible semi-parametric transformation model for the regression analysis for both complete and right-censored data. The proposed model is simple as the random error has a simple standard normal distribution. Our model is more flexible than the regression analysis based on Box-Cox transformed response since the transformation function is unknown. This model was studied in survival literature for analyzing interval-censored data (Lin and Wang, 2011), but little if any research has been reported for analyzing complete and right-censored data directly to the best of our knowledge.

A maximum likelihood approach based on ECM algorithms is developed to estimate the unknown transformation function and the regression parameters simultaneously for both complete and right-censored data. The proposed ECM algorithms are very easy to implement as they provides explicit expressions for all parameters in each iteration of the algorithms. The ECM algorithms are found to be robust to initial values, easy to implement, and fast to converge. Furthermore, the variance estimate is also obtained in closed form. The simulation results show that the proposed approach has excellent performance in estimating both the regression parameters and the baseline CDF (also the transformation function) in both complete and right-censored data. It also outperforms the regression analysis using the Box-Cox transformed response as seen in our and real-life application when the response variable is complete. Model diagnosis methods have been proposed to test the validity of the probit model and are illustrated in the real data applications. The proposed method can be further extended to more complicated survival data, such as partially interval-censored data, which contain a mixture of exactly observed observations and interval-censored observations.

Bibliography

Akaike, Hirotogu (1998). "Information theory and an extension of the maximum likelihood principle". In: *Selected papers of hirotugu akaike*. Springer, pp. 199–213.

Alaíz, Carlos M, Barbero, Alvaro, and Dorronsoro, José R (2013). "Group fused lasso". In: *International Conference on Artificial Neural Networks*. Springer, pp. 66–73.

Allison, Paul D (2008). "Convergence failures in logistic regression". In: *SAS Global Forum*. Vol. 360, pp. 1–11.

Anderson Jr, William N and Morley, Thomas D (1985). "Eigenvalues of the Laplacian of a graph". In: *Linear and multilinear algebra* 18.2, pp. 141–145.

Bennett, Kristin P (1994). "Global tree optimization: A non-greedy decision tree algorithm". In: *Computing Science and Statistics*, pp. 156–156.

Bertsekas, Dimitri P (1997). "Nonlinear programming". In: *Journal of the Operational Research Society* 48.3, pp. 334–334.

Bleakley, Kevin and Vert, Jean-Philippe (2011). "The group fused lasso for multiple change-point detection". In: *arXiv preprint arXiv:1106.4199*.

Böhning, Dankmar (1992). "Multinomial logistic regression algorithm". In: *Annals of the institute of Statistical Mathematics* 44.1, pp. 197–200.

Box, George EP and Cox, David R (1964). "An analysis of transformations". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 26.2, pp. 211–243.

Breiman, Leo (2017). *Classification and regression trees*. Routledge.

Breslow, Norman E (1975). "Analysis of survival data under the proportional hazards model". In: *International Statistical Review/Revue Internationale de Statistique*, pp. 45–57.

Buta, R (1987). “The structure and dynamics of ringed galaxies. III-Surface photometry and kinematics of the ringed nonbarred spiral NGC 7531”. In: *The Astrophysical Journal Supplement Series* 64, pp. 1–37.

Chambers, John M (2017). *Graphical Methods for Data Analysis: 0*. Chapman and Hall/CRC.

Cheng, Guang, Wang, Xiao, et al. (2011). “Semiparametric additive transformation model under current status data”. In: *Electronic Journal of Statistics* 5, pp. 1735–1764.

Chi, Eric C and Lange, Kenneth (2015). “Splitting methods for convex clustering”. In: *Journal of Computational and Graphical Statistics* 24.4, pp. 994–1013.

Cox, David R (1972). “Regression models and life-tables”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 34.2, pp. 187–202.

Davies, P Laurie and Kovac, Arne (2001). “Local extremes, runs, strings and multiresolution”. In: *Annals of Statistics*, pp. 1–48.

Dempster, Arthur P, Laird, Nan M, and Rubin, Donald B (1977). “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1, pp. 1–22.

Efron, Bradley, Hastie, Trevor, Johnstone, Iain, Tibshirani, Robert, et al. (2004). “Least angle regression”. In: *The Annals of statistics* 32.2, pp. 407–499.

