DEPARTMENT OF STATISTICS
University of Wisconsin
1210 West Dayton St.
Madison, WI 53706

TECHNICAL REPORT NO. 1003

December 20, 1998

SMOOTHING SPLINE ANALYSIS OF VARIANCE FOR POLYCHOTOMOUS RESPONSE DATA [1]

by

Xiwu Lin

[1]Research sponsored in part by NEI under Grant R01 EY09946 and in part by NSF under Grant DMS-9704758.

SMOOTHING SPLINE ANALYSIS OF VARIANCE FOR POLYCHOTOMOUS RESPONSE DATA

By
Xiwu Lin

A dissertation submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy
(Statistics)

at the
UNIVERSITY OF WISCONSIN – MADISON
1998

Abstract

We consider the penalized likelihood method with smoothing spline ANOVA for estimating non-parametric functions to data involving a polychotomous response. The fitting procedure involves minimizing the penalized likelihood in a Reproducing Kernel Hilbert Space. One Step Block SOR-Newton-Raphson Algorithm is used to solve the minimization problem. Generalized Cross-Validation or unbiased risk estimation is used to empirically assess the amount of smoothing (which controls the bias and variance trade-off) at each one-step Block SOR-Newton-Raphson iteration. Under some regular smoothness conditions, the one-step Block SOR-Newton-Raphson will produce a sequence which converges to the minimizer of the penalized likelihood for the fixed smoothing parameters. Monte Carlo simulations are conducted to examine the performance of the algorithm. The method is applied to polychotomous data from the Wisconsin Epidemiological Study of Diabetic Retinopathy to estimate the risks of cause-specific mortality given several potential risk factors at the start of the study. Strategies to obtain smoothing spline estimates for large data sets with polychotomous response are also proposed in this thesis. Simulation studies are conducted to check the performance of the proposed method.

Acknowledgements

I would like to express my sincerest gratitude to my advisor, Professor Grace Wahba, for her invaluable advice during the course of this dissertation.

Appreciation is extended to Professors Michael Kosorok, Mary Lindstrom, Olvi Mangasarian, and Kam-Wah Tsui for their service on my final examination committee, their careful reading of this thesis and their valuable comments. I would like to thank Ronald Klein, MD and Barbara Klein, MD for providing the WESDR data.

Fellow graduate students Fangyu Gao, Kin-Yee Chan, Chen Wang, Yonghua Chen and Yong Zeng have helped me in various ways in the course of this study and the actual writing of the thesis. These and other graduate students made my life in Madison an enjoyable one.

Finally, special thanks go to my family and my wife Xiaoqun for their love and support.

This research is sponsored in part by NEI under Grant R01 EY09946 and in part by NSF under Grant DMS-9704758.

Contents

Abstract i

Acknowledgements ii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Outline of the Thesis . . . 2

2 Penalized Polychotomous Regression using smoothing spline ANOVA 3
- 2.1 Polychotomous Logistic Regression . . . 3
- 2.2 Penalized Polychotomous Regression . . . 4
- 2.3 Smoothing Spline Analysis of Variance . . . 5
- 2.4 Penalized Polychotomous Regression Using Smoothing Spline Analysis of Variance . 6

3 Fitting the Penalized Polychotomous Regression 9
- 3.1 Introduction . . . 9
- 3.2 Block Nonlinear SOR methods . . . 9
- 3.3 Implementation of the Algorithm . . . 11
- 3.4 Choosing the Smoothing Parameters . . . 13
- 3.5 Bayesian Inference . . . 14
- 3.6 Monte Carlo Examples . . . 17

4 Strategies for Large Data Sets 20
- 4.1 Binary Case . . . 20
  - 4.1.1 Generalized Approximate Cross Validation . . . 20
  - 4.1.2 RGACV and One-Step-RGACV . . . 21
  - 4.1.3 Approximate Smoothing Spline . . . 24
  - 4.1.4 Minimizing the OneStepRGACV function . . . 25
  - 4.1.5 Bayesian Confidence Intervals for the Approximate Solution . . . 25
  - 4.1.6 Monte Carlo Simulation . . . 28
- 4.2 Polychotomous Case . . . 53
  - 4.2.1 Fitting Polychotomous Response Data by Individual Fitting . . . 53
  - 4.2.2 Randomized GACV for Penalized Polychotomous Regression . . . 60

5 Application to Wisconsin Epidemiological Study of Diabetic Retinopathy 64
- 5.1 Introduction . . . 64
- 5.2 Estimate the Risks of Cause-specific Mortality by Penalized Polychotomous Regression 65

6 Concluding Remarks 74
- 6.1 Summary . . . 74
- 6.2 Future Research . . . 74

Bibliography 75

Chapter 1

Introduction

1.1 Motivation

In many demographic medical studies, records of attribute vectors as well as records of the outcome for each example (patient) for n examples are available as training data. Usually, the outcome is a categorical random variable that takes on a finite number of values (nominal) which we refer to as classes. This is a multiple classification problem in statistics if we want to predict the outcome based on the attribute vectors. In some other situation, we might be interested in the estimation of the class probability given the attribute vectors.

Many methods have been proposed for the multiple classification problems. One of the popular modern multiple classification techniques is CART (Breiman, Friedman, Olshen and Stone, 1984), which approaches the multiple classification problem using recursive partitioning techniques. Hastie, Tibshirani and Buja (1994) introduce flexible discriminant analysis, which combines nonparametric regression techniques with discriminant analysis. Villalobos and Wahba (1983) proposed classification using an approach of estimating the posterior class probability based on maximum penalized log likelihood estimation using multivariate thin plate splines. Bose (1994) proposes classification using splines which employs least squares regression and additive cubic splines. In Computer Sciences, neural networks is one of the popular techniques for classification. See Ripley (1994) for details.

It can be shown that the optimal classification rule predicts Y to be $argmax_k P(Y = k|X)$. Most of the popular classification methods try to find $argmax_k P(Y = k|X)$ without precise estimation of the conditional class probability. For multiple classification problems, it is assumed that any two examples with the same attribute vector will always be in same class, whereas in some studies this is not necessarily the case. For example, in medical studies two patients with the same attribute vector will not necessarily have the same medical outcome. Clearly, multiple classification methods are not useful in such applications. Instead, we are more interested in estimating the probability of a particular outcome given the attribute vector.

A popular technique used to obtain an estimate of all the conditional class probabilities is multiple logistic regression (polychotomous regression). Traditionally, we assume linear (parametric) forms for all the logit functions to be estimated. The details of the linear polychotomous regression techniques can be found in Hosmer and Lemeshow (1989). However, the linear assumption or even quadratic or cubic models may not be adequate in some applications , and the results obtained by assuming linear forms might be misleading.

A variety of approaches have been proposed to allow more flexibility than is inherent in simple parametric models. We will not review the general literature, other than to note that regression splines have been used for this purpose by, for example Kooperberg, Bose and Stone (1997). In their paper, they combine MARS with polychotomous regression to provide estimates for conditional class probabilities. On the other hand, the smoothing spline analysis of variance, as a nonparametric method, has been successfully used in many area as a tool for data analysis. Wahba, Wang, Gu, Klein and Klein (1995, referred as WWGKK) provide a general setting for applying smoothing

spline ANOVA to data from exponential families. Their method is successfully applied to analyze medical data with Bernoulli outcomes. This is a motivation to use smoothing spline ANOVA to model data with polychotomous response.

In this thesis, we will investigate various approaches using smoothing spline ANOVA technique to obtain an estimate of the class probabilities for data with polychotomous responses. For moderate data sets, an iterative method based on penalized likelihood is proposed. For large data sets, two methods are proposed to overcome the computational difficulties. In one method, we propose a fast algorithm for a large data set with Bernoulli responses and model the polychotomous data based on the binary data algorithm. Alternatively, we can use the techniques employed in developing the fast algorithm for binary data to speed up the iterative method based on the penalized likelihood for polychotomous data.

1.2 Outline of the Thesis

In Chapter 2 of this thesis, we discuss the penalized polychotomous regression using Smoothing Splines Analysis of Variance. The penalized likelihood for the polychotomous response is established and the existence of the solution is investigated. We also review smoothing spline analysis of variance and apply it to the polychotomous regression.

In Chapter 3, we propose a numerical method called 'Block one-step SOR-Newton-Raphson' to solve the penalized polychotomous regression problem. A connection between the smoothing estimate and a Bayesian problem is also discussed in this chapter.

In Chapter 4, we first introduce a fast algorithm to get the smoothing spline estimate for binary data. A randomized version of generalized cross-validation is derived to choose the smoothing parameters. An approximate solution is proposed to speed up the computation. Bayesian confidence intervals are constructed for the approximate solution. To obtain smoothing spline estimate for large data sets with polychotomous response, we will discuss two possible strategies: (1) using the fast algorithm for binary data; (2) deriving a randomized GACV formula similar to that for binary data.

To illustrate the penalized polychotomous regression method, we apply it to investigate the association between some risk factors and the cause-specific mortality in a data set collected form the Wisconsin Epidemiologic Study of Retinopathy. This is done in Chapter 5. Finally, a concluding remark is made in Chapter 6.

Chapter 2

Penalized Polychotomous Regression using smoothing spline ANOVA

2.1 Polychotomous Logistic Regression

Assume that the categories of the outcome variable, Y, are coded $0, 1, \cdots, k$. Suppose the distribution of Y depends on the predictors $x_1, \cdots, x_d$, where $x = (x_1, ..., x_d)$ ranges over the subset $\mathcal{X}$ of $\mathcal{R}^d$. Now let x be distributed as a random vector, i.e. consider a random pair (X, Y). Suppose $P(Y = i|X = x) > 0$ and let

$$f_i(x) = log\frac{P(Y = i|X = x)}{P(Y = 0|X = x)}, \quad i = 1, \cdots, k, \tag{2.1.1}$$

then

$$P(Y = i|X = x) = \frac{exp(f_i(x))}{1 + exp(f_1(x)) + ... + exp(f_k(x))}, \quad i = 1, \cdots, k, \tag{2.1.2}$$

$$P(Y = 0|X = x) = \frac{1}{1 + exp(f_1(x)) + ... + exp(f_k(x))}. \tag{2.1.3}$$

We refer to (2.1.1) as the polychotomous regression model; when $k = 1$ it is referred to as the logistic regression model.

Denoting $p_i(t) = P(Y = i|X = t)$, we can write down the conditional likelihood of observing y given covariate X=t as follows,

$$\prod_{i=0}^{k} p_i(t)^{I[y=i]} = \exp\{\sum_{i=1}^{k} I[y = i]f_i(t) - \log(1 + \sum_{i=1}^{k} e^{f_i(t)})\},$$

so the negative log-likelihood is

$$-\sum_{i=1}^{k} I[y = i]f_i(t) + \log(1 + \sum_{i=1}^{k} e^{f_i(t)}).$$

Suppose we have observations $(t_1, y_1), \ldots, (t_n, y_n)$, then the negative log-likelihood based on the observations is

$$\mathcal{L}(y, f_1, ..., f_k) = -\sum_{j=1}^{n}\{\sum_{i=1}^{k} I[y_j = i]f_i(t_j) - \log(1 + \sum_{i=1}^{k} e^{f_i(t_j)})\}. \tag{2.1.4}$$

If we denote $y_{ij} = I[y_j = i]$, the negative log likelihood can be written as follows,

$$\mathcal{L}(y, f_1, ..., f_k) = \sum_{j=1}^{n} \{ - \sum_{i=1}^{k} y_{ij} f_i(t_j) + \log(1 + \sum_{i=1}^{k} e^{f_i(t_j)}) \}. \tag{2.1.5}$$

The usual parametric approach to the polychotomous regression problem is to use linear model

$$f_i(x) = \beta_{i0} + \beta_{i1} x_1 + \cdots + \beta_{id} x_d.$$

The parameters β_{ij} are obtained by the maximum likelihood method. The negative log likelihood is convex and twice differentiable, and iterative procedure such as Newton-Raphson method can be used to get the **ML** estimate of the parameters.

2.2 Penalized Polychotomous Regression

To achieve greater flexibility, many authors proposed nonparametric regression models to relax the rigid linear assumption. In particular, the penalized likelihood smoothing spline for data from exponential families (O'Sullivan, 1983; Wahba et al., 1995) assumes that the function is smooth but imposes some roughness penalty on the function. Following this approach, we can assume each logit function f_i is smooth but imposes a roughness penalty $J(f_1, \cdots, f_k)$ on the functions. More precisely, we will assume that $f_i \in \mathcal{H}^i$, where $\mathcal{H}^i$ is a reproducing kernel Hilbert space. A reproducing kernel Hilbert space(RKHS) is a Hilbert space of functions on $\mathcal{X}$ in which the evaluation functional is continuous (Aronszaj, 1950). The penalized polychotomous regression estimates $f_1, \cdots, f_k$ are obtained by finding $f_i \in \mathcal{H}^i$ to minimize the penalized likelihood

$$\mathcal{L}_\lambda(f_1, \cdots, f_k) = - \sum_{j=1}^{n} l_j(f_1, \cdots, f_k) + J_\lambda(f_1, \cdots, f_k), \tag{2.2.1}$$

where the first part is the negative log-likelihood and $l_j = \sum_{i=1}^{k} y_{ij} f_i(x_j) - log(1 + \sum_{i=1}^{k} e^{f_i(x_j)})$. It measures the goodness of fit. The second part is the penalty function. For simplicity and easy interpretation, we will assume that the penalty function is in additive form, i.e.,

$$J_\lambda(f_1, \cdots, f_k) = \sum_{i=1}^{k} \lambda_i J^i(f_i).$$

Suppose $\mathcal{H}^i = \mathcal{H}_0^i \oplus \mathcal{H}_1^i$, where $\mathcal{H}_0^i$ is finite dimensional (the "parametric" part, usually polynomials), and $\mathcal{H}_1^i$ (the "smooth" part) is the ortho-complement of $\mathcal{H}_0^i$ in $\mathcal{H}^i$. Let $J^i(f) = ||P_1^i f||^2$, where P_1^i is the orthogonal projection operator in $\mathcal{H}^i$ onto $\mathcal{H}_1^i$, then the penalized likelihood will become

$$\mathcal{L}(f_1, \cdots, f_k, \lambda) = - \sum_{j=1}^{n} l_j(f_1, \cdots, f_k) + \sum_{i}^{k} \lambda_i ||P_1^i f_i||^2. \tag{2.2.2}$$

Denoting $J_\perp$ be the null space of $\mathcal{H}^1 \times \cdots \times \mathcal{H}^k$ with respect to the penalty function J_λ, we have the following theorem.

Theorem 2.1 *If the minimizer of (2.2.2) exists in $J_\perp$, it uniquely exists in $\mathcal{H}^1 \times \cdots \times \mathcal{H}^k$*

Before we prove this theorem, we will first state two lemmas.

Lemma 2.1 *$\mathcal{L}(y, f_1, \ldots, f_k)$ in (2.1.5) is a convex function of $f_1, \ldots, f_k$.*

Proof. See page 438, example 5.3 of Theory of Point Estimation(Lehmann,1983).

The following Lemma is Theorem 4.1 from Gu and Qiu (1993).

Lemma 2.2 *Suppose L(g) is a continuous and strictly convex functional in a Hilbert space $\mathcal{H} = J_\perp \oplus \mathcal{H}_J$, where $\mathcal{H}_J$ has a square norm J(g) and $J_\perp$ is the null space of J(g) of finite dimension. If L(g) has a minimizer in $J_\perp$, then L(g) + J(g) has a unique minimizer in $\mathcal{H}$.*

Proof of Theorem 2.1
Let $\mathcal{H} = \{g | g(x, i) = f_i(x), \quad i = 1, \ldots, k, \text{ where } f_i \in H^i\}$.Then $\mathcal{H}$ is a Hilbert space with square semi-norm $J(g) = J(f_1, \ldots, f_k)$. Let $\mathcal{L}^*(g) = \mathcal{L}(y, f_1, ..., f_k)$. By Lemma 2.2, it suffices to show that $\mathcal{L}^*(g)$ is continuous and strictly convex in $\mathcal{H}$. Continuity is obvious. Strict convexity follows from Lemma 2.1. **Q.E.D.**

2.3 Smoothing Spline Analysis of Variance

Smoothing Spline Analysis (SS-ANOVA) models for Gaussian data are described in some generality in Wahba (1990, Chapter 10) where references to the previous literature are given. Wahba *et al.* (1995) and others, discussed further various aspects of these models. The code RKPACK (Gu 1989) will fit specified SS-ANOVA models given Gaussian data. The code GRKPACK (Wang 1997) which calls subroutines in RKPACK will fit specified SS-ANOVA models given data from one parameter exponential families.

Given a fairly arbitrary function $f(x_1, \cdots, x_d)$, a (functional) ANOVA decomposition of f may be defined as

$$f(x_1, ..., x_d) = \mu + \sum_{\alpha=1}^{d} f_\alpha(x_\alpha) + \sum_{\alpha\beta} f_{\alpha\beta}(x_\alpha, x_\beta) + ... + f_{1,...,d}(x_1, ..., x_d), \tag{2.3.1}$$

where the f_α are the main effects, $f_{\alpha\beta}$ are the two factor interactions, and so on. For those f satisfying some measurability conditions, a unique ANOVA decomposition of the above form can always be defined as follows. Let $d\mu_\alpha$ be a probability measure on $\mathcal{T}^{(\alpha)}$ and define the averaging operator $\mathcal{E}_\alpha$ on $\mathcal{T}$ by

$$(\mathcal{E}_\alpha f)(x) = \int_{\mathcal{T}^{(\alpha)}} f(x_1, ..., x_d) d\mu_\alpha(x_\alpha). \tag{2.3.2}$$

Then the identity is decomposed as

$$\begin{aligned} I &= \textstyle\prod_\alpha(\mathcal{E}_\alpha) + (I - \mathcal{E}_\alpha)) \\ &= \textstyle\prod_\alpha \mathcal{E}_\alpha + \sum_\alpha (I - \mathcal{E}_\alpha) \prod_{\beta\neq\alpha} \mathcal{E}_\beta + \sum_{\alpha<\beta} (I - \mathcal{E}_\alpha)(I - \mathcal{E}_\beta) \prod_{\gamma\neq\alpha,\beta} \mathcal{E}_\gamma \\ &\quad + ... + \textstyle\prod_\alpha (I - \mathcal{E}_\alpha) \end{aligned} \tag{2.3.3}$$

The components of this decomposition generate the ANOVA decomposition of f of the form (2.3.1) by $C = (\prod_\alpha \mathcal{E}_\alpha)f$, $f_\alpha = ((I - \mathcal{E}_\alpha) \prod_{\beta\neq\alpha} \mathcal{E}_\beta)f$, $f_{\alpha\beta} = ((I - \mathcal{E}_\alpha)(I - \mathcal{E}_\beta) \prod_{\gamma\neq\alpha,\beta} \mathcal{E}_\gamma)f$, and so forth.

The idea behind Smoothing Spline ANOVA is to construct an RKHS $\mathcal{H}$ of functions on $\mathcal{T}$ so that the components of the SS-ANOVA decomposition represent an orthogonal decomposition of f in $\mathcal{H}$. Then RKHS methods can be used to explicitly impose smoothness penalties of the form

$\sum_\alpha \lambda_\alpha J_\alpha(f_\alpha) + \sum_{\alpha\beta} \lambda_{\alpha\beta} J_{\alpha\beta}(f_{\alpha\beta}) + ...$, where, however, the series will be truncated at some point. This is done as follows. Let $\mathcal{H}^{(\alpha)}$ be an RKHS of functions on $\mathcal{T}^{(\alpha)}$ with $\int_{\mathcal{T}^{(\alpha)}} f_\alpha(x_\alpha) d\mu_\alpha = 0$ for $f_\alpha(x_\alpha) \in \mathcal{H}^{(\alpha)}$, and let $[1^{(\alpha)}]$ be the one dimensional space of constant functions on $\mathcal{T}^{(\alpha)}$. Construct $\mathcal{H}$ as

$$\mathcal{H} = \prod_{j=1}^{d} (\{[1^{(\alpha)}]\} \oplus \{\mathcal{H}^{(\alpha)}\}) = [1] \oplus \sum_\alpha \mathcal{H}^{(\alpha)} \oplus \sum_{\alpha<\beta} [\mathcal{H}^{(\alpha)} \otimes \mathcal{H}^{(\beta)}] \oplus \dots \tag{2.3.4}$$

where $[1]$ denotes the constant function on $\mathcal{T}$. With some abuse of notation, factors of the form $[1^\alpha]$ are omitted whenever they multiply a term of a different form. Thus $\mathcal{H}^{(\alpha)}$ is a shorthand for $[1^{(1)}] \otimes \cdots \otimes [1^{(\alpha\perp 1)}] \otimes [1^{(\alpha+1)}] \otimes \cdots \otimes [1^{(d)}]$ (which is a subspace of $\mathcal{H}$). The components of the ANOVA decomposition are now in mutually orthogonal subspaces of $\mathcal{H}$. Note that the components will depend on the measures $d\mu_\alpha$ and these should be chosen in specific application so that the fitted mean, main effects, two factor interactions, etc. have reasonable interpretations.