Friedman, Jerome, Hastie, Trevor, Höfling, Holger, Tibshirani, Robert, et al. (2007). “Pathwise coordinate optimization”. In: *The annals of applied statistics* 1.2, pp. 302–332.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Rob (2010). “Regularization paths for generalized linear models via coordinate descent”. In: *Journal of statistical software* 33.1, p. 1.

Friedman, Jerome H et al. (1991). “Multivariate adaptive regression splines”. In: *The annals of statistics* 19.1, pp. 1–67.

Fritsch, Jürgen, Finke, Michael, and Waibel, Alex (1997). “Adaptively growing hierarchical mixtures of experts”. In: *Advances in Neural Information Processing Systems*, pp. 459–465.

Gnanadesikan, Ramanathan and Wilk, Martin B (1968). “Probability plotting methods for the analysis of data”. In: *Biometrika* 55.1, pp. 1–17.

Goldstein, Tom, O'Donoghue, Brendan, Setzer, Simon, and Baraniuk, Richard (2014). "Fast alternating direction optimization methods". In: *SIAM Journal on Imaging Sciences* 7.3, pp. 1588–1623.

Grubinger, Thomas, Zeileis, Achim, and Pfeiffer, Karl-Peter (2011). *evtree: Evolutionary learning of globally optimal classification and regression trees in R*. Tech. rep. Working Papers in Economics and Statistics.

Harrison Jr, David and Rubinfeld, Daniel L (1978). "Hedonic housing prices and the demand for clean air". In: *Journal of environmental economics and management* 5.1, pp. 81–102.

Hoefling, Holger (2010). "A path algorithm for the fused lasso signal approximator". In: *Journal of Computational and Graphical Statistics* 19.4, pp. 984–1006.

Jacobs, Robert A, Jordan, Michael I, Nowlan, Steven J, Hinton, Geoffrey E, et al. (1991). "Adaptive mixtures of local experts." In: *Neural computation* 3.1, pp. 79–87.

Jacobs, Robert A, Peng, Fengchun, and Tanner, Martin A (1997). "A Bayesian approach to model selection in hierarchical mixtures-of-experts architectures". In: *Neural Networks* 10.2, pp. 231–241.

James, Nicholas A and Matteson, David S (2013). "ecp: An R package for nonparametric multiple change point analysis of multivariate data". In: *arXiv preprint arXiv:1309.3295*.

Johnson, Nicholas A (2013). "A dynamic programming algorithm for the fused lasso and l 0-segmentation". In: *Journal of Computational and Graphical Statistics* 22.2, pp. 246–260.

Jordan, Michael I and Jacobs, Robert A (1994). "Hierarchical mixtures of experts and the EM algorithm". In: *Neural computation* 6.2, pp. 181–214.

Kanaujia, Atul and Metaxas, Dimitris (2006). "Learning ambiguities using Bayesian mixture of experts". In: *2006 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*. IEEE, pp. 436–440.

Kaplan, Edward L and Meier, Paul (1958). "Nonparametric estimation from incomplete observations". In: *Journal of the American statistical association* 53.282, pp. 457–481.

Li, Ni, Sun, Liuquan, and Sun, Jianguo (2010). "Semiparametric transformation models for panel count data with dependent observation process". In: *Statistics in Biosciences* 2.2, pp. 191–210.

Lin, X. and Wang, L (2011). "Bayesian proportional odds models for analyzing current status data: univariate, clustered, and multivariate". In: *Communications in Statistics - Simulation and Computation* 40, pp. 1171–1181.

Lin, Xiaoyan and Wang, Lianming (2010). "A semiparametric probit model for case 2 interval-censored failure time data". In: *Statistics in medicine* 29.9, pp. 972–981.

Loh, Wei-Yin (2014). "Fifty years of classification and regression trees". In: *International Statistical Review* 82.3, pp. 329–348.

Louis, Thomas A (1982). "Finding the observed information matrix when using the EM algorithm". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 44.2, pp. 226–233.

Manly, BFJ (1976). "Exponential data transformations". In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 25.1, pp. 37–42.

Meng, Xiao-Li and Rubin, Donald B (1993). "Maximum likelihood estimation via the ECM algorithm: A general framework". In: *Biometrika* 80.2, pp. 267–278.

Norouzi, Mohammad, Collins, Maxwell, Johnson, Matthew A, Fleet, David J, and Kohli, Pushmeet (2015). "Efficient non-greedy optimization of decision trees". In: *Advances in Neural Information Processing Systems*, pp. 1729–1737.