Next, $\mathcal{H}^{(\alpha)}$ is decomposed into a parametric part and a smooth part, by letting $\mathcal{H}^{(\alpha)} = \mathcal{H}_\pi^{(\alpha)} \oplus \mathcal{H}_S^{(\alpha)}$, where $\mathcal{H}_\pi^{(\alpha)}$ is finite dimensional (the "parametric" part) and $\mathcal{H}_S^{(\alpha)}$ (the "smooth" part) is the ortho-complement of $\mathcal{H}_\pi^{(\alpha)}$ in $\mathcal{H}^{(\alpha)}$. Elements of $\mathcal{H}_\pi^{(\alpha)}$ are not penalized through the device of letting $J_\alpha(f_\alpha) = \|P_S^{(\alpha)} f_\alpha\|^2$ where $P_S^{(\alpha)}$ is the orthogonal projector onto $\mathcal{H}_S^{(\alpha)}$. $[\mathcal{H}^{(\alpha)} \otimes \mathcal{H}^{(\beta)}]$ is now a direct sum of four orthogonal subspaces: $[\mathcal{H}^{(\alpha)} \otimes \mathcal{H}^{(\beta)}] = [\mathcal{H}_\pi^{(\alpha)} \otimes \mathcal{H}_\pi^{(\beta)}] \oplus [\mathcal{H}_\pi^{(\alpha)} \otimes \mathcal{H}_S^{(\beta)}] \oplus [\mathcal{H}_S^{(\alpha)} \otimes \mathcal{H}_\pi^{(\beta)}] \oplus [\mathcal{H}_S^{(\alpha)} \otimes \mathcal{H}_S^{(\beta)}]$. By convention the elements of the finite dimensional space $[\mathcal{H}_\pi^{(\alpha)} \otimes \mathcal{H}_\pi^{(\beta)}]$ will not be penalized. Continuing this way results in an orthogonal decomposition of $\mathcal{H}$ into sums of products of unpenalized finite dimensional subspaces, plus main effects 'smooth' subspaces, plus two factor interaction spaces of the form parametric $\otimes$ smooth, smooth $\otimes$ parametric and smooth $\otimes$ smooth and similarly for three and higher factor subspaces.

When a model is chosen, we can regroup and write the model space as

$$\mathcal{M} = \mathcal{H}_0 \oplus \sum_{l=1}^{q} \mathcal{H}_l, \tag{2.3.5}$$

where $\mathcal{H}_0$ is a finite dimensional space containing functions which are not going to be penalized. The norms on the composite $\mathcal{H}_l$ are the tensor product norms induced by the norms on the component subspaces, $||f||^2 = ||P_0 f||^2 + \sum_{l=1}^{q} ||P_l f||^2$, where P_l is the orthogonal projector in $\mathcal{M}$ onto $\mathcal{H}_l$. The smoothing spline ANOVA estimate of f is the solution to the following variational problem

$$\min_{f \in \mathcal{M}} \{\sum_{i=1}^{n} (y_i - f(x_i))^2 + n \sum_{l=1}^{q} \lambda_l ||P_l f||^2\}. \tag{2.3.6}$$

The first term in (2.3.6) is the sum of squared residuals which measures the goodness of fit while the second part is the penalty on roughness of the estimate. The λ_l's are smoothing parameters controlling the trade-off between goodness of fit and roughness. These smoothing parameters can be estimated from data by the generalized cross validation method or by the unbiased risk method (see Wahba 1990).

2.4 Penalized Polychotomous Regression Using Smoothing Spline Analysis of Variance

We assume that the data are polychotomous response data and we have chosen a model space $\mathcal{M}_i = \mathcal{H}_0^i \oplus \sum_{l=1}^{q_i} \mathcal{H}_l^i$ for each logit function f_i. As a direct generalization of (2.2.2) to multivariate functions

and a direct generalization of (2.3.6) to polychotomous response data, a penalized polychotomous regression smoothing spline analysis of variance estimate is the solution to the following variational problem:

$$\min_{f_i \in \mathcal{M}_i, i=1,\cdots,k} \{ -\sum_{j=1}^{n} l_j(f_{1j}, \cdots, f_{kj}) + \frac{n}{2} \sum_{i=1}^{k} \sum_{l=1}^{q_i} \lambda_{il} ||P_l^i f_i||^2 \}, \tag{2.4.1}$$

where $f_{ij} = f_i(x_j)$. The first part in (2.4.1) is the negative log likelihood. It measures the goodness of fit. In the second part, P_l^i is the orthogonal projector in $\mathcal{M}_i$ onto $\mathcal{H}_l^i$ and $||P_l^i f||^2$ is a roughness penalty. The λ_{il}'s are a set of smoothing parameters which controls the trade-off between goodness of fit and roughness of the estimate. We will discuss how to choose the smoothing parameters and solve the variational problem in the next chapter. If we let $\lambda_{il} = \lambda_i/\theta_{il}$, (2.4.1) becomes

$$\min_{f_i \in \mathcal{M}_i, i=1,\cdots,k} \{ -\sum_{j=1}^{n} l_j(f_{1j}, \cdots, f_{kj}) + \frac{n}{2} \sum_{i=1}^{k} \lambda_i ||P_*^i f_i||_{\Theta_i}^2 \}, \tag{2.4.2}$$

where $P_*^i = \sum_{l=1}^{q_i} P_l^i$ is the orthogonal projection in $\mathcal{M}_i$ onto $\mathcal{H}_*^i = \sum_{l=1}^{q_i} \mathcal{H}_l^i$ and

$$||f||_{\Theta_i}^2 = ||P_0^i f||^2 + \sum_{l=1}^{q_i} \theta_{il}^{\perp 1} ||P_l^i f||^2,$$

is a modified norm of $\mathcal{M}_i$ indexed by $\Theta_i = (\theta_{i1}, \cdots, \theta_{iq_i})$. We denote by R_l^i the reproducing kernel for $\mathcal{H}_l^i$ under the original norm. It can be shown that $\theta_{il} R_l^i$ is the RK for $\mathcal{H}_l^i$ under the norm $||\cdot||_{\Theta_i}$. Thus the RK for $\mathcal{H}_*^i$ under $||\cdot||_{\Theta_i}$ is

$$R_{\Theta_i} = \sum_{l=1}^{q_i} \theta_{il} R_l^i. \tag{2.4.3}$$

Since the RK of the tensor product space is the product of the RK's of the component space, the computation of the R_l^i's is straightforward. For example, if $R_{\mathcal{H}_\pi^{(j)}}(\cdot,\cdot)$ and $R_{\mathcal{H}_S^{(k)}}(\cdot,\cdot)$ are the RK corresponding to the Hilbert spaces $\mathcal{H}_\pi^{(j)}$ and $\mathcal{H}_S^{(k)}$ respectively, the RK corresponding to the tensor product space $\mathcal{H}_\pi^{(j)} \otimes \mathcal{H}_S^{(k)}$ is

$$R_{\mathcal{H}_\pi^{(j)}}(x_j(j_1), x_j(j_2)) R_{\mathcal{H}_S^{(k)}}(x_k(k_1), x_k(k_2)),$$

where $x_u(v)$ denotes the uth coordinate of the vth design point.

Similar to Wahba (1990), we will show that the minimizer of the penalized likelihood for polychotomous response data is within a finite dimensional linear space.

Theorem 2.2 *The solution to (2.4.2) has the form*

$$f_i(t) = \phi^i(t)^T d^i + \xi^i(t)^T c, \tag{2.4.4}$$

where $\{\phi_v^i\}_{v=1}^{M_i}$ *is a set of basis functions spanning the null space* $\mathcal{H}_0^i$, $\phi^i(t)^T = (\phi_1^i(t), \cdots, \phi_{M_i}^i(t))$, $\xi^i(t)^T = (R_{\Theta_i}(x_1, t), \cdots, R_{\Theta_i}(x_n, t))$.

Proof See Wahba (1990).

Substituting (2.4.4) into (2.4.2), we can estimate c^i and d^i by minimizing

$$\begin{aligned} & I_\lambda(c,d) \\ = \quad & -\sum_{j=1}^n l_j(\phi^1(x_j)^T d^1 + \xi^i(x_j)^T c^1, \cdots, \phi^k(x_j)^T d^k \\ & +\xi^k(x_j)^T c^k) + \tfrac{n}{2}\sum_{i=1}^k \lambda_i c^{iT} Q_{\Theta_i} c^i \end{aligned} \tag{2.4.5}$$

where Q_{Θ_i} is an $n \times n$ matrix with entry $Q_{\Theta_i}(l,j) = R_{\Theta_i}(x_l, x_j)$. Since l_j's are not quadratic, (2.4.5) can not be solved explicitly. In the next chapter, we will discuss how to obtain the estimate numerically.

Chapter 3

Fitting the Penalized Polychotomous Regression

3.1 Introduction

As mentioned in Chapter 2, we need to use numerical methods to obtain the solution of the penalized polychotomous regression since a closed form solution can not be obtained. Since we can easily obtain the gradient and Hessian of the penalized negative log likelihood, methods without using the gradient and Hessian will not be considered. Technically, the Newton-Raphson algorithm can be used to obtain the solution because it is a quadratic convergent algorithm. However, the computational complexity of the Newton-Raphson algorithm for this problem will be $O((nk + M_1 + \cdots + M_k)^3)$ since we need to solve a $(nk + M_1 + \cdots + M_k)) \times (nk + M_1 + \cdots + M_k)$ linear system in each iteration. Meanwhile this algorithm requires computer memory on the order of $O((nk + M_1 + \cdots + M_k)^2)$. Usually, $M_1, \cdots, M_k$ are small so nk will decide the computational complexity and the required memory in a given application. The Newton-Raphson algorithm will definitely be desirable when nk is not large. However, nk might be large or very large in lots of applications, and the Newton-Raphson will not be desirable in these situations.

In this chapter, an iterative method called Block one-step SOR-Newton-Raphson is proposed to solve the problem when n is moderate and nk is large. This method is a combination of the SOR method and the Newton-Raphson method. The computational complexity for this method is $O(n^3)$ and the convergence for this method is superlinear. We sacrifice the convergent rate a little bit while reducing the computational complexity dramatically in each iteration. Methods which are designed to solve the problem when n is large will be considered in Chapter 4.

We will first review the Nonlinear SOR method in Section 3.2. In Section 3.3 we discuss the implementation of block one-step SOR-Newton-Raphson method to the penalized polychotomous regression problem. We discuss the method for choosing the smoothing parameters in Section 3.4. Connections between the smoothing spline estimate of the penalize polychotomous regression problem and a Bayesian problem is investigate in section 3.5. Some Monte Carlo simulations are conducted in section 3.6 to illustrate the performance of the smoothing spline estimates.

3.2 Block Nonlinear SOR methods

In this section, we will review some iterative methods to solve a large nonlinear system.

Assume we are concerned with the following nonlinear system

$$\begin{cases} f_1(x_1, \cdots, x_m) &= 0 \\ \quad \vdots & \vdots \quad \vdots \\ f_m(x_1, \cdots, x_m) &= 0. \end{cases}$$

By partitioning the x as $x = (x^1, \cdots, x^p)$, and by grouping, the above nonlinear system will become

$$\begin{cases} F_1(x^1, \cdots, x^p) &= 0 \\ \quad \vdots & \vdots \quad \vdots \\ F_p(x^1, \cdots, x^p) &= 0. \end{cases}$$

The basic step for the block nonlinear SOR is as follows. First, we solve the ith nonlinear system

$$F_i((x^1)^{k+1}, \cdots, (x^{i\perp 1})^{k+1}, x^i, (x^{i+1})^k, \cdots, (x^p)^k) = 0 \tag{3.2.1}$$

for x^i and set $(x^i)^{k+1} = (x^i)^k + \omega(x^i - (x^i)^k)$. In order to obtain $x^{k+1} = ((x^1)^{k+1}, \cdots, (x^p)^{k+1})$ from $x^k = ((x^1)^k, \cdots, (x^p)^k)$, we successively update the block component of x by the above method until all components are updated. The ω in the updating formula is called the relaxation parameter. The process is called block nonlinear Gauss-Seidel method if we set ω equal to 1 in every update. See Ortega and Rheinboldt (1970) for details.

Notice that in the block nonlinear SOR process described above, we still need to solve a nonlinear system in each update. In most applications, we usually don't have a closed form solution for the nonlinear system (3.2.1) and the solution should be obtained by the Newton-Raphson method. In this case, the nonlinear process is called block nonlinear SOR-Newton-Raphson.

Furthermore, if we use one step Newton-Raphson iteration (the value from previous SOR iterate taken as the initial value) to approximate the solution of the nonlinear system (3.2.1), the nonlinear SOR process is called the block one-step SOR Newton-Raphson method accordingly. Specifically, the updating formula for the block one-step SOR-Newton-Raphson is

$$(x^i)^{k+1} = (x^i)^{k+1} - \omega[\partial_i F_i(y^{k,i})]^{\perp 1} F_i(y^{k,i}), \tag{3.2.2}$$

where

$$y^{k,i} = ((x^1)^{k+1}, \ldots, (x^{i\perp 1})^{k+1}, (x^i)^k, \ldots, (x^l)^k).$$

In the statistics literature, the nonlinear system usually arises from a minimization or maximization problem in which we need to find a set of parameters to minimize (or maximize) a function. Specifically, suppose we are going to find $x \in R^m$ to minimize a twice differentiable multivariate function $g(x)$, then the updating formula for the block one-step SOR-Newton-Raphson method will become

$$(x^i)^{k+1} = (x^i)^k - \omega[\nabla^2_{ii} g(y_{k,i})]^{\perp 1} \nabla_i g(y^{k,i}), \tag{3.2.3}$$

where $\nabla^2_{ii} g$ is the submatrix of the Hessian and $\nabla_i g$ is the sub-vector of the gradient.

By putting some conditions on the nonlinear system we are going to solve or the function we are going to minimize, we will have some convergence properties for the general block nonlinear SOR and the block one-step SOR-Newton method. We will state the convergent results which appeared in Ortega and Rheinboldt (1970) in the remain of this section.

Let $F'(x) = D(x) - L(x) - U(x)$ be the decomposition of $F'(x)$ into block diagonal, strictly block lower-triangular and strictly block upper-triangular parts, where

$$D(x) = \begin{pmatrix} \frac{\partial F_1}{\partial x^1} & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \frac{\partial F_p(x)}{\partial x^p} \end{pmatrix}.$$

For $\omega > 0$, let

$$H_\omega(x) = [D(x) - \omega L(x)]^{\perp 1}[(1-\omega)D(x) + \omega U(x)]. \tag{3.2.4}$$

The local convergence of the block nonlinear SOR procedures is stated in the following lemma. The proof of this lemma can be found in Ortega and Rheinboldt (1970).

Lemma 3.3 (Local Convergence and Rate of Convergence) *Assume $F : R^m \to R^m$ be continuously differentiable over a compact set S_0, and $x^* \in S_0$ such that $F(x^*) = 0$. If $D(x^*)$ is nonsingular and $\rho(H_\omega(x^*)) < 1$, then there exists an open ball $S = S(x^*, \delta)$ in S_0 such that for any $x^0 \in S$, both the block nonlinear SOR and the block one-step SOR-Newton sequence converge to x^*, and they share the same convergent factor $R_1(x^k, x^*) = \rho(H_\omega(x^*))$.*

We will state the global convergence result in term of the minimization problem.

Lemma 3.4 (Global Convergence) *Assume $g \in C^2(R^m)$, $\nabla^2 g(x) > 0$ and $S_0 = \{x | g(x) \leq g(x^0)\}$ is bounded, then for suitable chosen relaxation parameter ω, the iterative sequence from the block one-step SOR-Newton method converges to the unique solution x^*.*

The proof of the above lemma can be found in Schechter (1968). From the above lemma, we can see that in general the block one-step SOR-Newton-Raphson method with fixed ω is not guaranteed to converge globally. In practice, we can either change the initial value or tune the relaxation parameter to make the algorithm converge. The following lemma adapted from Varga (1962) can be used to check the conditions for the local convergence.

Lemma 3.5 *Let $A = D - E - E^T$ be a symmetric positive definite matrix, and D is also positive definite. Denote $H_\omega = (D - \omega E)^{\perp 1}((1-\omega)D + \omega E)$. If $D - \omega E$ is nonsingular for $0 \leq \omega \leq 2$, then $\rho(H_\omega) < 1$ for $0 < \omega < 2$.*

By applying the above lemma, we have the following corollary.

Corollary 3.1 *If $A = D - E - E^T$ is symmetric positive define and D is block diagonal matrix, E is strictly block lower triangular matrix. If D is nonsingular, then for $0 < \omega < 2$, we have $\rho(H_\omega) < 1$.*

According to Corollary 3.1, we note that if A is Hessian of a twice differentiable convex function, we will always have $\rho(H_\omega) < 1$ for $0 < \omega < 2$. Specifically, the local convergent property holds if we use block nonlinear Gauss-Seidal or block one step Gauss-Seidal-Newton-Raphson method to find the minimizer of a twice differentiable convex function.

3.3 Implementation of the Algorithm

In this section, we will describe how to apply the block one step SOR-Newton-Raphson method to get the estimate for the penalized polychotomous regression numerically. ω will be taken to be 1 in our implementation.

For polychotomous response data, we have

$$l_j(f_1, \cdots, f_k) = -\sum_{i=1}^{k} y_{ij} f_i(x_j) + log(1 + \sum_{i=1}^{k} e^{f_i(x_j)}), \tag{3.3.1}$$

where $y_{ij} = I[y_j = i]$. Let $u_{ij} = -dl_j/df_{ij}$, $u_i^T = (u_{i1}, \cdots, u_{in})$, $w_{ij} = -d^2 l_j df_{ij}^2$, $W_i = diag(w_{i1}, \cdots, w_{in})$, and $S_i = (\phi^i(x_1), \cdots, \phi^i(x_n))$. Also, with abuse of notation, let $f_{ij} = f_i(x_j)$ and $f_i^T = (f_{i1}, \cdots, f_{in})$. Then we have

$$u_{ij} = -y_{ij} + \frac{e^{\Phi^i(x_j)^T d^i + \xi^i(x_j)^T c^i}}{1 + \sum_{l \neq i} e^{f_l(x_j)} + e^{\Phi^i(x_j)^T d^i + \xi^i(x_j)^T c^i}}, \tag{3.3.2}$$

$$w_{ij} = \frac{e^{\Phi^i(x_j)^T d^i + \xi^i(x_j)^T c^i}}{1 + \sum_{l \neq i} e^{f_l(x_j)} + e^{\Phi^i(x_j)^T d^i + \xi^i(x_j)^T c^i}} \frac{1 + \sum_{l \neq i} e^{f_l(x_j)}}{1 + \sum_{l \neq i} e^{f_l(x_j)} + e^{\Phi^i(x_j)^T d^i + \xi^i(x_j)^T c^i}}, \tag{3.3.3}$$

hence $\partial I_\lambda / \partial c^i = Q_{\Theta_i} u_i + n\lambda_i Q_{\Theta_i} c^i$, $\frac{\partial I_\lambda}{\partial d^i} = S_i^T u_i$, $\partial^2 I_\lambda / \partial c^i \partial {c^i}^T = Q_i W_i Q_i + n\lambda_i Q_i$, $\partial^2 I_\lambda / \partial c^i \partial d^i = Q_i W_i S_i$ and $\partial^2 I_\lambda \partial d^i \partial {d^i}^T = S_i^T W_i S_i$.

The Block one-step SOR Newton-Raphson updating formula for the coefficient (c^i, d^i) becomes

$$\begin{pmatrix} c^i \\ d^i \end{pmatrix} = \begin{pmatrix} c^i_\perp \\ d^i_\perp \end{pmatrix} - \begin{pmatrix} Q_i W_{i\perp} Q_i + n\lambda_i Q_i & Q_i W_{i\perp} S_i \\ S_i^T W_{i\perp} Q_i & S_i^T W_{i\perp} S_i \end{pmatrix}^{\perp 1} \begin{pmatrix} Q_i u_{i\perp} + n\lambda_i Q_i c^i_\perp \\ S_i^T u_i \end{pmatrix}, \tag{3.3.4}$$

where the subscript minus indicates the quantities evaluated at the latest update. By rearranging (3.3.4) we will have the following linear system,

$$\begin{pmatrix} Q_i W_{i\perp} Q_i + n\lambda_i Q_i & Q_i W_{i\perp} S_i \\ S_i^T W_{i\perp} Q_i & S_i^T W_{i\perp} S_i \end{pmatrix} \begin{pmatrix} c^i - c^i_\perp \\ d^i - d^i_\perp \end{pmatrix} = \begin{pmatrix} -Q_i u_{i\perp} - n\lambda_i Q_i c^i_\perp \\ -S_i^T u_{i\perp} \end{pmatrix}. \tag{3.3.5}$$

According to Theorem 1.1, $f_i = S_i d^i + Q_i c^i, i = 1, \cdots, k$ is always unique as long as S_i's are of full rank. If Q_i is nonsingular, (3.3.5) is equivalent to the linear system

$$\begin{pmatrix} W_{i\perp} Q_i + n\lambda_i Q_i & Q_i W_{i\perp} \\ S_i^T & 0 \end{pmatrix} \begin{pmatrix} c^i \\ d^i \end{pmatrix} = \begin{pmatrix} W_{i\perp} f_{i\perp} - u_{i\perp} \\ 0 \end{pmatrix}. \tag{3.3.6}$$

If Q_i is singular, any solution to (3.3.6) is also a solution to (3.3.5). Let

$$\tilde{Q}_i = {W_{i\perp}}^{1/2} Q_i {W_{i\perp}}^{1/2}, \tilde{c}^{(i)} = {W_{i\perp}}^{\perp 1/2} c^i, \tilde{S}_i = {W_{i\perp}}^{1/2} S_i,$$

$$\tilde{d}^i = d^i, \text{ and } \tilde{y}_i = {W_{i\perp}}^{\perp 1/2} (W_{i\perp} f_{i\perp} - u_{i\perp});$$

(3.3.6) can be simplified to

$$\begin{cases} (\tilde{Q}_{\Theta_i} + n\lambda_i)\tilde{c}^i + \tilde{S}^i \tilde{d}^i = \tilde{y}_i \\ \tilde{S}_i^T \tilde{c}^i = 0 \end{cases}. \tag{3.3.7}$$

It can be shown that the solution of the linear system (3.3.7) is equivalent to the solution of the following variational problem, find $\tilde{c}^i, \tilde{d}^i$ to minimize

$$\frac{1}{n} ||\tilde{y}_i - (\tilde{Q}_{\Theta_i} \tilde{c}^i + \tilde{S}_i \tilde{d}^i)||^2 + \lambda_i (\tilde{c}^i)^T \tilde{Q}_{\Theta_i} \tilde{c}^i. \tag{3.3.8}$$

3.4 Choosing the Smoothing Parameters

In Section 3.3, the smoothing parameters $\lambda_{il} = \lambda_i/\theta_{il}$ are fixed. As all $\lambda_{il} \to 0$, f_i follows the data and is very wiggly. It then has small bias but large variance. As all $\lambda_{il} \to \infty$, f_i is forced in the null space $\mathcal{H}_i^0$, which is a parametric fit. It then has large bias but small variance. As the λ_{il}'s vary, we have a family of models. Therefore choosing appropriate smoothing parameters is crucial for effectively estimating the true functions from data by fitting smoothing spline models. Choosing the λ_{il}'s is equivalent to choosing λ_i and $\Theta_i = (\theta_{i1}, \cdots, \theta_{iq_i})$ after imposing an identifiability constraint on λ_i and Θ_i. We call λ_i's the main smoothing parameters and Θ_i's the subsidiary smoothing parameters.

Reconsider the linear system (3.3.7), it is easy to see that the solution of (3.3.6) gives the minimizer of

$$(\tilde{\tilde{y}}_i - f_i)^T W_{i\perp}(\tilde{\tilde{y}}_i - f_i) + \frac{n}{2}\lambda_i \sum_{l=1}^{q_i} \theta_{il}||P_i^l f_i||^2, \tag{3.4.1}$$

where $\tilde{\tilde{y}}_i = f_{i\perp} - W_{i\perp}^{\perp 1} u_{i\perp}$. The one step block SOR-Newton procedure iteratively reformulates the problem of updating each logit function to estimate the function f_i from the pseudo-data by weighted penalized least squares successively. The following lemma shows that the pseudo-data approximately have the usual data structure if $f_{1\perp}, \cdots, f_{k\perp}$ are not far away from $f_1, \cdots, f_k$.

Lemma 3.6 *If $|f_{ij\perp} - f_{ij}| = o(1)$ uniformly in j, and $p_i(t)$ bounded away from 0 and 1, then*

$$\tilde{\tilde{y}}_{ij} = f_{ij} + \epsilon_{ij} + o_p(1)$$

where ϵ_{ij} has mean 0 and variance $w_{ij}^{\perp 1}$, and $\epsilon_{i1}, \cdots, \epsilon_{in}$ are independent.

Proof Let $p_{ij} = p_i(x_j)$. Then $E(y_{ij}) = p_{ij}$, $Var(y_{ij}) = p_{ij}(1 - p_{ij}) = w_{ij}$, $u_{ij} = p_{ij} - y_{ij}$. Hence, we have $E(u_{ij}/w_{ij}) = 0$ and $Var(u_{ij}/w_{ij}) = w_{ij}^{\perp 1}$.
Let

$$\gamma = f_{ij\perp} - u_{ij\perp}/w_{ij\perp} - (f_{ij} - u_{ij}/w_{ij}) = f_{ij\perp} - f_{ij} - (\frac{p_{ij\perp} - y_{ij}}{w_{ij\perp}} - \frac{p_{ij} - y_{ij}}{w_{ij}})$$

Then

$$E(\gamma) = f_{ij\perp} - f_{ij} - \frac{p_{ij\perp} - p_{ij}}{w_{ij\perp}}$$

Since there exists $0 < c_1 < c_2 < 1$ such that $c_1 \le p_i(t) \le c_2$, we have $c_1 \le p_{ij} \le c_2$. From $|f_{ij\perp} - f_{ij}| = o(1)$ uniformly in j, we have $|p_{ij\perp} - p_{ij}| = o(1)$ uniformly in j. Hence, for large n, there exists $0 < c_1^* < c_2^* < 1$ (does not depend on n) such that $c_1^* \le p_{ij\perp} \le c_2^*$. Then, for large n, there exists $0 < d_1 < d_2 < 1$ (does not depend on n) such that $d_1 \le w_{ij\perp} \le d_2$. Hence, $E(\gamma) = o(1)$.
Meanwhile

$$Var(\gamma) = (\frac{1}{w_{ij\perp}} - \frac{1}{w_{ij}})^2 w_{ij} = (w_{ij\perp} - w_{ij})^2 w_{ij}^{\perp 1} w_{ij}^{\perp 1} = o(1).$$

So

$$\tilde{\tilde{y}}_{ij} = f_{ij\perp} - u_{ij\perp}/w_{ij\perp} = f_{ij} - u_{ij}/w_{ij} + \gamma = f_{ij} + \epsilon_{ij} + o_p(1),$$

where $\epsilon_{ij} = -u_{ij}/w_{ij}$ has mean 0 and variance $w_{ij}^{\perp 1}$. The independence of $\epsilon_{i1}, \cdots, \epsilon_{in}$ follows from the independence of $y_{i1}, \cdots, y_{in}$. **Q.E.D.**

From the above discussion, we can use well known methods to select smoothing parameters at each update of the block one step Newton-Raphson procedure. Two of the commonly recognized

data driven methods for choosing smoothing parameters are the *generalized cross validation* (GCV) and the *unbiased risk* methods (Wahba 1990). The GCV method estimates smoothing parameter by minimizing the GCV score

$$V(\lambda_i, \Theta_i) = \frac{1/n||(I - A(\lambda_i, \Theta_i))W_{i\perp}{}^{1/2}\tilde{\tilde{y}}_i||^2}{[(1/n)trace(I - A(\lambda_i, \Theta_i))]^2}$$

and the UBR score

$$U(\lambda_i, \Theta_i) = \frac{1}{n}||(I - A(\lambda_i, \Theta_i)W_{i\perp}{}^{1/2}\tilde{\tilde{y}}||^2 + \frac{2}{n}\sigma^2 trA(\lambda_i, \Theta_i),$$

where $A(\lambda_i, \Theta_i)$ satisfies

$$(w_{i1\perp}{}^{1/2} f_i(t_1), \ldots, w_{i1\perp}{}^{1/2} f_i(t_n))^T = A(\lambda_i, \Theta_i)(w_{i1\perp}{}^{1/2}\tilde{\tilde{y}}_{i1}, \cdots, w_{in\perp}{}^{1/2}\tilde{\tilde{y}}_{in})^T,$$

$f_i(t_j)$'s are computed from the above linear system, $\tilde{\tilde{y}}_i = (\tilde{\tilde{y}}_{i1}, \ldots, \tilde{\tilde{y}}_{in})^T$, and $\tilde{\tilde{y}}_{ij} = f_{ij\perp} - u_{ij\perp}/w_{ij\perp}$. When using the UBR method, we use $\sigma^2 = 1$. A generic code RKPACK (Gu, 1989) can be used to solve the linear system in each update and estimate λ and Θ via GCV or the UBR method at the same time. The whole iterative process will stop when the relative weighted mean square error is less than a threshold. However, since changing λ and Θ at each update means modifying the problem successively, convergence is not guaranteed.

3.5 Bayesian Inference

We will first extend Gaussian posterior calculations to the case where the responses are vector. Let

$$F_\xi^i(x) = \sum_{v=1}^{m_i} \tau_{v,i}\phi_{v,i}(x) + b_i^{1/2} \sum_{\beta=1}^{q_i} \sqrt{\theta_{\beta,i}} Z_{\beta,i}(x)$$

where $\tau = (\tau_{1,1}, \ldots, \tau_{m_1,1}, \ldots, \tau_{m_k,k})^T \sim N(0, \xi I)$, $Z_{\beta,i}$ are i.i.d, zero mean Gaussian stochastic processes, independent of τ, with $E[Z_{\beta,i}(s)Z_{\beta,i}(t)] = R_{\beta,i}(s,t)$
Let

$$Z^i(x) = \sum_{\beta=1}^{q_i} \sqrt{\theta_{\beta,i}} Z_{\beta,i}(x),$$

then

$$E[Z^i(s)Z^i(t)] = R_i(s,t)$$

where $R_i(s,t) = \sum_{\beta=1}^{q_i} \theta_{\beta,i} R_{\beta,i}(s,t)$. Suppose observations have the form

$$y_{ij} = F_\xi^i(x_j) + \epsilon_{ij}, \qquad i = 1, \cdots, k \text{ and } j = 1, \cdots, n$$

where $\epsilon = (\epsilon_{11}, \ldots, \epsilon_{kn}) \sim N(0, \sigma^2 W^{\perp 1})$, with W positive definite and known. Let $n\lambda_i = \sigma^2/b_i$, $f_\lambda(x) = (f_\lambda^1(x), \ldots, f_\lambda^k(x))^T$ and $F_\xi(x) = (F_\xi^1(x), \ldots, F_\xi^k(x))^T$, we have

$$f_\lambda(x) = \lim_{\xi \to \infty} E(F_\xi(x)|y),$$

where f_λ is the minimizer of the penalized weighted least square problem

$$(y - f)^T W (y - f) + n \sum_{i=1}^{k} \lambda_i \sum_{\beta=1}^{q_i} \theta_{\beta,i}^{\perp 1} \| P_\beta f^i \|^2. \tag{3.5.1}$$

Denote

$$Q = \begin{pmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & \dots & \vdots \\ \vdots & \dots & \ddots & \\ 0 & & & Q_k \end{pmatrix}, \quad S = \begin{pmatrix} S_1 & 0 & \dots & 0 \\ 0 & S_2 & \dots & \vdots \\ \vdots & \dots & \ddots & \\ 0 & & & S_K \end{pmatrix},$$

and

$$M = Q + n \begin{pmatrix} \lambda_1 I_{n\times n} & 0 & \dots & 0 \\ 0 & \lambda_2 I_{n\times n} & \dots & \vdots \\ \vdots & \dots & \ddots & \vdots \\ 0 & \dots & \dots & \lambda_k I_{n\times n} \end{pmatrix} W^{\perp 1}$$

where $(Q_i)_{uv} = R_i(x_u, x_v)$ and $(S_i)_{uv} = \phi_{v,i}(x_u)$. Similar to Wang (1994), We have the following theorem.

Theorem 3.3 *Let* $g^i_{0,v}(x) = \tau_{v,i}\phi_{v,i}(x)$ *and* $g^i_\beta(x) = b_i^{1/2}\sqrt{\theta_{\beta,i}}Z_{\beta,i}(x)$. *Then*

$$\begin{array}{lcl} E(g^i_{0,v}(x)|\mathbf{y}) & = & d^i_v\phi_{v,i}(x) \\ E(g^i_\beta(x)|\mathbf{y}) & = & \sum_{j=1}^n c^i_j\theta_{\beta,i}R_{\beta,i}(x,x_j) \\ \frac{1}{b_i^{1/2}b_j^{1/2}}Cov(g^i_{0,v}(s), g^j_{0,u}(t)|\mathbf{y}) & = & \phi_{v,i}(s)\phi_{u,i}(t)e^T_{v,i}(S^TM^{\perp 1}S)^{\perp 1}e_{u,i} \\ \frac{1}{b_i^{1/2}b_j^{1/2}}Cov(g^i_\beta(s), g^j_{0,v}(t)|\mathbf{y}) & = & -d_{v,\beta,i}(s)\phi_{v,j}(t) \\ \frac{1}{b_i}Cov(g^i_\beta(s), g^i_\beta(t)|\mathbf{y}) & = & \theta_{\beta,i}R_{\beta,i}(s,t) - \sum_{j=1}^n c_{j,\beta,i}(s)\theta_{\beta,i}R_{\beta,i}(t,x_j) \\ \frac{1}{b_l^{1/2}b_i^{1/2}}Cov(g^l_\gamma(s), g^i_\beta(t)|\mathbf{y}) & = & -\sum_{j=1}^n c_{j,\gamma,l}(s)\theta_{\beta,i}R_{\beta,i}(t,x_j) \end{array}$$

where $e_{v,i}$ is the $((i-1)n)+v)th$ unit vector, $d_{\beta,i}(s)^T = (d_{1,\beta,i}(s), \cdots, d_{M_i,\beta,i}(s))$ and $c_{\beta,i}(s)^T = (c_{1,\beta,i}(s), \cdots, c_{n,\beta,i}(s))$ are given by

$$\begin{pmatrix} * \\ \vdots \\ * \\ d_{\beta,i}(s) \\ * \\ \vdots \\ * \end{pmatrix} = (S^TM^{\perp 1}S)^{\perp 1}S^TM^{\perp 1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \theta_{\beta,i}R_{\beta,i}(s,x_1) \\ \vdots \\ \theta_{\beta,i}R_{\beta,i}(s,x_n) \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} * \\ \vdots \\ * \\ c_{\beta,i}(s) \\ * \\ \vdots \\ * \end{pmatrix} = [M^{\perp 1} - M^{\perp 1}S(S^TM^{\perp 1}S)^{\perp 1}S^TM^{\perp 1}] \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \theta_{\beta,i}R_{\beta,i}(s,x_1) \\ \vdots \\ \theta_{\beta,i}R_{\beta,i}(s,x_n) \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Next, we can use Laplace method to approximate the posterior distribution based on polychotomous data ($k+1$ categories). With abuse of notation we will denote

$$y = (y_{11}, \cdots, y_{1n}, \cdots, y_{k1}, \cdots, y_{kn})^T.$$

Suppose the priors for the logit functions are $f_1(x) \sim F_\xi^1(x), \cdots, f_k(x) \sim F_\xi^k(x)$. Let ζ, η be any one of $\tau_{v,i}\phi_{v,i}(x)$, $\tau_{u,j}\phi_{u,j}(z)$, $\sqrt{\theta_{\beta,i}}Z_{\beta,i}(x)$ or $\sqrt{\theta_{\alpha,j}}Z_{\alpha,j}(z)$ for arbitrary points x and z. Denoting

$$f^T = (f_1(x_1), \ldots, f_1(x_n), \ldots, f_k(x_1), \cdots, f_k(x_n)),$$

the sampling distribution of y given f is proportional to $exp\{-L_y(f)\}$. Letting $\xi \to \infty$, we have the posterior distribution

$$\pi(\zeta,\eta)|y) \propto \int p(f|y)q(f)r(\zeta,\eta|f)df,$$

where $p(f|y) \propto exp\{-L_y(f)\}$,

$$q(f) \propto exp\{-\frac{1}{2b}f^T(Q^{\perp 1} - Q^{\perp 1}S(S^TQ^{\perp 1}S)^{\perp 1}SQ^{\perp 1})f\}$$

and $r(\zeta,\eta|f)$ is Gaussian with mean and variance given in Theorem 3.3 with $\sigma^2 = 0$ and $y = f$.

Denoting $\hat{p}(f|y)$ be the approximation using Taylor expansion centered at the mode f_* of $p(f|y)q(f)$, and approximating $\pi(\zeta,\eta|y)$ by

$$\hat{\pi}(\zeta,\eta)|y) \propto \int \hat{p}(f|y)q(f)r(\zeta,\eta|f)df,$$

we have the following theorem. The proof of this theorem is the same as in Gu (1992).

Theorem 3.4 *The approximate posterior density $\hat{\pi}(\zeta,\eta|y)$ is Gaussian with mean and covariance given in Theorem 3.3.*

Based on the above result, we can construct approximate Bayesian confidence interval for each component, each logit function and the difference of logit $f_i - f_j$.

To apply the results derived above, it is necessary to compute the quantities involved. From Theorem 3.3, we can see that the computation focuses on the computing of $(S^TM^{\perp 1}S)^{\perp 1}$, $c_{\beta,i}(s)$ and $d_{\beta,i}(s)$. We will discuss the calculation of these quantities in the following.

It can be shown that the solution of the variational problem (3.5.1) has the expression as those in Theorem 1.2, where $c^T = (c_1^T, \cdots, c_k^T)^T = M^{\perp 1}(I - S(S^TM^{\perp 1}S)^{\perp 1}S^TM^{\perp 1})y$ and $d^T =$

$(d_1^T, \cdots, d_k^T)^T = (S^T M^{\perp 1} S)^{\perp 1} S^T M^{\perp 1} y$. c and d can be calculated by backfitting algorithm. By replacing y with

$$(\theta_{\beta,1} R_{\beta,1}(s, x_1), \cdots, \theta_{\beta,1} R_{\beta,1}(s, x_n), \cdots, \theta_{\beta,k} R_{\beta,k}(s, x_1), \cdots, \theta_{\beta,k} R_{\beta,k}(s, x_n))^T,$$

we can use backfitting to get $c_{\beta,i}(s)$ and $d_{\beta,i}(s)$.

To calculate $(S^T M^{\perp 1} S)$, we will first compute $M^{\perp 1} a$ for a given vector a. Denoting $z = M^{\perp 1} a$, we can obtain z by solving the linear system $Mz = a$. Again, we can use Gauss-Seidal (linear SOR with $\omega = 1$) method to solve this linear system.

3.6 Monte Carlo Examples

In this section, we conduct several simulations to evaluate the performance of the proposed method. The comparative Kullback-Leibler distance (CKL) will be used to measure the performance. The sampling CKL between the estimate and the true probabilities for polychotomous response data is

$$CKL(p, \hat{p}) = -\frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{k} p_i(x_j) log \hat{p}_i(x_j)$$

where $p(x) = (p_0(x), \cdots, p_k(x))^T$ and x_j's are design points.

The first example is for univariate case. The domain and range are taken as $\mathcal{X} = [0, 1]$ and $\mathcal{Y} = \{0, 1, 2\}$. The conditional class probabilities are taken as $p_1(x) = e^{f_1(x)}/(1 + e^{f_1(x)} + e^{f_2(x)})$, $p_2(x) = e^{f_2(x)}/(1 + e^{f_1(x)} + e^{f_2(x)})$ and $p_3(x) = 1 - p_1(x) - p_2(x)$, where $f_1(x) = 0.3e^{x^2} + 0.4cos(2.7x)$ and $f_2(x) = x^2 + 2cos(3x)$. Two different sample sizes are used: $n = 200$ and $n = 500$. We generated the design points x_i from an uniform distribution on [0, 1] and generated the polychotomous responses using the underlying functions. Designs and responses are generated for 200 replicates for each simulation. The penalized likelihood for this example is

$$\sum_{j=1}^{n} (-\sum_{i=1}^{2} y_{ij} f_i(x_j) + log(1 + e^{f_1(x_j)} + e^{f_2(x_j)})) + \frac{n\lambda_1}{2} \int_0^1 (f_1''(t))^2 dt + \frac{n\lambda_2}{2} \int_0^1 (f_2''(t))^2 dt.$$

The algorithm proposed in this chapter is used to get the smoothing spline estimate for each simulated data set. We select the 5th, 50th and 95th percentile best estimates ordered by CKL. Their probability estimates are plotted in Figure 1.