Ramsay, James O et al. (1988). "Monotone regression splines in action". In: *Statistical science* 3.4, pp. 425–441.

Rasmussen, Carl E and Ghahramani, Zoubin (2002). "Infinite mixtures of Gaussian process experts". In: *Advances in neural information processing systems*, pp. 881–888.

Saito, Kazumi and Nakano, Ryohei (1996). "A constructive learning algorithm for an HME". In: *Proceedings of International Conference on Neural Networks (ICNN'96)*. Vol. 2. IEEE, pp. 1268–1273.

Sanderson, Conrad and Curtin, Ryan (2016). "Armadillo: a template-based C++ library for linear algebra". In: *Journal of Open Source Software* 1.2, p. 26.

Schlegel, Peter (1970). "The explicit inverse of a tridiagonal matrix". In: *Mathematics of Computation* 24.111, p. 665.

Schlesselman, J (1971). "Power families: a note on the Box and Cox transformation". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 33.2, pp. 307–311.

Shapiro, Samuel Sanford and Wilk, Martin B (1965). "An analysis of variance test for normality (complete samples)". In: *Biometrika* 52.3/4, pp. 591–611.

Stamey, Thomas A, Kabalin, John N, McNeal, John E, Johnstone, Iain M, Freiha, Fuad, Redwine, Elise A, and Yang, Norman (1989). "Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients". In: *The Journal of urology* 141.5, pp. 1076–1083.

Team, R Core et al. (2013). "R: A language and environment for statistical computing". In:

Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288.

Tibshirani, Robert, Saunders, Michael, Rosset, Saharon, Zhu, Ji, and Knight, Keith (2005). "Sparsity and smoothness via the fused lasso". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67.1, pp. 91–108.

Tibshirani, Ryan Joseph (2011). *The solution path of the generalized lasso*. Stanford University.

Tseng, Paul (1991). "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities". In: *SIAM Journal on Control and Optimization* 29.1, pp. 119–138.

Ueda, Naonori and Ghahramani, Zoubin (2000). "Optimal model inference for Bayesian mixture of experts". In: *Neural Networks for Signal Processing X. Proceedings of the 2000 IEEE Signal Processing Society Workshop (Cat. No. 00TH8501)*. Vol. 1. IEEE, pp. 145–154.

Waterhouse, SR and Robinson, AJ (1995). "Pruning and growing hierachical mixtures of experts". In:

Wytock, Matt, Sra, Suvrit, and Kolter, Jeremy Z (2014). "Fast Newton methods for the group fused lasso." In: *UAI*, pp. 888–897.

Xing, Hong-Jie and Hu, Bao-Gang (2008). "An adaptive fuzzy c-means clustering-based mixtures of experts model for unlabeled data classification". In: *Neurocomputing* 71.4-6, pp. 1008–1021.

Xu, Da, Zhao, Shishun, Hu, Tao, Yu, Mengzhu, and Sun, Jianguo (2019). "Regression analysis of informative current status data with the semiparametric linear transformation model". In: *Journal of Applied Statistics* 46.2, pp. 187–202.

Yeo, In-Kwon and Johnson, Richard A (2000). "A new family of power transformations to improve normality or symmetry". In: *Biometrika* 87.4, pp. 954–959.

Yuan, Ming and Lin, Yi (2006). "Model selection and estimation in regression with grouped variables". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1, pp. 49–67.

Zeng, Donglin, Mao, Lu, and Lin, DY (2016). "Maximum likelihood estimation for semiparametric transformation models with interval-censored data". In: *Biometrika* 103.2, pp. 253–271.

Zhang, Bin, Tong, Xingwei, Zhang, Jing, Wang, Chunjie, and Sun, Jianguo (2013). "Efficient estimation for linear transformation models with current status data". In: *Communications in Statistics-Theory and Methods* 42.17, pp. 3191–3203.

Zhang, Zhigang (2009). "Linear transformation models for interval-censored data: prediction of survival probability and model checking". In: *Statistical Modelling* 9.4, pp. 321–343.