The second example is for the multivariate case. Here we present an example which has three-category response and two predictors. The domain and range are taken as $\mathcal{X} = [0, 1] \times [0, 1]$ and $\mathcal{Y} = \{0, 1, 2\}$. The sample size in this experiment is 500. The covariates are generated uniformly from $[0, 1] \times [0, 1]$. The logit functions are taken as

$$f_1(x_1, x_2) = 3.5e^{\perp(2.0(x \perp 0.5)^2 + 8.0(y \perp 0.8)^2)} + 1.5e^{\perp((x+y \perp 0.4)^2 + 15.0*(x \perp y)^2)} - 1.5$$

and

$$f_2(x_1, x_2) = 2.0(x - y)^2 - 0.4(x + y)^2.$$

The polychotomous responses are generated using the above logit functions. Responses are generated for 100 replicates. The following functional ANOVA decomposition is used,

$$f_1(x_1, x_2) = const + h_1(x_1) + h_2(x_2) + h_{12}(x_1, x_2).$$

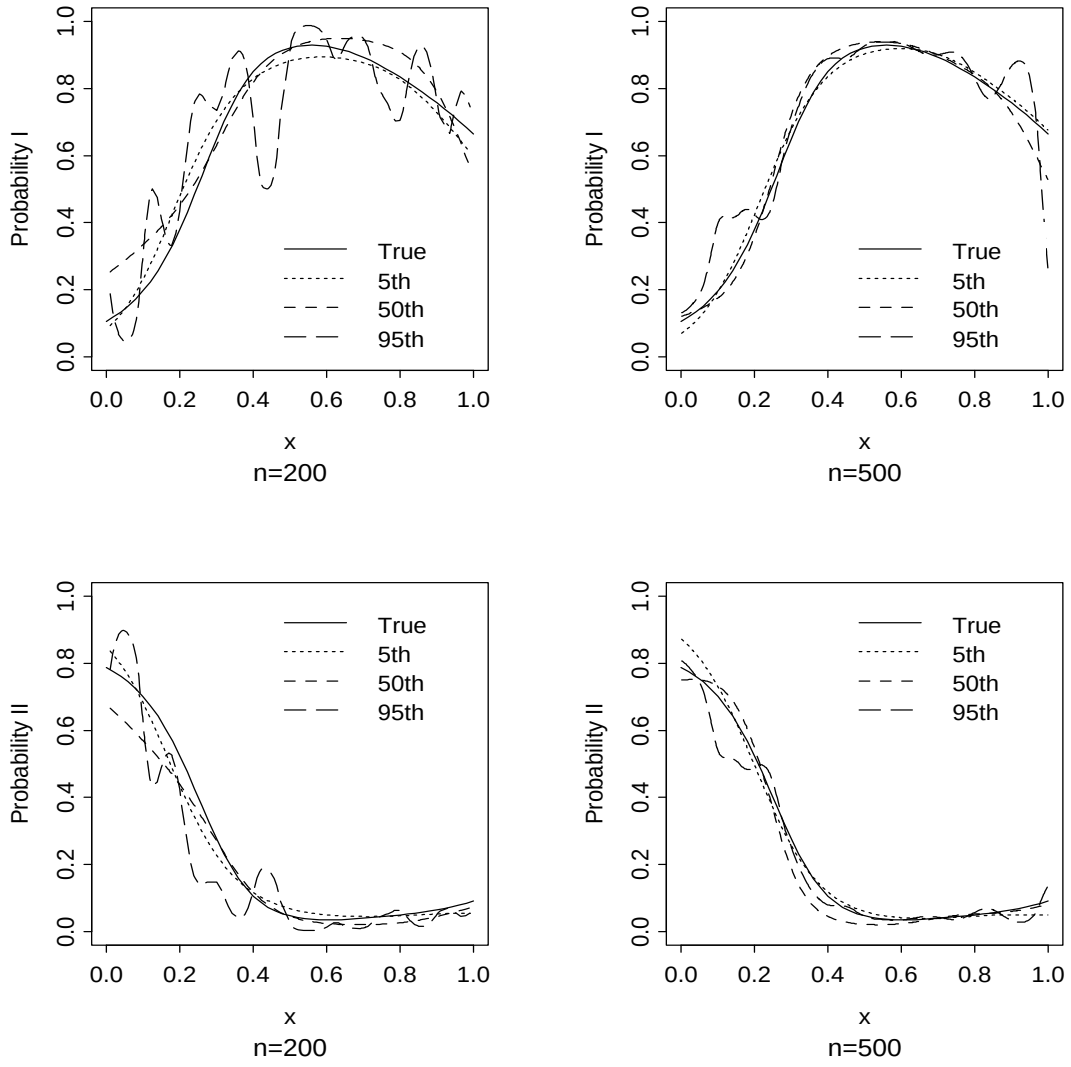


Figure 1: Estimates of $P(Y = 1|X = t)$ (Probability I) and $P(Y = 2|X = t)$ (Probability II). Solid lines are the true functions. Three dashed lines in each graph are the 5th, 50th and 95th percentile best estimates ordered by comparative Kullback-Leibler distance among the 200 simulations.

The form of decomposition for f_2 is the same. We select the 5th, 50th, and 95th percentile best estimates ordered by CKL. Their probability estimates are plotted in Figure 2.

From Figure 1 and Figure 2, we can see that the penalized polychotomous regression can capture the shape of an underlying model and produce a good estimate most of the time. We experience similar conclusions for all the other examples we did.

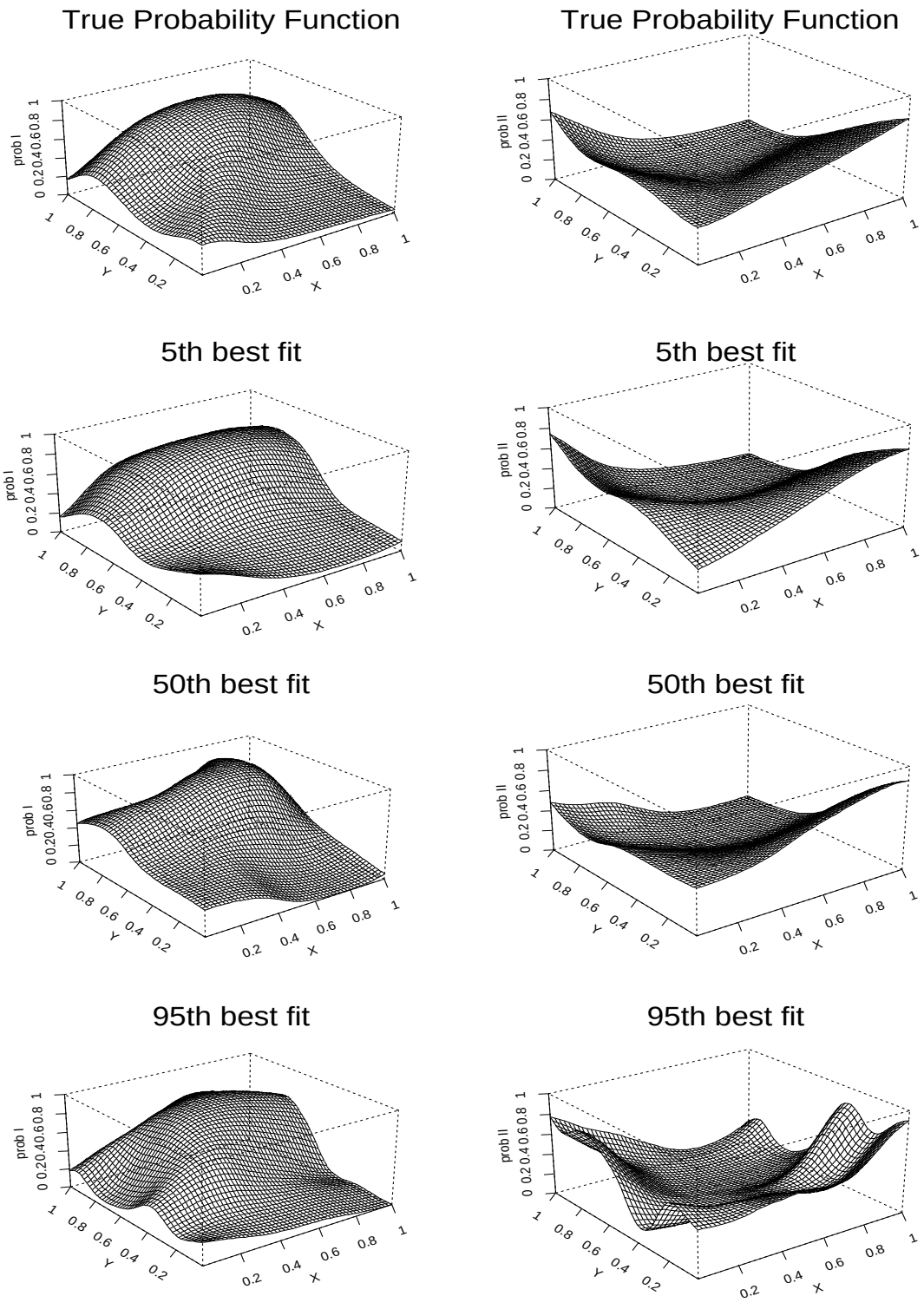


Figure 2: The true probability functions and their estimates.

Chapter 4

Strategies for Large Data Sets

As mentioned in Chapter 3, the algorithm proposed there is not desirable when n (number of observations) is very large. In this chapter, we will discuss some strategies on how to apply the smoothing spline to model large data sets with polychotomous responses.

4.1 Binary Case

When k=1, the polychotomous response data reduces to binary data. In this case, the algorithm proposed in Chapter 3 will reduce to the iterated UBR method proposed in Wahba *et al.* (1995). Although it has been successfully applied in practice, it can not be used to deal with large data sets due to its computational capacity. For large data sets, we will first propose a randomized version of generalized approximate cross validation to choose the smoothing parameters for binary data. At the same time, strategies to obtain an approximate smoothing spline are also discussed. Combining these two schemes, we can apply the smoothing spline ANOVA to a very large data set with binary responses. We also construct a Bayesian Confidence Interval for the approximate smoothing spline. The performance of the proposed method compared with the Iterated UBR method will be studied through Monte Carlo Simulations.

4.1.1 Generalized Approximate Cross Validation

Although it appears that the algorithms based on UBR generally converges, it is not guaranteed to do so, since changing λ along the iteration also changes the optimization problem. Based on this consideration, Xiang and Wahba (1996) began with a leaving-out-one or ordinary cross validation (OCV) estimate of $CKL(\lambda)$, namely

$$OCV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[-y_i f_\lambda^{\perp i}(x_i) + b(f_\lambda(x_i))]$$

where $f_\lambda^{\perp i}$ is the fit of f_λ based on leaving out the ith data point. Computing $f_\lambda^{\perp i}$ repeatedly for large n is out of the question, they have obtained a generalized approximate cross validation (GACV) by a series of approximations, including one similar to that used in obtaining GCV (wahba 1990). The result is

$$GACV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[-y_i f_\lambda(x_i) + b(f_\lambda(x_i))] + \frac{Tr(A)}{n}\sum_{i=1}^{n}\frac{y_i(y_i - \mu_\lambda(x_i))}{n - Tr(W^{1/2}AW^{1/2})} \tag{4.1.1}$$

where A is the $n \times n$ inverse Hessian of the penalized likelihood I_λ with respect to the component of $f = (f_\lambda(x_1, \cdots, f_\lambda(x_n))'$, $\mu_\lambda(x_i) = b'(f_\lambda(x_i)$ and W is diagonal matrix with diagonal entries $w_i = b''(f_\lambda(x_n)), i = 1, \cdots, n$. It can be shown that $A = [W(f_\lambda) + n\Sigma_\lambda]^{\perp 1}$, where Σ_λ satisfies $f^T\Sigma_\lambda f = c^T Q_\lambda c$. It can be seen that Σ_λ is not of full rank, and in general its direct computation will be a numerically unstable.

4.1.2 RGACV and One-Step-RGACV

Let A be a symmetric, non-negative definite matrix, and $\epsilon = (\epsilon_1, \cdots, \epsilon_n)'$, where the ϵ are independent, identically distributed normal variables with common variance σ^2. It is easy to see that the expected value of $\frac{1}{\sigma^2}\epsilon' A\epsilon$ is $Tr(A)$. $\frac{1}{\sigma^2}\epsilon' A\epsilon$ is called a randomized trace estimate of $Tr(A)$. In the Gaussian case, Girard (1991) has shown that using a randomized trace estimate as part of the evaluation of the GCV function gives a negligibly different estimate of smoothing parameters from an exact calculation of the GCV function, for large n. See also Wahba, Johnson, Gao, and Gong (1995).

In the calculation of the GACV function, we need to compute $Tr(A)$ and $Tr(W^{1/2}AW^{1/2})$. The direct computation would involve the inverse of a large matrix which requires computational complexity of $O(n^3)$. Meanwhile, as indicated in the last section, in general the direct computation of A will be numerically unstable for large n. Based on these considerations,a method which does not require the direct computation is highly desired. Similar to the randomized trace estimate in Gaussian case, we can use randomized trace estimate of $Tr(A)$ and $Tr(W^{1/2}AW^{1/2})$ as part of the evaluation of the GACV function.

Considering the disturbance $\epsilon \sim N(\mathbf{0}, \sigma^2 I_n)$, we have $E(\epsilon^T A\epsilon) = \sigma^2 Tr(A)$, $E(\epsilon^T WA\epsilon) = \sigma^2 Tr(WA) = \sigma^2 Tr(W^{1/2}AW^{1/2})$. Hence, we can use $\epsilon^T A\epsilon/\sigma^2$ to estimate $Tr(A)$ and $\epsilon^T WA\epsilon/\sigma^2$ to estimate $Tr(W^{1/2}AW^{1/2})$.

Notice that the penalized log likelihood can be rewritten as follows,

$$I_\lambda(f, Y) = -\sum_{i=1}^{n} l_j(y_j, f(x_j)) + \frac{n}{2} f' \Sigma_\lambda f \tag{4.1.2}$$

where $l_j(y_j, f(x_j)) = y_j f(x_j) - b(f(x_j))$ is the log likelihood, $f = (f_1, \cdots, f_n)^T$ and $f_j = f(x_j)$. For a fixed λ, let $\hat{f}_\lambda^Y$ be the minimizer of (4.1.2) with respect to data Y. If we put a small disturbance on Y to get a new pseudo data set, $Y + \epsilon$, we expect $\hat{f}_\lambda^{Y+\epsilon}$ and $\hat{f}_\lambda^Y$ to be very close according to the following lemma.

Lemma 4.7 *For a fixed λ, the minimizer $\hat{\mathbf{f}}_\lambda^{\mathbf{Y}}$ of (4.1.2) is continuous in Y.*

Proof Let $Y \to Y_0$ and denote $\mathbf{f}$ and $\mathbf{f_0}$ to be their corresponding solutions. Hence,

$$-Y + b'(\mathbf{f}) + n\Sigma_\lambda \mathbf{f} = 0 \quad \text{and} \quad -Y_0 + b'(\mathbf{f_0}) + n\Sigma_\lambda \mathbf{f_0} = 0.$$

Subtract the second equation from the first one we get,

$$b'(\mathbf{f}) - b'(\mathbf{f_0}) + n\Sigma_\lambda(\mathbf{f} - \mathbf{f_0}) = Y - Y_0.$$

For fixed Y_0, if Y is within a small neighborhood of Y_0, then $\mathbf{f}$ is bounded. For any sequence of $Y \to Y_0$ within a small neighborhood of Y_0, denote $\mathbf{f}^*$ to be one attraction point of the corresponding sequence of $\mathbf{f}$. Then we have,

$$b'(\mathbf{f}^*) - b'(\mathbf{f_0}) + n\Sigma_\lambda(\mathbf{f}^* - \mathbf{f_0}) = 0.$$

The solution to the above nonlinear system is unique by observing that the derivative of the left side with respect to $\mathbf{f}^*$ is a positive definite matrix. As a result, we have $\mathbf{f}^* = \mathbf{f_0}$. So the bounded sequence of $\mathbf{f}$ converges to $\mathbf{f_0}$ since it has one unique attraction point $\mathbf{f_0}$. Hence $\mathbf{f} \to \mathbf{f_0}$ as

$Y \to Y_0$.**Q.E.D.**
Since $\hat{f}_\lambda^{y+\epsilon}$ and $\hat{f}_\lambda^{y}$ are minimizers of (4.1.2), we have

$$\frac{\partial I_\lambda}{\partial f}(\hat{f}_\lambda^{y+\epsilon}, Y+\epsilon) = 0 \quad \text{and} \quad \frac{\partial I_\lambda}{\partial f}(\hat{f}_\lambda^{y}, Y) = 0.$$

Using a Taylor expansion to expand $\frac{\partial I_\lambda}{\partial f}(\hat{f}_\lambda^{y+\epsilon}, Y+\epsilon)$ at $(\hat{f}_\lambda^{y}, Y)$ we have the following equation,

$$\begin{array}{ll} \frac{\partial I_\lambda}{\partial f}(\hat{f}_\lambda^{y+\epsilon}, Y+\epsilon) & = \frac{\partial I_\lambda}{\partial f}(\hat{f}_\lambda^{y}, Y) + \frac{\partial^2 I_\lambda}{\partial f^T \partial f}(f^*, Y^*)(\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y}) \\ & + \frac{\partial^2 I_\lambda}{\partial Y^T \partial f}(\hat{f}^*, Y^*)(Y+\epsilon - Y) \end{array} \tag{4.1.3}$$

where (f^*, Y^*) is some point between $(\hat{f}_\lambda^{y}, Y)$ and $(\hat{f}_\lambda^{y+\epsilon}, Y+\epsilon)$.
Notice that

$$\frac{\partial^2 I_\lambda}{\partial f^T \partial f} = W(f) + n\Sigma_\lambda \qquad \text{and} \qquad \frac{\partial^2 I_\lambda}{\partial Y^T \partial f} = -I_{n\times n}.$$

Therefore, from (4.1.3) we get

$$\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y} = (W(\hat{f}_\lambda^{*}) + n\Sigma_\lambda)^{\perp 1}\epsilon.$$

When ϵ is very small, $(\hat{f}_\lambda^{y+\epsilon}, Y+\epsilon)$ is very close to $(\hat{f}_\lambda^{y}, Y)$. Hence $(\hat{f}^*, Y^*)$ is very close to $(\hat{f}_\lambda^{y}, Y)$. $W(\hat{f}^*)$ can be approximated by $W(\hat{f}_\lambda^{y})$ and we will have

$$\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y} \approx (W(\hat{f}_\lambda^{y}) + n\Sigma_\lambda)^{\perp 1}\epsilon = A\epsilon. \tag{4.1.4}$$

This gives us the following lemma.

Lemma 4.8 $\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y} = A\epsilon + o(|\epsilon|)$.

When σ is small hence ϵ will be small, we can use $\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y}$ to approximate $A\epsilon$ by lemma 4.8. Thus, we have

$$\epsilon^T A\epsilon = \epsilon^T(A\epsilon) \approx \epsilon^T(\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y}).$$

Thus,

$$Tr(A) \approx \epsilon^T(\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y})/\sigma^2$$

and

$$Tr(W^{1/2}AW^{1/2}) = Tr(WA) \approx \epsilon^T W(\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y})/\sigma^2.$$

By replacing $Tr(A)$ and $Tr(W^{1/2}AW^{1/2})$ in (4.1.1) with their estimates and use $\epsilon^T\epsilon/n$ to estimate σ^2, we have a randomized version of the GACV function,

$$RGACV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[-y_i f_\lambda(x_i) + b(f_\lambda(x_i))] + \frac{\epsilon^T(\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y})}{n}\frac{\sum_{i=1}^{n} y_i(y_i - \mu_\lambda(x_i))}{\epsilon^T\epsilon - \epsilon^T W(\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y})}. \tag{4.1.5}$$

Thus, we replace the computation of a large matrix inverse problem with a iterative procedure similar to that used to get the estimate $\hat{f}_\lambda^{y}$. Suppose we just use one replicate of perturbation, we need about twice the time needed to get the $\hat{f}_\lambda^{y}$ to evaluate the RGACV. If the time to get one estimate is expensive, it still requires a lot of computing time. Besides, we need to choose the σ very carefully so that $A\epsilon$ can be well approximated by $\hat{f}_\lambda^{y+\epsilon} - \hat{f}_\lambda^{y}$. This motivates us to look for an alternative way to calculate $A\epsilon$.

Consider the Newton Raphson procedure when we solving the nonlinear system for the perturbed data $Y+\epsilon$. If we take the solution $\hat{f}_\lambda^Y$ to the nonlinear system for the original data Y as the initial value for a Newton-Raphson calculation of $\hat{f}_\lambda^{y+\epsilon}$ things become even simpler. Letting $\hat{f}_\lambda^{y+\epsilon,1}$ be the result of one step in a Newton-Raphson iteration gives

$$\hat{f}_\lambda^{y+\epsilon,1} = \hat{f}_\lambda^y - (\frac{\partial^2 I_\lambda}{\partial f^T \partial f}(\hat{f}_\lambda^y, Y+\epsilon))^{\perp 1}\frac{\partial I_\lambda}{\partial f}(\hat{f}_\lambda^y, Y+\epsilon). \tag{4.1.6}$$

Notice that

$$\frac{\partial I_\lambda}{\partial f}(\hat{f}_\lambda^y, Y+\epsilon) = -\epsilon + \frac{\partial I_\lambda}{\partial f}(\hat{f}_\lambda^y, Y) = -\epsilon$$

and

$$[\frac{\partial^2 I_\lambda}{\partial f^T \partial f}(\hat{f}_\lambda^y, Y+\epsilon)]^{\perp 1} = [\frac{\partial^2 I_\lambda}{\partial f^T \partial f}(\hat{f}_\lambda^y, Y)]^{\perp 1}.$$

Thus,

$$\hat{f}_\lambda^{y+\epsilon,1} = \hat{f}_\lambda^y + [\frac{\partial^2 I_\lambda}{\partial f^T \partial f}(\hat{f}_\lambda^y, Y)]^{\perp 1}\epsilon.$$

Hence we have

Lemma 4.9

$$\hat{f}_\lambda^{y+\epsilon,1} - \hat{f}_\lambda^y = A\epsilon. \tag{4.1.7}$$

Hence we have the following one step randomized generalized approximate cross-validation formula

$$\begin{aligned} OneStepRGACV(\lambda) = \quad & \frac{1}{n}\sum_{i=1}^n [-y_i f_\lambda(x_i) + b(f_\lambda(x_i))] \\ & + \frac{\epsilon^T(\hat{f}_\lambda^{y+\epsilon,1} \perp \hat{f}_\lambda^y)}{n} \frac{\sum_{i=1}^n y_i(y_i \perp \mu_\lambda(x_i))}{\epsilon^T \epsilon \perp \epsilon^T W(\hat{f}_\lambda^{y+\epsilon,1} \perp \hat{f}_\lambda^y)}. \end{aligned} \tag{4.1.8}$$

To reduce the variance in the term after the second '+' sign in (4.1.8), we may draw R independent replicate vectors $\epsilon_1, \cdots, \epsilon_R$, and replace the term after the '+' in (4.1.8) by

$$\frac{1}{R}\sum_{r=1}^R \frac{\epsilon_r^T(\hat{f}_\lambda^{y+\epsilon_r,1} - \hat{f}_\lambda^y)}{n} \times \frac{\sum_{i=1}^n y_i(y_i - \mu_\lambda(x_i))}{\epsilon_r^T\epsilon_r - \epsilon_r^T W(\hat{f}_\lambda^{y+\epsilon_r,1} - \hat{f}_\lambda^y)}$$

to obtain a R-replicate version of OneStepRGACV.

We summarize the One-Step Randomized Generalized Cross Validation in the following algorithm:

Algorithm: One-Step Randomized GACV algorithm

1. For fixed λ,
   - Based on the original data set Y, we iterate the Newton-Raphson algorithm till it converges to get $\hat{f}_\lambda^y$.
   - Generate perturbation $\epsilon \sim N(\mathbf{0}, \sigma^2 I_n)$, add it to the data set Y to get the pseudo data set $Y+\epsilon$.
   - Take $\hat{f}_\lambda^{y+\epsilon,0} = \hat{f}_\lambda^y$ as initial values, calculate the first Newton-Raphson step based on the nonlinear system $\frac{\partial I_\lambda}{\partial f}(f, Y+\epsilon) = 0$, call it $\hat{f}_\lambda^{y+\epsilon,1}$
   - Take $\hat{f}_\lambda^{y+\epsilon} = \hat{f}_\lambda^{y+\epsilon,1}$ and apply formula (4.1.8) to evaluate OneStepRGACV value.
2. Find λ to minimize $OneStepRGACV(\lambda)$, call it $\hat{\lambda}$.
3. $\hat{f} = \hat{f}_{\hat{\lambda}}^y$ is our final estimate for f.

The performance of using this new criteria to choose the smoothing parameters compared with the iterated UBR method will be studied through Monte Carlo simulations later in this chapter.

4.1.3 Approximate Smoothing Spline

For large n, and the 'true' function f not too 'complex', it can be seen that f_λ of (2.4.4) should be well approximated in the span of a much smaller subset of the ξ_i. See Wahba (1980) and Xiang (1996). Suppose the number of basis function used is k, and denote the basis functions as $\xi_{i_1}, \xi_{i_2}, \cdots, \xi_{i_k}$ which is a subset chosen from $\xi_1, \xi_2, \cdots, \xi_n$. Typically for medical risk data, k may be at most a few hundred even with n very large. In order to get a good approximation, we need the chosen k basis functions to have the least possible correlation. The closer the design points, the closer the corresponding basis functions. As a result, if we choose the design points having maximum separation, their corresponding basis functions should be expected to have least correlation. Considering this problem from another point of view, what we want is to group points into k groups with those groups spaced as far as possible from each other. Thus, the classical cluster analysis can be used to choose the representative design points, i.e, we cluster the n data points into k groups, take one representative point from each group to form the k basis functions. For clustering the data, we will use the FASTCLUS procedure in SAS, which is designed for the disjoint clustering of very large data sets in minimal time. With each cluster, we select the design point closest to the cluster center as the representative point to be included in our subset. Assume $x_{i_j}, j = 1, \cdots, k$ are the selected points, then their corresponding basis functions will be $\xi_{i_j}(x) = Q_\lambda(x_{i_j}, x)$. We approximate f_λ as

$$f_\lambda(x) = \sum_{v=1}^{M} d_v \phi_v(x) + \sum_{j=1}^{k} c_j \xi_{i_j}(x). \tag{4.1.9}$$

Denote X_λ^k to be the $n \times k$ matrix with ijth entry $Q_\lambda(x_i, x_{i_j})$ and Q_λ^k to be the $k \times k$ matrix with ijth entry $Q_\lambda(x_{i_i}, x_{i_j})$, and let R_β^k be the $k \times k$ matrix with ijth entry $R_\beta(x_{i_i}, x_{i_j})$. It is easy to see that for f_λ of the form (4.1.9) we have $\|P^\beta f_\lambda\|^2 = c' R_\beta^k c$. Then the Newton-Raphson update for finding the minimizer $c = (c_1, \cdots, c_k)'$ and d is equivalent to solving the following linear system

$$\begin{pmatrix} (X_\lambda^k)' W_\perp X_\lambda^k + n Q_\lambda^k & (X_\lambda^k)' W_\perp S \\ S' W_\perp X_\lambda^k & S' W_\perp S \end{pmatrix} \begin{pmatrix} c - c_\perp \\ d - d_\perp \end{pmatrix} = \begin{pmatrix} -(X_\lambda^k)' u_\perp - n Q_\lambda^k c_\perp \\ -S' u_\perp \end{pmatrix}, \tag{4.1.10}$$

It is highly possible that the coefficient matrix of the linear system (4.1.10) would be computationally singular even if it is nonsingular in theory. In order to get a stable solution, QR factorization with pivoting is used. Also, when we solve the linear system using the QR decomposition, we select a cutoff parameter τ (such as the machine precision times the largest absolute diagonal element of the R matrix). Whenever $|r_{ii}| \le \tau$ (where r_{ii} denotes the diagonal element of the R matrix in the QR decomposition), the corresponding solution is set to be zero.

In practice, the following procedure can be used to get an approximate solution for large data sets.

Guideline to decide the number of bases:

(I) Decide the number of basis functions to start with.

(II) For a fixed number k, use Cluster method to cluster the data into k groups. A representative point is chosen from each group to form the basis functions. Solve the variation problem in the approximating space.

(III) Increase the number of basis functions by some factor (e.g, 2), repeat step (II)

(IV) Stop this procedure if the difference between solutions based on two consecutive steps is smaller than a given tolerance, as judged by
$\frac{||f_{\hat{\lambda}(2k)} \perp f_{\hat{\lambda}(k)}||}{||f_{\hat{\lambda}(k)}||} \leq 10^{\perp 4}$.

4.1.4 Minimizing the OneStepRGACV function

In this section, we will discuss some efficient ways to search for the minimizer λ of the OneStepRGACV function. The OneStepRGACV function involves the solution of some nonlinear system so it cannot be explicitly expressed in terms of the smoothing parameters. Thus, we don't have the first or second derivative of the OneStepRGACV. Hence, optimization method which does not require the derivative information is highly desired. For one smoothing parameter case, golden section method should be a good method in finding the minimizer λ. Our major concern is in the situation that there are more than one smoothing parameter, say 4 or 5 smoothing parameters.

Standard optimization methods, such as Newton method or conjugate gradients are not suitable for our problem here since they require first and possibly second derivatives. Powell method and down hill simplex method might be possible ways to use since they don't require any derivative.

For all iterative procedure for solving a nonlinear optimization problem, a starting guess usually should be provided. A good starting guess might make the search faster. Although the issue of how to set a good initial guess is somewhat problem dependent, we believe that when the shape of the function is not too 'complicated' some automatic way to decide the starting point which is not too far away from the minimum (or local minimum) should be possible. In this thesis, we will investigate using computer experiment design technique to set a possible good starting guess.

The basic idea of computer experiment design is as follows. First we generate some design points at the possible region and evaluate the function value over the design points. Then a smooth surface (e.g., thin-plate spline, quadratic polynomial, etc.) can be interpolated over the design points and the minimum is found for the interpolating surface. In the case of using low degree polynomial, the least square solution can be used. We can use the minimizer of the interpolating (or least square) surface as the starting guess for the down hill simplex search (or Powell search). Since this is just a pre-screen procedure, a small number of design point should be enough for our purpose if a good design method is employed. With a small number of design points and high dimension, we decide to use Latin hypercube design to generate the design points.

For a very large data set, we may encounter the situation that one step randomized GACV function is still expensive to evaluate, i.e., in the case that it is exceedingly expensive to get a smoothing spline fit for a fixed λ. Fortunately we will see that the surface of the OneStepRGACV is generally in a very good shape so that the minimizer of the interpolating surface would be good enough. In this case, we can use the computer experiment design to locate the minimum roughly. To be conservative, we can use multi-stage computer design to look for the minima.

4.1.5 Bayesian Confidence Intervals for the Approximate Solution

The basis for our approach is a finite-dimensional Bayesian formulation of the smoothing spline estimation problem similar to the approach used by Silverman (1985). For the exact smoothing spline estimation, the conclusions of Silverman's approach parallel closely those of Wahba (1978, 1983). Due to the setup of the variational problem for the approximate smoothing spline estimate, the argument used by Wahba (1978, 1983) is difficult to be extended to our setting but the argument used by Silverman can be easily extended.

Firstly let's look at the Bayesian formulation of the approximate smoothing spline estimate.

Suppose on domain $\mathcal{T}$ one observes $y_j = f(t_j) + \epsilon_j$, $j = 1, \cdots, n$, where $t_j \in \mathcal{T}$, and $(\epsilon_1, \cdots, \epsilon_n) \sim N(\mathbf{0}, \sigma^2 W^{\perp 1})$ with W (positive definite) known. By section 4.1.3, the approximate smoothing spline estimate of $f(t)$ is in the finite-dimensional space $span\{\phi_1, \cdots, \phi_m, \xi_{i_1}, \cdots, \xi_{i_k}\}$. Hence f can be written as $f(t) = \sum_{l=1}^{m} d_l \phi_l(t) + \sum_{u=1}^{k} c_u \xi_{i_u}$. Define $b = \frac{n\lambda}{\sigma^2}$. Using the notation $\stackrel{c}{=}$ to mean "equals up to a constant", take the prior log likelihood to be

$$l_{prior}(c, d) \stackrel{c}{=} -\frac{1}{2} b c^T \Sigma_{11} c, \tag{4.1.11}$$

where $(\Sigma_{11})_{lj} =< \xi_{i_l}, \xi_{i_j} >$, and $\xi_{i_1}, \cdots, \xi_{i_k}$ are the selected basis from $\xi_1, \cdots, \xi_n$. Following standard Bayesian manipulation, we have the posterior log likelihood as follows,

$$l_{post}(c, d) \stackrel{c}{=} -\frac{1}{2} b c^T \Sigma_{11} c - \frac{1}{2\sigma^2} (Y - \Sigma c - Sd)^T W (Y - \Sigma c - Sd), \tag{4.1.12}$$

where $(\Sigma)_{lj} =< \xi_l, \xi_{i_j} >$ and $(S)_{iu} = \phi_u(x_i)$. This leads to the following theorem.

Theorem 4.5 *The posterior distribution of (c,d) is multivariate normal with mean* $(\hat{c}, \hat{d})$ *and covariance matrix* $\sigma^2 M^{\perp 1}$, *where*

$$M = \begin{pmatrix} \Sigma^T W \Sigma + n\lambda \Sigma_{11} & \Sigma^T W S \\ S^T W \Sigma & S^T W S \end{pmatrix} \tag{4.1.13}$$

and

$$\begin{pmatrix} \hat{c} \\ \hat{d} \end{pmatrix} = M^{\perp 1} \begin{pmatrix} \Sigma^T \\ S^T \end{pmatrix} Y. \tag{4.1.14}$$

Defining $y = (y_1, \cdots, y_n)^T$ and $f = (f(t_1), \cdots, f(t_n))^T$ with abuse of notation, the connections with the approximate spline smoothing becomes clear by noting that

$$-\frac{2\sigma^2}{n} l_{post}(f) \stackrel{c}{=} \frac{1}{n} (y - f)^T W (y - f) + \lambda \int f''(t)^2 dt$$

i.e., the approximate smoothing spline estimate $\hat{f}$ is the posterior mean in the Bayesian formulation described above.

Further, from the posterior variance/covariance of (c, d) obtained above we obtain the posterior variance of $f(s)$ which is given in the following theorem.

Theorem 4.6

$$Var_{post}(f(s)) = \sigma^2 u^T M^{\perp 1} u, \tag{4.1.15}$$

where $u = (\xi_{i_1}(s), \cdots, \xi_{i_k}(s), \phi_1(s), \cdots, \phi_m(s))^T$.

Defining the influence matrix $A(\lambda)$ satisfying $\hat{y} = A(\lambda) y$, it is easy to verify that

$$A(\lambda) = (\Sigma \quad S) M^{\perp 1} \begin{pmatrix} \Sigma^T \\ S^T \end{pmatrix} W.$$

On applying Theorem 4.6, we obtain Corollary 4.2.

Corollary 4.2 $Var_{post}(f) = \sigma^2 A(\lambda) W^{\perp 1}$.

Now let to turn to the Bayesian formulation of approximate smoothing spline estimate in Non-Gaussian case (binary data especially). it is assumed that the sampling likelihood of y is proportional to $exp\{-\frac{1}{\sigma^2}L(y|f)\} = exp\{-\frac{1}{\sigma^2}L_y(c,d)\}$, where $L(y|f) = L_y(c,d)$ is the negative log likelihood which is convex and $f(s) = \sum_{i=1}^{m} d_i\phi_i(s) + \sum_{j=1}^{k} c_i\xi_{i_j}(s)$. For binary data sets, σ is equal to 1. The approximate smoothing spline is the solution of the penalized likelihood problem

$$min\{L_y(f) + \frac{n}{2}\lambda J(f)\}, \tag{4.1.16}$$

where $f \in span\{\phi_1, \cdots, \phi_m, \xi_{i_1}, \cdots, \xi_{i,k}\}$. By substituting, we solve for (c,d) by the solution of the following variational problem

$$min\{L_y(c,d) + \frac{n}{2}c^T\Sigma_{11}c\}. \tag{4.1.17}$$

Under the same prior specified for the Gaussian case and following standard Bayesian manipulation, we have the posterior log likelihood as follows,

$$l_{post}(c,d) \stackrel{c}{=} -\frac{1}{\sigma^2}L_y(c,d) - \frac{n}{2}bc^T\Sigma_{11}c. \tag{4.1.18}$$

Letting $\hat{L}_y(c,d)$ be such an approximation of $L_y(c,d)$ with the Taylor expansion centered at the mode (c_*, d_*) of the posterior distribution $exp\{l_{post}(c,d)\}$, one gets

$$\hat{L}_y(c,d) = (\tilde{\tilde{y}} - W^{\perp 1}u - \Sigma c - Sd)^T W(\tilde{\tilde{y}} - \Sigma c - Sd), \tag{4.1.19}$$

where $\tilde{\tilde{y}} = \Sigma c_* + Sd_* - W^{\perp 1}u$, $u = (\partial L/\partial f)|_{f_*}$, $W = (\partial^2 L/\partial f \partial f^T)|_{f_*}$ and $f_* = \Sigma c_* + Sd_*$. We approximate the posterior likelihood $l_{post}(c,d)$ via

$$\tilde{l}_{post}(c,d) \stackrel{c}{=} -\frac{1}{\sigma^2}\hat{L}_y(c,d) - \frac{n}{2}bc^T\Sigma_{11}c. \tag{4.1.20}$$

Theorem 4.7 *The approximate posterior distribution $exp\{\tilde{l}_{post}(c,d)\}$ is Gaussian with mean $(\hat{c},\hat{d})$ and covariance given in Theorem 4.5, where $(\hat{c},\hat{d})$ is the solution of (4.1.17) and the matrix W is the Hessian matrix defined above.*

Proof. It is easy to see that $exp\{\tilde{l}_{post}(c,d)\}$ is identical to a Gaussian sampling likelihood with covariance $\sigma^2 W^{\perp 1}$ and observations $\tilde{\tilde{y}}$; hence, the mean and covariance of $exp\{\tilde{l}_{post}(c,d)\}$ can be calculated via Theorem 4.5. Hence, it is left to show that

$$\begin{pmatrix}\hat{c}\\ \hat{d}\end{pmatrix} = \begin{pmatrix}\Sigma^T W\Sigma + n\lambda\Sigma_{11} & \Sigma^T WS\\ S^T W\Sigma & S^T WS\end{pmatrix}^{\perp 1}\begin{pmatrix}\Sigma^T\\ S^T\end{pmatrix}\tilde{\tilde{y}}.$$

We have $(c_*, d_*) = (\hat{c},\hat{d})$ by noting that

$$-\sigma l_{post}(c,d) \stackrel{c}{=} L_y(c,d) + \frac{n}{2}\lambda c^T\Sigma_{11}c.$$

By definition of (c_*, d_*), it is easy to show that (c_*, d_*) satisfies

$$\begin{pmatrix}\Sigma^T W\Sigma + n\lambda\Sigma_{11} & \Sigma^T WS\\ S^T W\Sigma & S^T WS\end{pmatrix}\begin{pmatrix}c_*\\ d_*\end{pmatrix} = \begin{pmatrix}\Sigma^T W\tilde{\tilde{y}}\\ S^T W\tilde{\tilde{y}}\end{pmatrix}.$$

This completes the proof.

Similar to Theorem 4.6 and Corollary 4.2, it is easy to see that $Var_{post}(f(s)) \approx \sigma^2 u^T M^{\perp 1} u$ and $Var_{post}(f) = \sigma^2 A(\lambda) W^{\perp 1}$.

For the computation of the approximate variance and covariance, we can take advantage of the intermediate results when we solve for the estimate. When we solve for the the estimate $(\hat{c}, \hat{d})$, we need to solve a linear system of the form $Mx = z$. This is done through QR decomposition. Hence, when we calculate posterior variance and covariance, the major computation $M^{\perp 1} u$ can be done using the existing QR decomposition in the last step of Newton-Raphson iteration for obtaining the solution $(\hat{c}, \hat{d})$.

4.1.6 Monte Carlo Simulation

In this section, we conduct Monte Carlo simulations to check the performance of the OneStepRGACV in term of finding the optimal smoothing parameters in the case of Bernoulli data. The Comparative Kullback-Leibler distance (CKL) will be used to measure the performance. The sampling CKL between two probabilities $p_1(t)$ and $p_2(t)$ for binary data is defined as follows,

$$CKL(p_1, p_2) = \frac{1}{n} \sum_{i=1}^{n} [-p_1(x_i) log(p_2(x_i)) - (1 - p_1(x_i)) log(1 - p_2(x_i))],$$

where $x_i, \cdots, x_n$ are the design points.

I. Single Smoothing Parameter

The following four test functions (used by Cox and Chang (1990) and Xiang (1996)) will be used in our simulation study.

$$\begin{aligned}
\eta_1(x) &= 2sin(10x), \\
\eta_2(x) &= 3 - (5x - 2.5)^2, \\
p_3(x) &= \begin{cases} -1.6x + .9 & \text{if } x \leq .5 \\ +1.6x - .7 & \text{if } x > .5, \end{cases} \\
p_4(x) &= \begin{cases} 3.5x/3 & if\ x \leq .6 \\ .7 & if\ x > .6, \end{cases}
\end{aligned}$$

where η_i indicates that the function is for the true logit while p_i stands for the probability. We plot the above four functions (in probability scale) in Figure 3.