Appendix A

Quantities involved in the variance estimate of $\hat{\theta}$

The first part in Louis' method involves finding the second derivatives of $Q(\theta, \theta^{(d)})$ with respect to θ. The detailed formula of these quantities are

$$\begin{aligned}
\frac{\partial^2 Q(\theta,\theta^{(d)})}{\partial\beta\partial\beta'} &= \sum_{i=1}^{N} -X_i X_i^T, \\
\frac{\partial^2 Q(\theta,\theta^{(d)})}{\partial\gamma_0^2} &= -N, \\
\frac{\partial^2 Q(\theta,\theta^{(d)})}{\partial\gamma_k^2} &= -\sum_{i=1}^{N}\{b_k^2(y_i) + \gamma_k^{-2} h_{ik}^{(d)}\delta_i\}, \\
\frac{\partial^2 Q(\theta,\theta^{(d)})}{\partial\beta\partial\gamma_0} &= \sum_{i=1}^{N} X_i^T, \\
\frac{\partial^2 Q(\theta,\theta^{(d)})}{\partial\beta\partial\gamma_k} &= \sum_{i=1}^{N} b_k(y_i) X_i^T, \\
\frac{\partial^2 Q(\theta,\theta^{(d)})}{\partial\gamma_0\partial\gamma_k} &= -\sum_{i=1}^{N} b_k(y_i), \\
\frac{\partial^2 Q(\theta,\theta^{(d)})}{\partial\gamma_l\partial\gamma_k} &= -\sum_{i=1}^{N} b_l(y_i) b_k(y_i), \qquad k \neq l.
\end{aligned}$$

The quantities involved in $var(\partial \log L_c/\partial\theta|\mathcal{D},\theta)$ are

$$
\begin{aligned}
var(\frac{\partial \log L_c}{\partial \beta}|\mathcal{D},\theta) &= \sum_{i=1}^{N} X_i X_i^T var(Z_i|\mathcal{D},\theta), \\
var(\frac{\partial \log L_c}{\partial \gamma_0}|\mathcal{D},\theta) &= \sum_{i=1}^{N} var(Z_i|\mathcal{D},\theta), \\
var(\frac{\partial \log L_c}{\partial \gamma_k}|\mathcal{D},\theta) &= \sum_{i=1}^{N} b_k^2(y_i) var(Z_i|\mathcal{D},\theta) + \gamma_k^{-2} var(u_{ik}|\mathcal{D},\theta)\delta_i, \\
cov(\frac{\partial \log L_c}{\partial \beta}, \frac{\partial \log L_c}{\partial \gamma_0}|\mathcal{D},\theta) &= \sum_{i=1}^{N} X_i^T var(Z_i|\mathcal{D},\theta), \\
cov(\frac{\partial \log L_c}{\partial \gamma_k}, \frac{\partial \log L_c}{\partial \gamma_0}|\mathcal{D},\theta) &= \sum_{i=1}^{N} b_k(y_i) var(Z_i|\mathcal{D},\theta), \\
cov(\frac{\partial \log L_c}{\partial \beta}, \frac{\partial \log L_c}{\partial \gamma_k}|\mathcal{D},\theta) &= -\sum_{i=1}^{N} b_k(y_i) X_i^T var(Z_i|\mathcal{D},\theta), \\
cov(\frac{\partial \log L_c}{\partial \gamma_k}, \frac{\partial \log L_c}{\partial \gamma_l}|\mathcal{D},\theta) &= \sum_{i=1}^{N} b_k(y_i) b_l(y_i) var(Z_i|\mathcal{D},\theta) + \gamma_k {\gamma_l}^{-1} cov(u_{ik}, u_{il}|\mathcal{D},\theta)\delta_i,
\end{aligned}
$$

where

$$
\begin{aligned}
var(u_{ik}|\mathcal{D},\theta) &= \frac{\gamma_k m_k(y_i)}{\sum_{q=1}^{K} \gamma_q m_q(y_i)} \{1 - \frac{\gamma_k m_k(y_i)}{\sum_{q=1}^{K} \gamma_q m_q(y_i)}\} \delta_i, \\
cov(u_{ik}, u_{il}|\mathcal{D},\theta) &= \frac{\gamma_k \gamma_l m_l(y_i) m_k(y_i)}{\{\sum_{q=1}^{K} \gamma_q m_q(y_i)\}^2} \delta_i, \qquad k \neq l \\
var(Z_i|\mathcal{D},\theta) &= \left\{1 + \frac{u_i \phi(u_i)}{\Phi(u_i)} - \frac{\phi(u_i)^2}{\Phi(u_i)^2}\right\}(1 - \delta_i),
\end{aligned}
$$

and $u_i = \gamma_0 + \sum_{k=1}^{K} \gamma_k b_k(y_i) - X_i^T \beta$ for $i = 1, \cdots, N$.