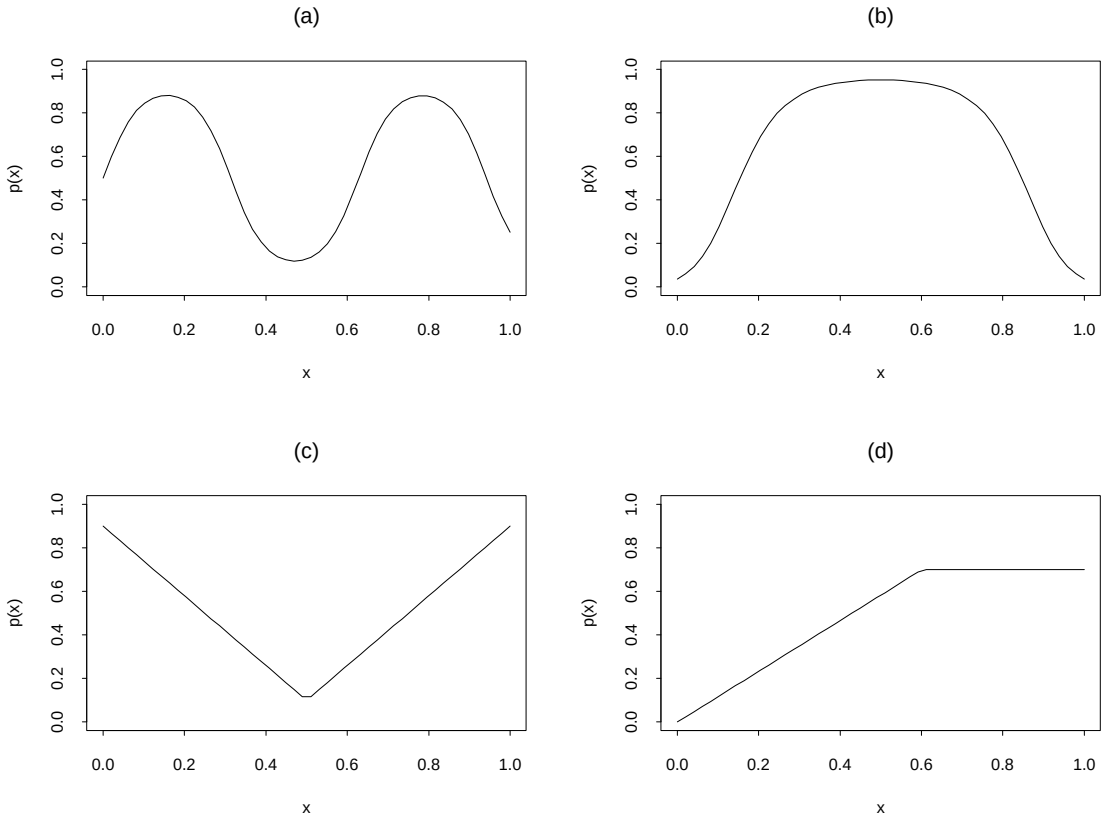


Figure 3: The true $p(x)$: (a) η_1, (b) η_2, (c) p_3 and (d) p_4.

First, we use $\eta(x) = \eta_1(x) = 2sin(10x)$ as a true function and generate the data $y_i = 1$ or 0 according to $p(x_i) = \frac{e^{\eta(x_i)}}{1+e^{\eta(x_i)}}$ with the design point chosen at $x_i = (i - 0.5)/500,\ \ i = 1, \cdots, 500$. Figure 4(a) gives a plot of $CKL(\lambda)$ and ten replicates of $OneStepRGACV(\lambda)$. In each replicates, R was taken as 1, and δ was generated anew as a Gaussian random vector with $\sigma_\delta = 0.001$. The minimizer of the CKL is at the fill-in square and the 10 minimizers of the 10 replicates of $ranGACV$ are the open circle. From the plot, we can see that any one of these 10 provides a rather good estimate of the λ that goes with the fill-in circle. Figure 4(b) gives the same experiment except that this time the number of replicates R was taken as 5. It can be seen that the minimizers of $OneStepRGACV(\lambda)$ become 'even more reliable estimates of the minimizer of CKL, and the CKL at all of the $OneStepGACV$ estimates are actually quite close to its minimum value.

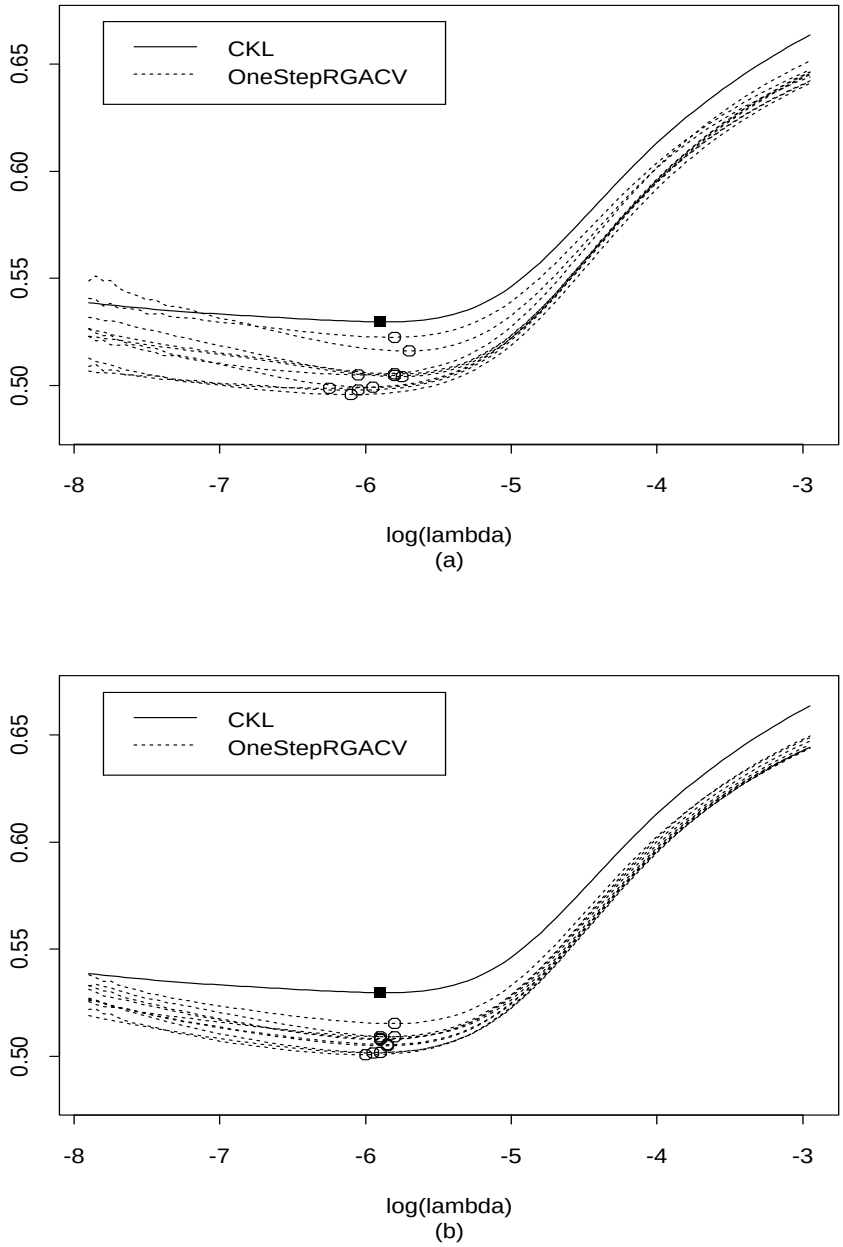


Figure 4: 10 replicates of OneStepRGACV compared with CKL (a) one replicate for each curve. (b) Five replicates for each curve.

Next, we use a simulation to check how sensitive the one-step randomized GACV is with respect to the change of perturbed size and the number of replicates. Two different logistic functions are used. They are

$$\begin{aligned} \eta_1(x) &= 2sin(10x), \\ \eta_2(x) &= 3 - (5x - 2.5)^2. \end{aligned}$$

The results are shown in Figure 5 and Figure 6. We can see that the result is not too sensitive to the change of the perturbed size except that when the size is very small. This might be due to the rounding error.

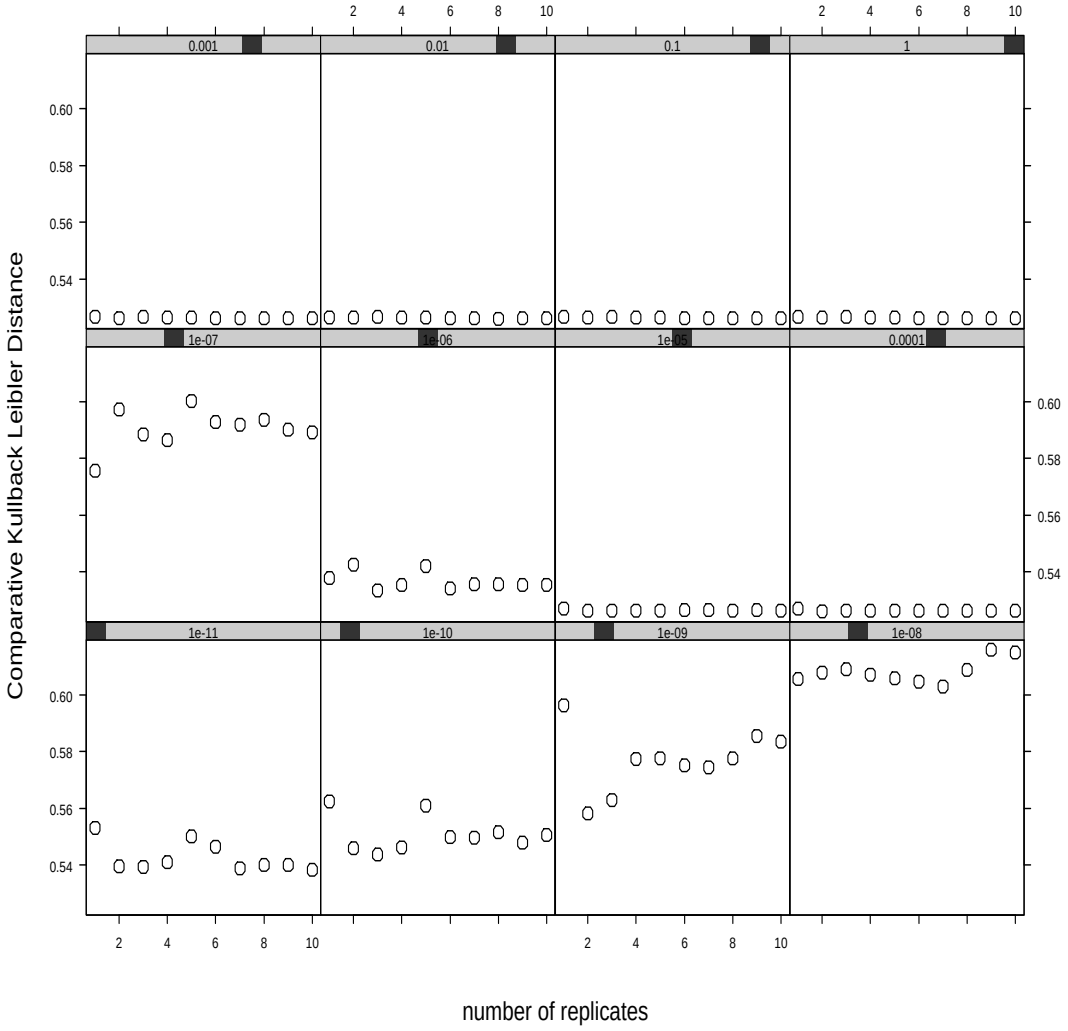


Figure 5: For η_1: performance of OneStepRGACV for different size of perturbation (the number's in grey title bars) and number of replicates as measured by the CKL.

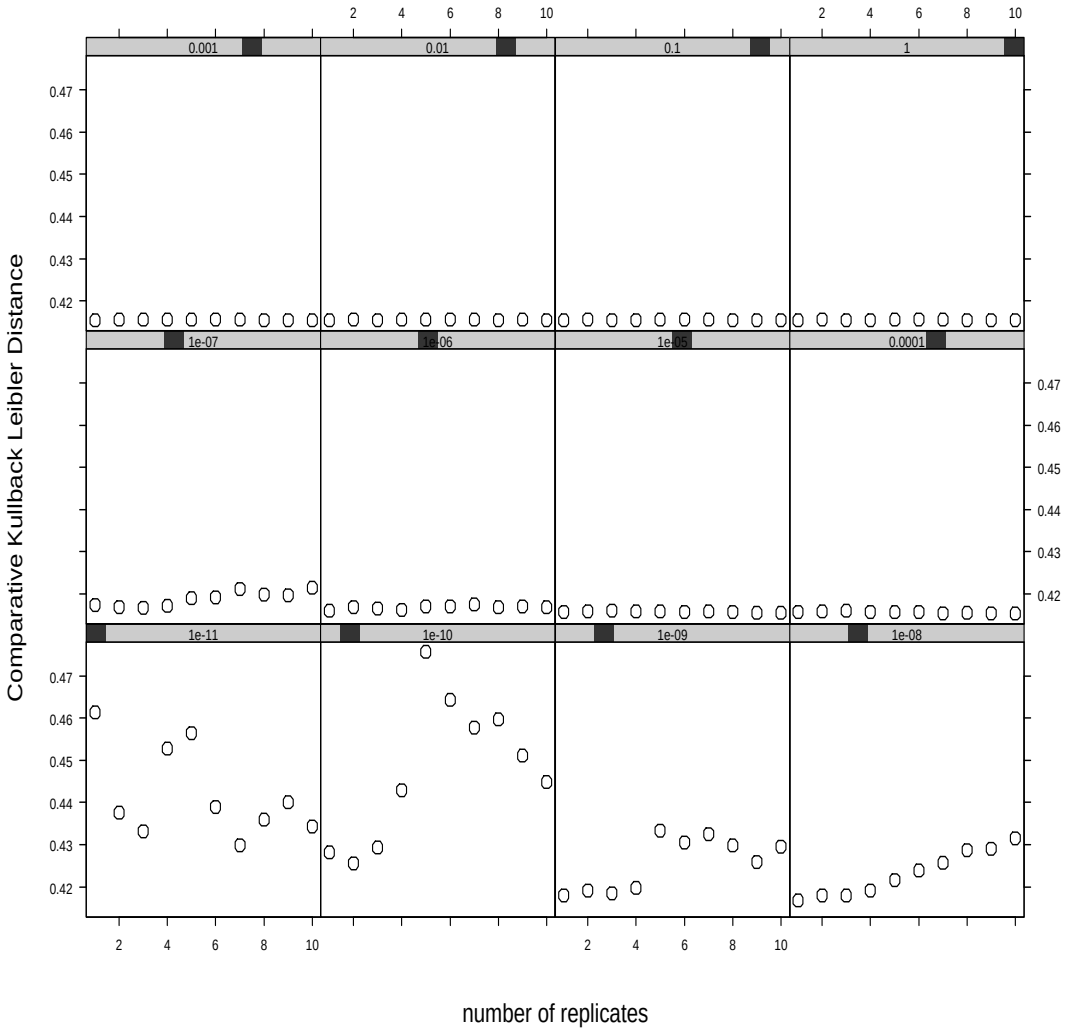


Figure 6: For η_2: performance of OneStepRGACV for different size of perturbation (the number's in grey title bars) and number of replicates as measured by the CKL.

Next, we use a simulation to check the performance of the OneStepRGACV function as compared with the RGACV function. We use $\sigma = 0.001$, $R = 5$ to calculate the OneStepRGACV and RGACV functions. All the four test functions will be used in this comparison. For each function, we generate 500 observations at design points $x_i = (i - .5)/500,\ i = 1, \cdots, 500$. According to $p(x_i)$, we generate Bernoulli data and fit the data with λ's chosen from OneStepRGACV and RGACV respectively. Then we calculate the CKL distance for OneStepRGACV fit and RGACV fit. The experiment is repeated 200 times. We plot CKL_{RGACV} versus $CKL_{OneStepRGACV}$ from the 200 runs in Figure 7. For any given data set, the smaller CKL value indicate that the corresponding method produces better fit. From figure 7 we find that OneStepRGACV performs as well as RGACV does. Hence, we recommend using the OneStepRGACV instead of RGACV to approximate the GACV since the calculation of OneStepRGACV is faster. Simulation in Xiang (1996) shows that the RGACV method outperforms the UBR method using the same four testing functions above. Hence, we can expect that the OneStepRGACV will outperform the UBR method.

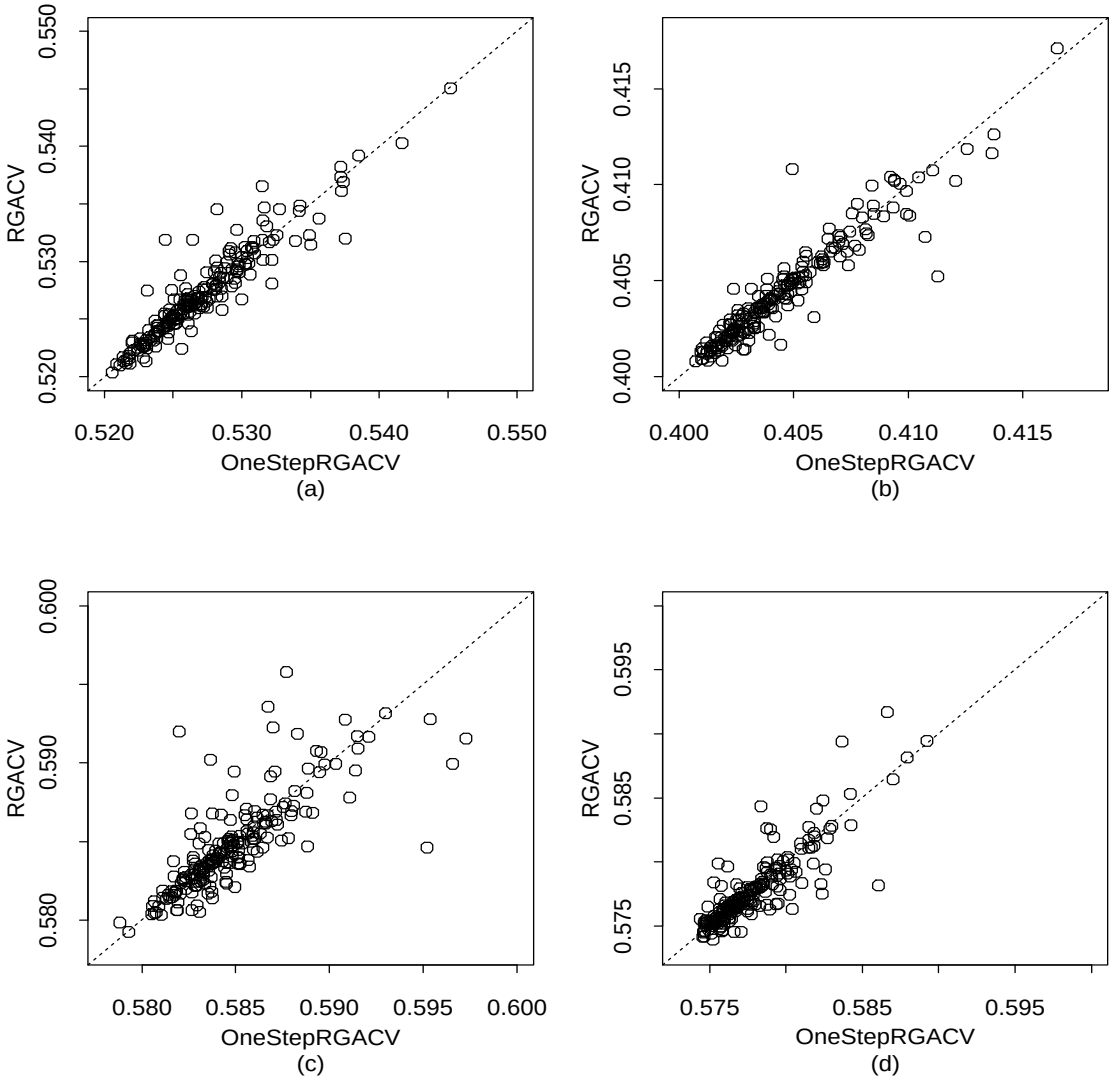


Figure 7: CKL Comparison of OneStepRGACV with RGACV for (a)η_1, (b)η_2, (c)p_3 and p_4.

II. Multiple Smoothing Parameters

Example 1 In this simulation example, the bivariate additive function

$$f(x_1, x_2) = 5sin(2\pi x_1) - 3sin(2\pi x_2)$$

is used as true logit function for generating data. The test function in probability scale is plotted in Figure 8. We generate 500 design points (x_{1i}, x_{2i}) uniformly from the square $(0,1) \times (0,1)$ and the response $y_i = 1$ or 0 according to $p(x_{i1}, x_{i2}) = exp(f(x_{i1}, x_{i2}))/(1 + exp(f(x_{i1}, x_{i2})))$. We fit an additive model

$$f_{\lambda_1,\lambda_2}(x_1, x_2) = f_{\lambda_1}(x_1) + f_{\lambda_2}(x_2)$$

to this set of data.

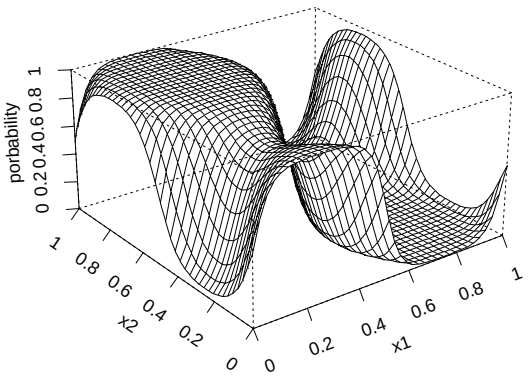
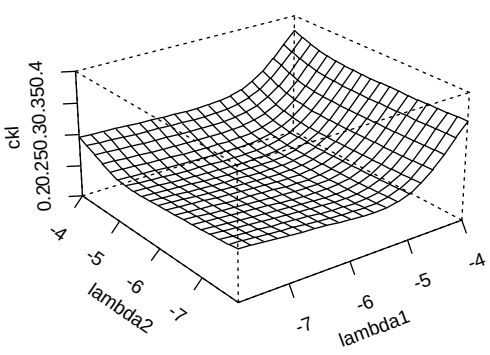
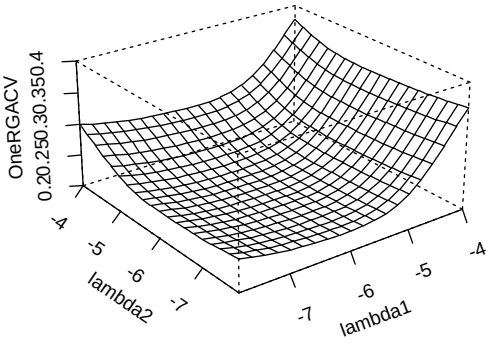


Figure 8: True test function for *Example 1*.

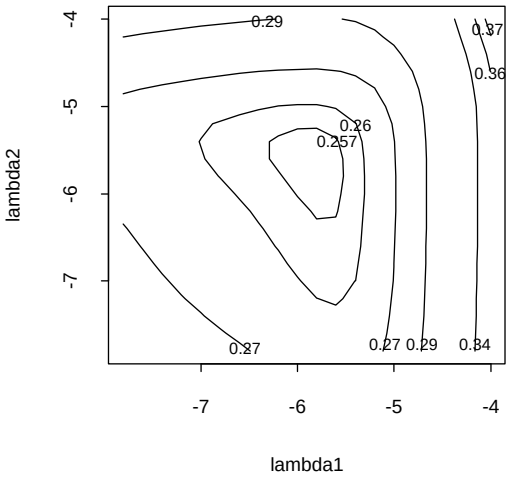
We use $\sigma = 1$ and 5 replicates in the calculation of OneStepRGACV function. Besides, 50 representative points are used in getting the approximate smoothing spline. For each pair of (λ_1, λ_2), we can evaluate the $OneStepRGACV(\lambda_1, \lambda_2)$ function. Using a 20×20 grid, we can draw a $OneStepRGACV(\lambda_1, \lambda_2)$ surface. In addition, for the fit at each pair of (λ_1, λ_2), the $CKL(\lambda_1, \lambda_2)$ based on the true function and $f_{\lambda_1,\lambda_2}(x_1, x_2)$ can also be calculated. We plot the CKL surface and OneStepRGACV surface as well as their contour plots in figure 9. From this plot, we can see that the OneStepRGACV function is a good proxy for CKL. To examine the possibility of using computer design method to search for the minimizer of OneStepRGACV function, we use Latin hypercube design to sample 20 design points over the smoothing parameter space and evaluate the OneStepRGACV function over the design points. Then a thin plate spline is used to interpolate the OneStepRGACV function over the design points. The contour plot of the interpolated surface is shown in Figure 10. By comparing the Figure 9 with Figure 10, we can see that the minimizer of the interpolated surface is close to the minimizer of the OneStepRGACV function.



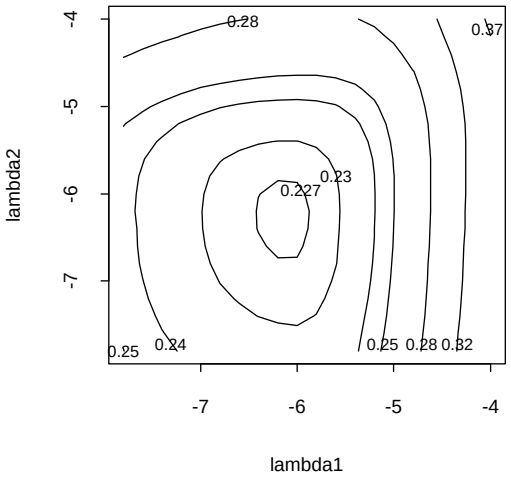
(a1) True CKL Surface



(a2) OneStepRGACV Surface



(b1) Contour of CKL



(b2) Contour of OneStepRGACV

Figure 9: OneStepRGACV and CKL surface and their contour plots.

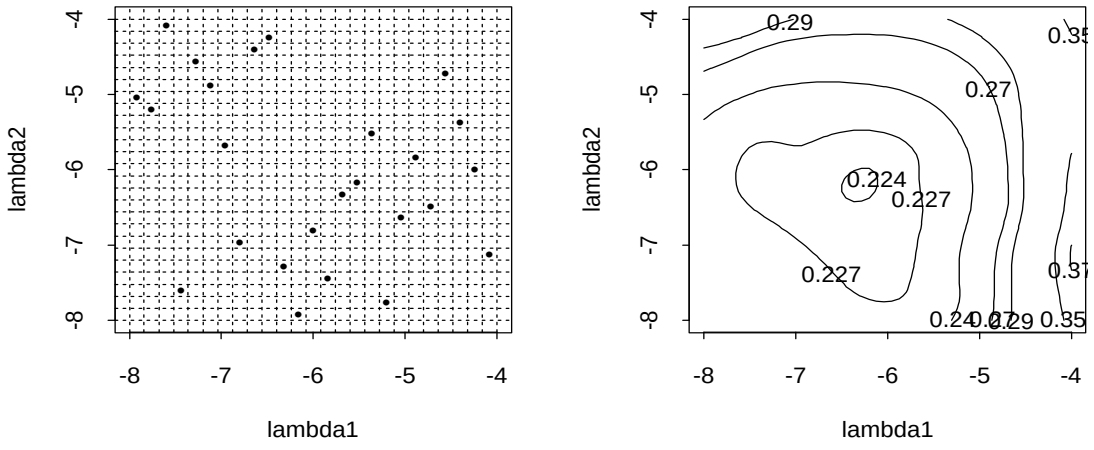


Figure 10: Left: Latin Hypercube Design. Right: contour plot of thin plate spline interpolation over the design points.

Next, we do the simulations to check the performance of the OneStepRGACV method as compared with the iterated UBR method. Again, we use $\sigma = 1$ and 5 replicates to calculate the OneStepRGACV function. We randomly generate 200 runs and plot the results in figure 11. The downhill simplex method is used to search for the minimizer of the OneStepRGACV function. From the plot we can see that one step randomized GACV method outperforms the iterated UBR method.

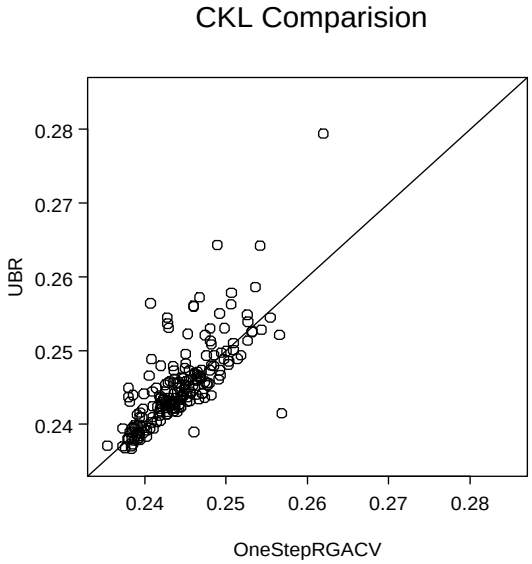


Figure 11: CKL Comparison of OneStepRGACV with iterative UBR based on 200 runs.

Example 2 The second Monte Carlo Simulation is done as follows. The India Pima data set (see Wang, 1994) is used here. The concerned response of this data set is whether a person tested positive for diabetes. The following two covariates are used:
X_1— Plasma glucose concentration a 2 hours in an oral glucose tolerance test
X_2— Body Mass Index (bmi).
The Smoothing spline ANOVA Model

$$logit(p(x_1, x_2)) = const + f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2)$$

is fitted using the GRKPACK (i.e, the iterated UBR is used to choose the smoothing parameters). Then we use the fitted logistic function as the true function to generate the data set. Figure 12 shows the scatter plot of the covariates and the fitted probability surface (i.e the true test function). Denote observations of the covariates as $(x_{1i}, x_{2i}), i = 1, \cdots, 500$, the fitted logit value for each observed subject as $f(x_{1i}, x_{2i})$, then the response y_i is generated to be 0 or 1 according to the probability $exp(f(x_{1i}, x_{2i}))/(1.0 + exp(f(x_{1i}, x_{2i})))$. To compare the performance of the OneStepRGACV with the iterated UBR method, 200 sets of data are generated and the CKL's are calculated for both methods. There are five smoothing parameters in this example. Figure 13(a) shows the pairwise comparison of $OneStepRGACV$ and iterated UBR methods, where CKL_{UBR} is plotted against $CKL_{OneStepRGACV}$. A point on the diagonal line means the two methods tie each other whereas a point above the diagonal line means $CKL_{OneStepRGACV}$ is smaller then CKL_{UBR}, which suggests $OneStepRGACV$ performs better than UBR on that set of data. From this figure, we can see that most of the case the $CKL_{OneStepRGACV}$ are almost the same as or less than CKL_{UBR}. We also use two-stage design (21 points in each stage) to search for the minimizer and plot the comparison of design search with the downhill simplex method in Figure 13(b)-(d). We use quadratic polynomial to interpolate the OneStepRGACV function over the design points. From the plot we can see that by using the design search, we can locate a point not far away from the optimal point.

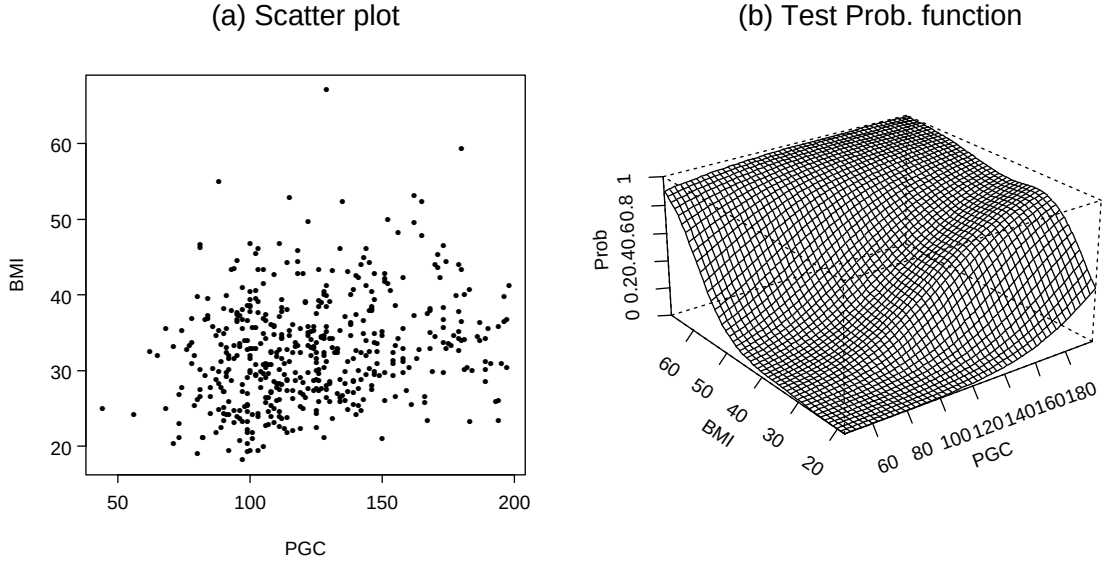


Figure 12: Scatter plots of the covariates and the probability surface of the test function.

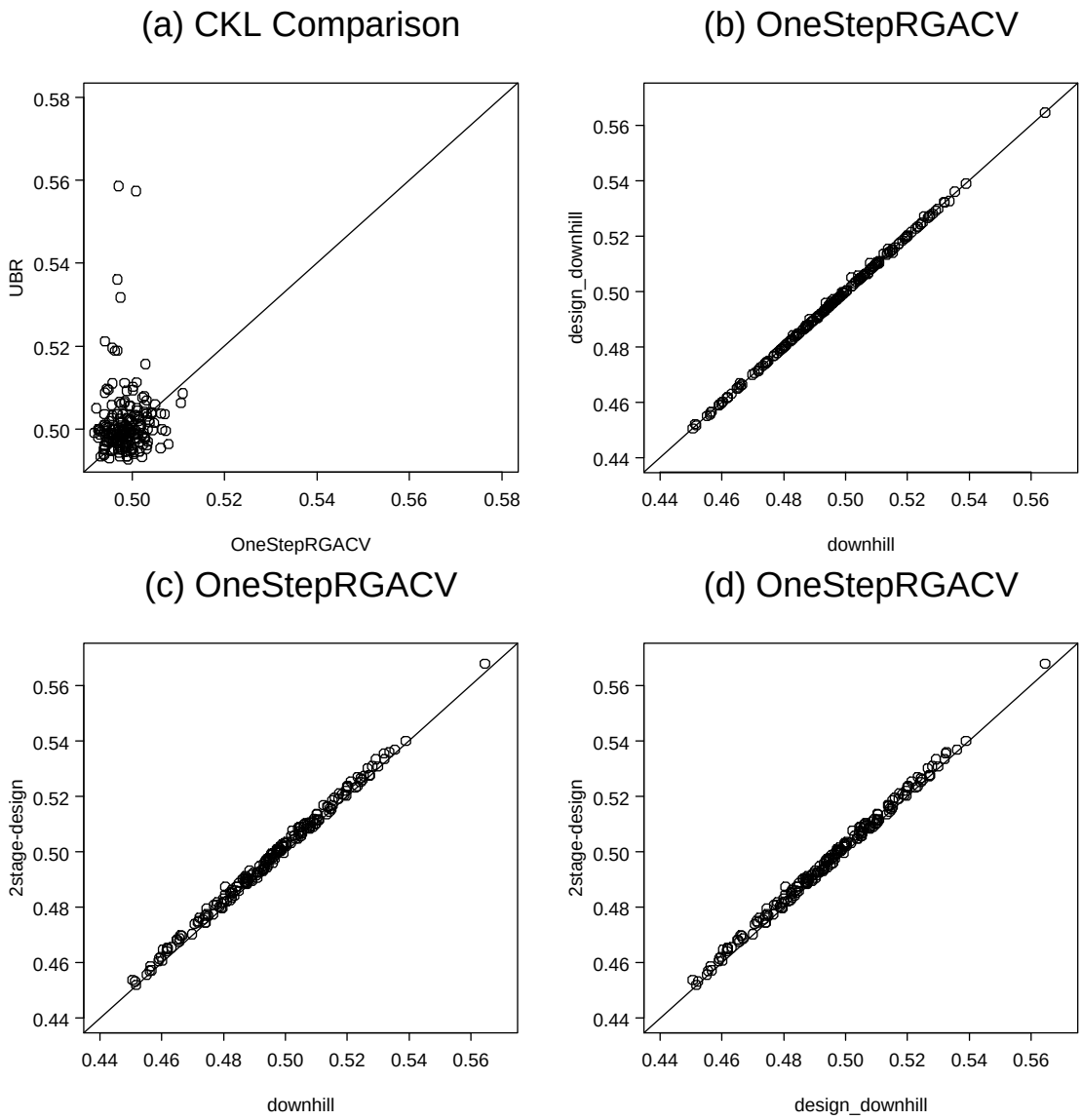


Figure 13: Pima Example—(a) CKL comparison: UBR vs OneStepRGACV , 200 runs. (b)—(d) Comparisons among several searching algorithms for finding the minimizer of the OneStepRGACV function: downhill simplex with a initial guess, downhill simplex with initial value chosen by design method, two-stage design.

Example 3 This example is similar to the *Example 2* except that we take the WESDR (Wisconsin Epidemiology Study of Diabetes Retinopathy) data. Three covariates dur, gly and bmi are used, and the progression of retinopathy is treated as response. First, the following ANOVA model is fitted by iterated UBR method (GRKPACK),

$$logit(p(dur, gly, bmi)) = c + f_1(dur) + f_2(gly) + f_3(bmi) + f_{13}(dur, bmi).$$

The fitted logit function is used as true test function for our simulation. 100 sets of data are generated. The ANOVA model above is fitted for each simulated data set. Iterated UBR and OneStepRGACV are used to choose the smoothing parameters and their performances are compared in form of the CKL distance between the fitted function and the test function. For OneStepRGACV method, we use $\sigma = 1$ and $R = 5$ in the calculation of OneStepRGACV. In the meantime, we use 50 basis functions to get the approximate smoothing spline when we use the OneStepRGACV method. Figure 14(a) shows the comparison results. Also different search algorithms for finding the minimizer of OneStepRGACV are compared. Three different search methods: (1) Downhill Simplex method with a good starting guess, (2) Downhill simplex with the initial guess decided by design method and (3) two-stage computer design method. Again, we use quadratic interpolation in the design method. Figure 14(b)–(d) plot the comparisons of OneStepRGACV for these three search algorithms. From Figure 14(a), we can see that OneStepRGACV outperforms the iterated UBR method. From the comparisons of OneStepRGACV value in Figure 14(b)–(d), we can see that in term of the object function OneStepRGACV the design method can gives us smoothing parameters which OneStepRGACV value is close to the minimum value.

In the OneStepRGACV approach, we use the OneStepRGACV function as a criteria to select the smoothing parameters and few basis functions to get the approximate solution. Next, we compare the fits from these two methods in some of the data sets in which the OneStepRGACV and the iterated UBR have similar performance, i.e. the CKLs are close. We plot the fitted surface of one such data set from both methods in Figures 15 to 24. From the plots, we can see that the fitted surface are almost identical. This phenomenon remains the same in all the other data sets we examined. Notice that the iterated UBR uses all the data points to form the basis functions while the OneStepRGACV only uses 50 representative points to form the basis functions. From this, we can roughly conclude the approximate solution by using 50 basis function is almost identical to the exact smoothing spline using all the basis functions. Hence, we conclude that the difference between these two methods are due to the way we choose the smoothing parameters.

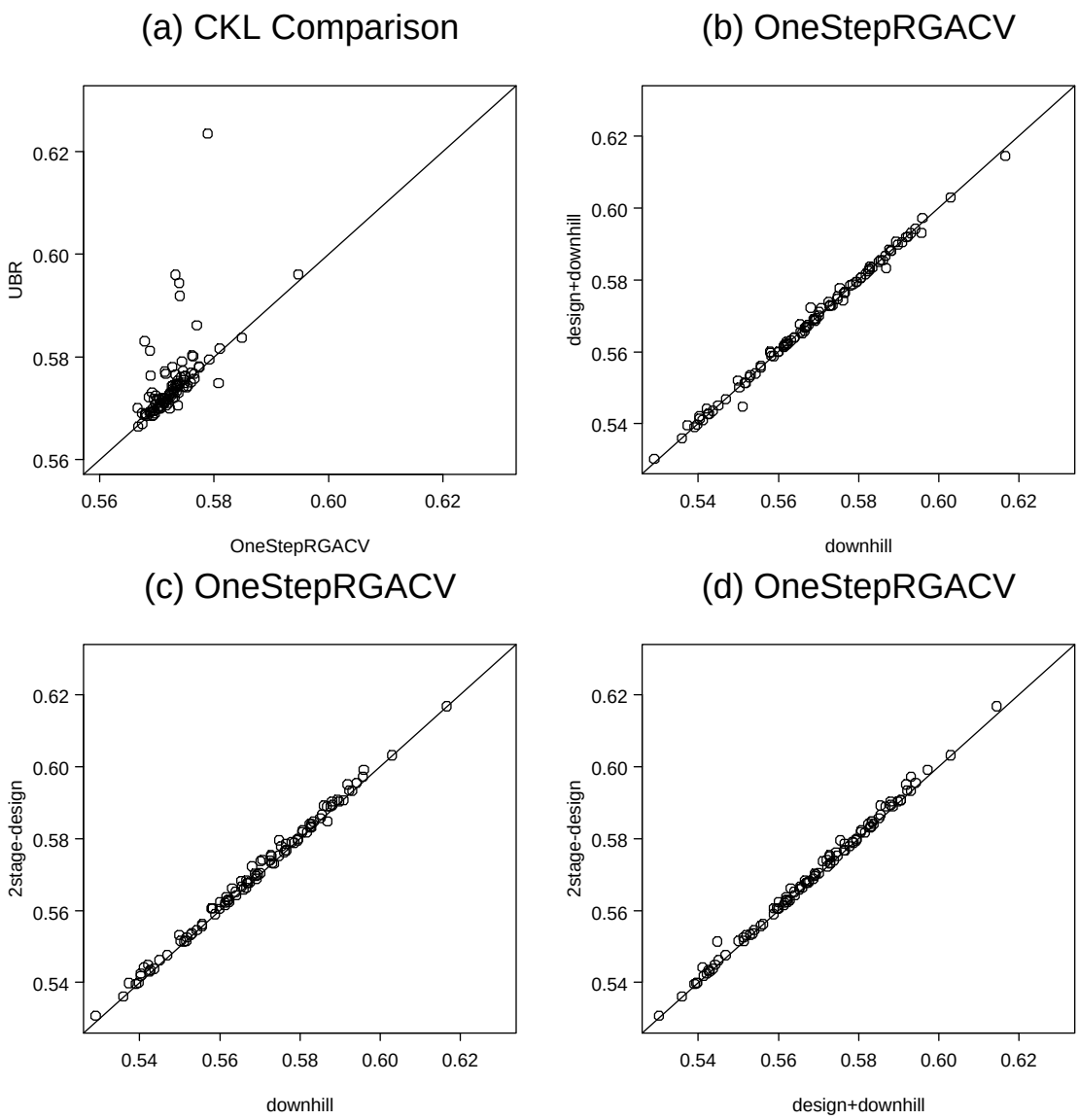


Figure 14: Wesdr Example— (a) CKL comparison: UBR vs OneStepRGACV, 100 runs; (b)-(d) Comparisons of different search algorithms.

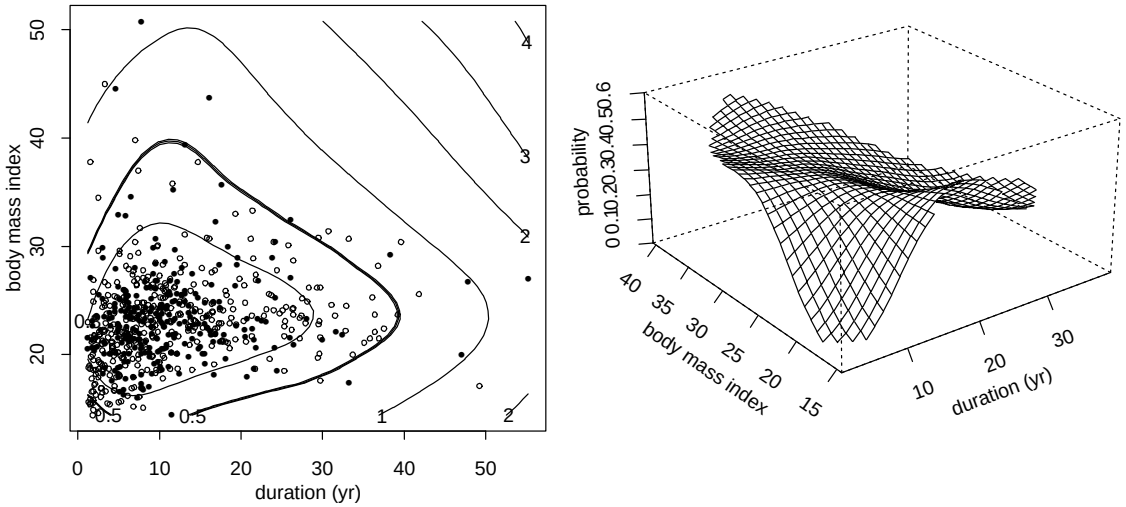


Figure 15: Wesdr Example—Left: data and contours of constant posterior standard deviation at the median **glycosylated hemoglobin** as a function of duration of **duration** and **bmi**. A solid point indicates a progression and a circle indicates a non-progression. Right: estimated probability in the defined region, as a function of **duration** and **bmi** at the median value of **glycosylated hemoglobin**. Sample Size–669. Method: OneStepRGACV, nrep=5, $\sigma = 1$, 50 bases.

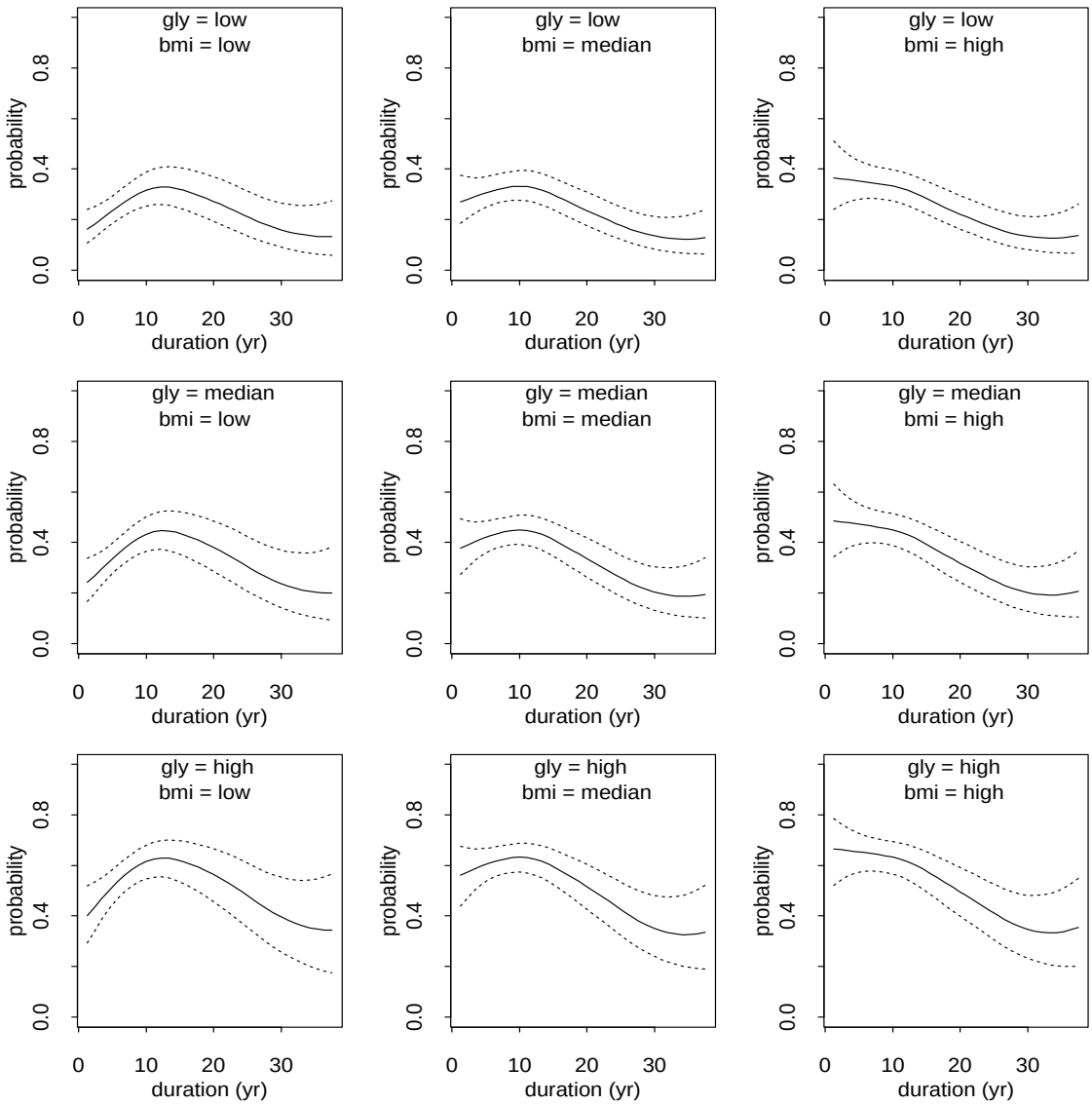


Figure 16: Wesdr Example—Cross sections of estimated probability of progression as a function of **duration** with their 90% Bayesian confidence intervals, at three levels of **gly** and three levels of **bmi**. Low, median and high denote .25, .5 and .75 percentiles. Method: OneStepRGACV.

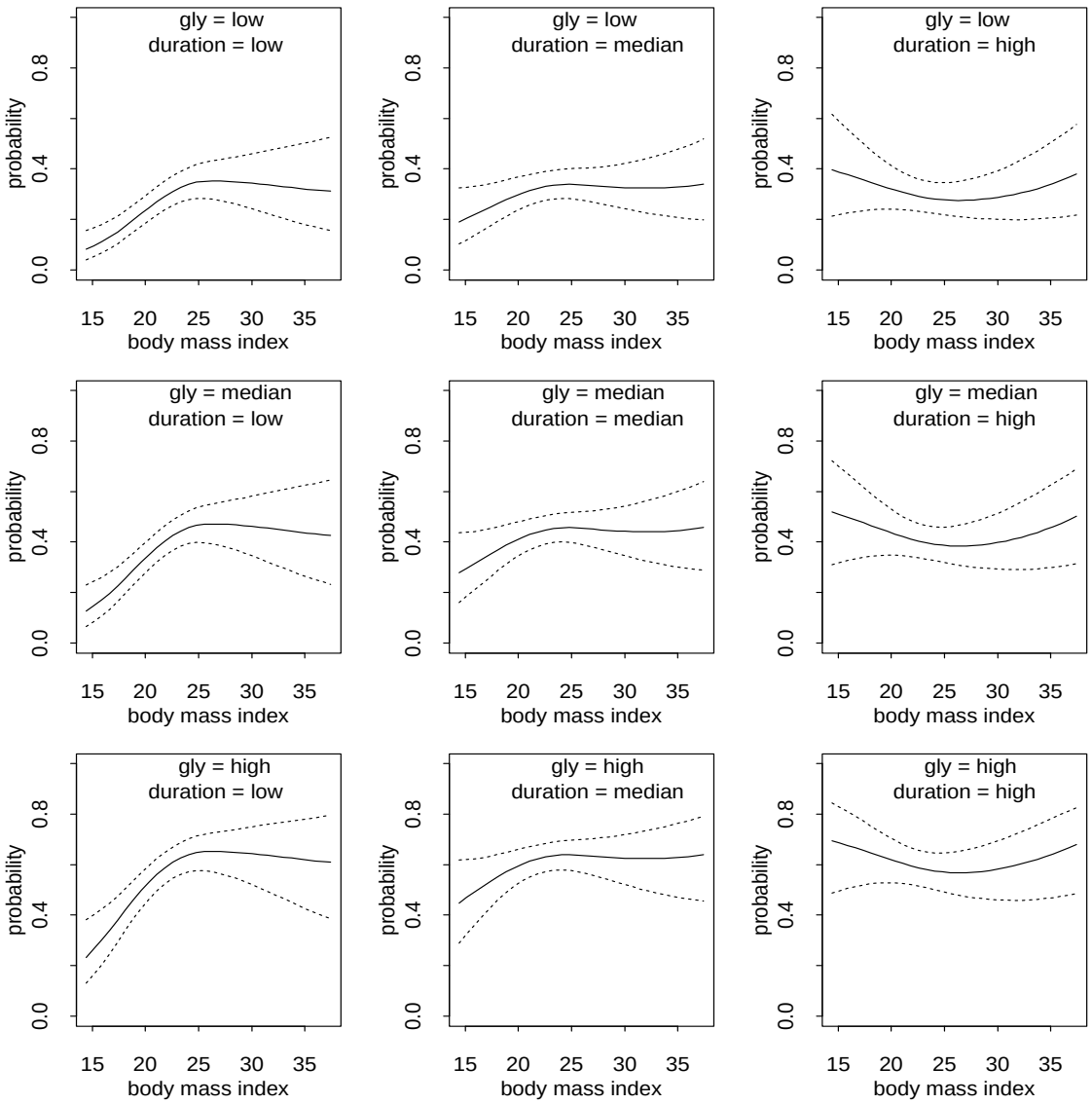


Figure 17: Wesdr Example—Cross sections of estimated probability of progression as a function of **bmi** with their 90% Bayesian confidence intervals, at three levels of **gly** and three levels of **duration**. Low, median and high denote .25, .5 and .75 percentiles. Method: OneStepRGACV.

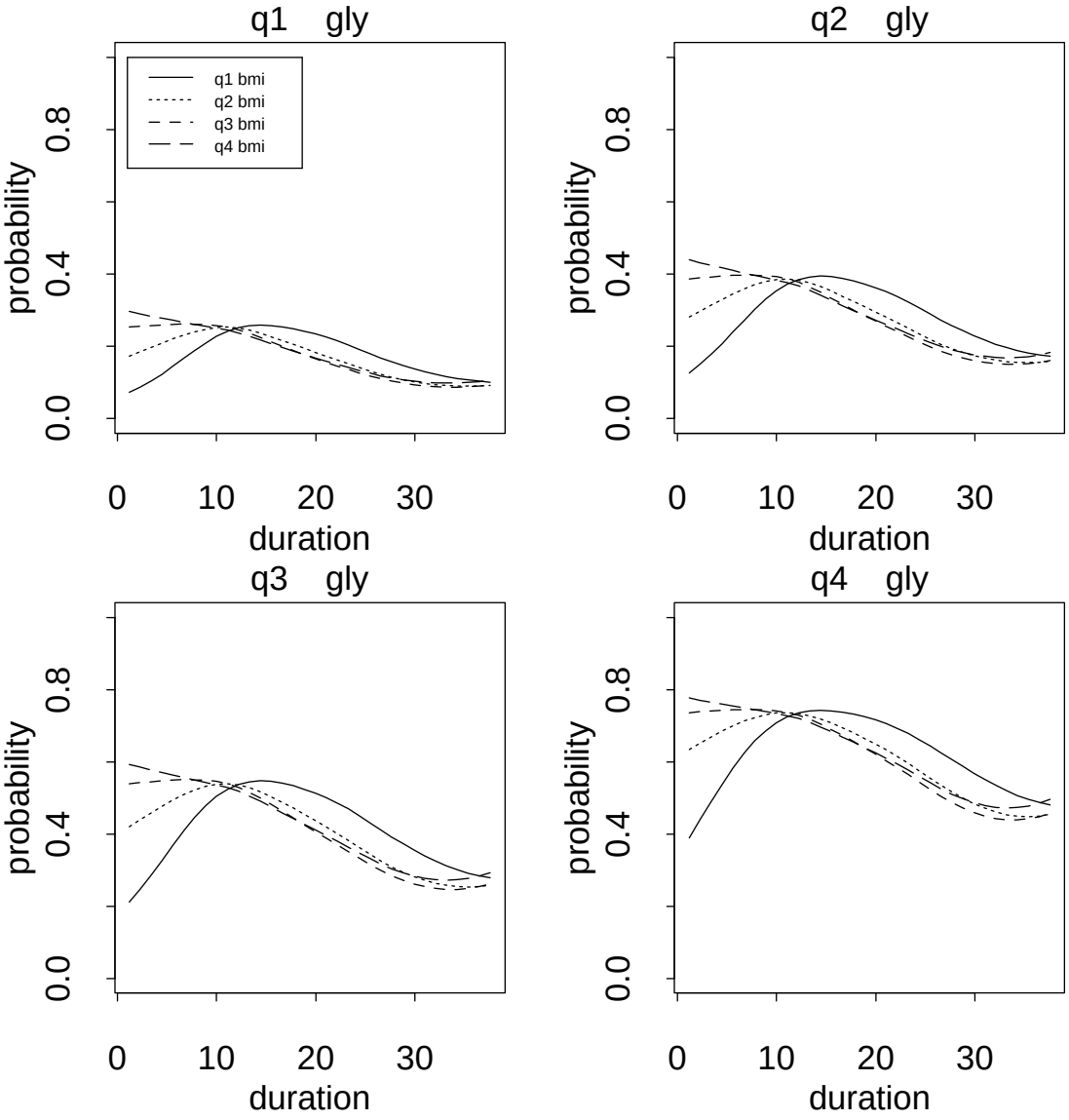


Figure 18: Wesdr Example—Cross sections of estimated probability of progression as a function of **duration**, at four levels of **gly** and four levels of **bmi**. q1, q2, q3 and q4 are the quantiles at $0.125, 0.375, 0.625$ and 0.875. Method: OneStepRGACV.

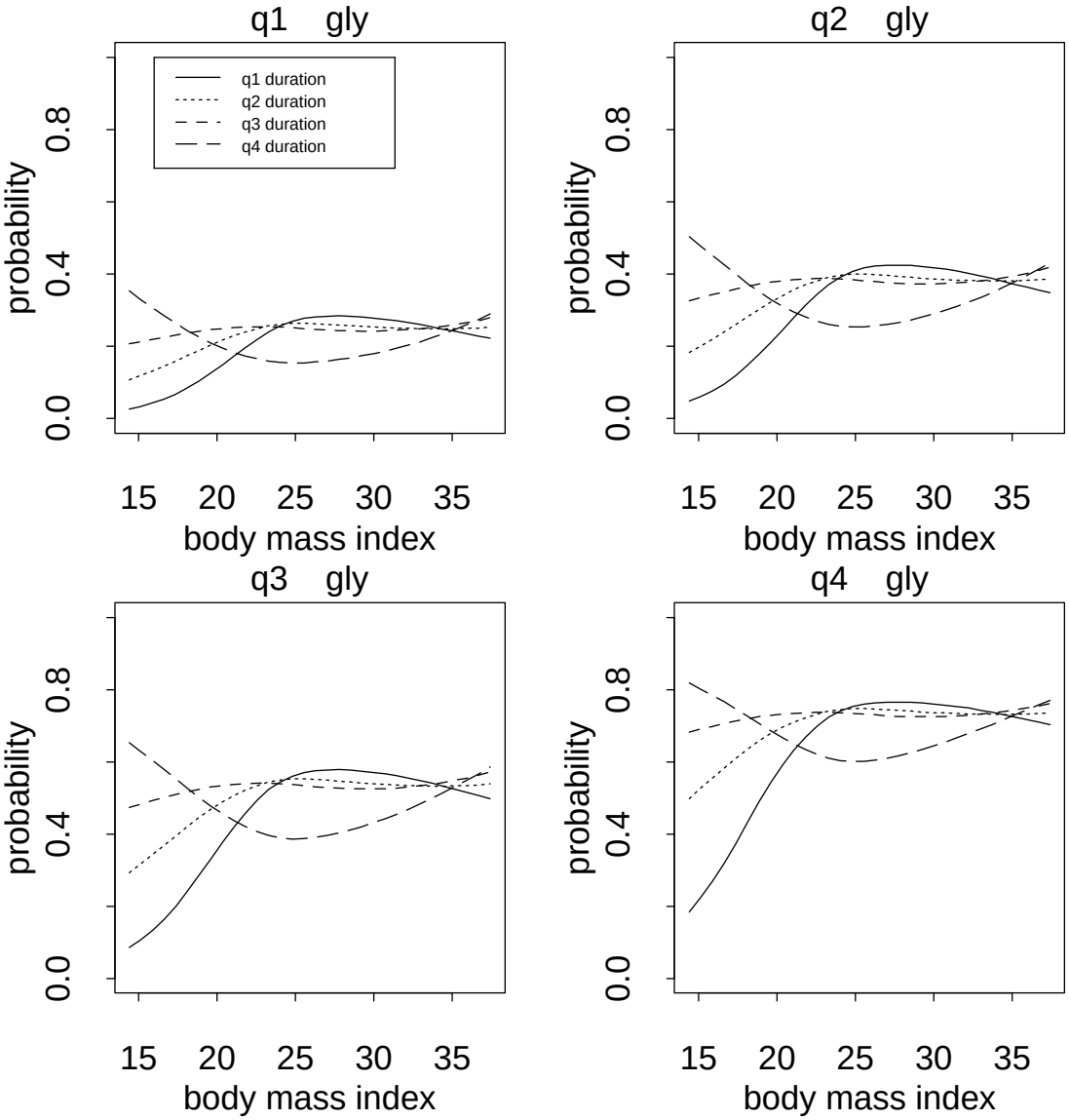


Figure 19: Wesdr Example—Cross sections of estimated probability of progression as a function of **bmi**, at four levels of **gly** and four levels of **duration**. q1, q2, q3 and q4 are the quantiles at $0.125, 0.375, 0.625$ and 0.875. Method: OneStepRGACV.

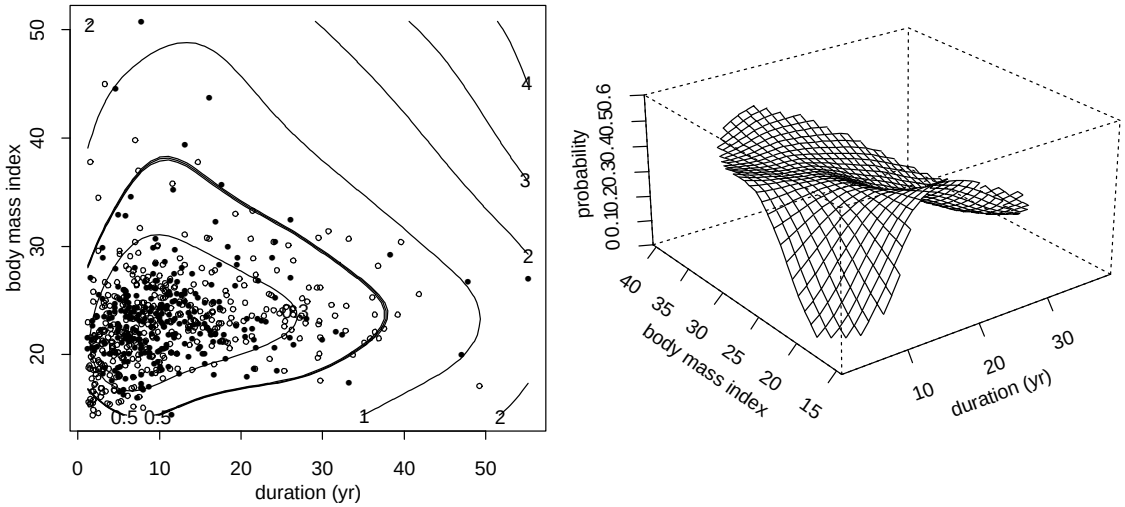


Figure 20: Wesdr Example—Left: data and contours of constant posterior standard deviation at the median **glycosylated hemoglobin** as a function of duration of **duration** and **bmi**. A solid point indicates a progression and a circle indicates a non-progression. Right: estimated probability in the defined region, as a function of **duration** and **bmi** at the median value of **glycosylated hemoglobin**. Method: GRKPACK.

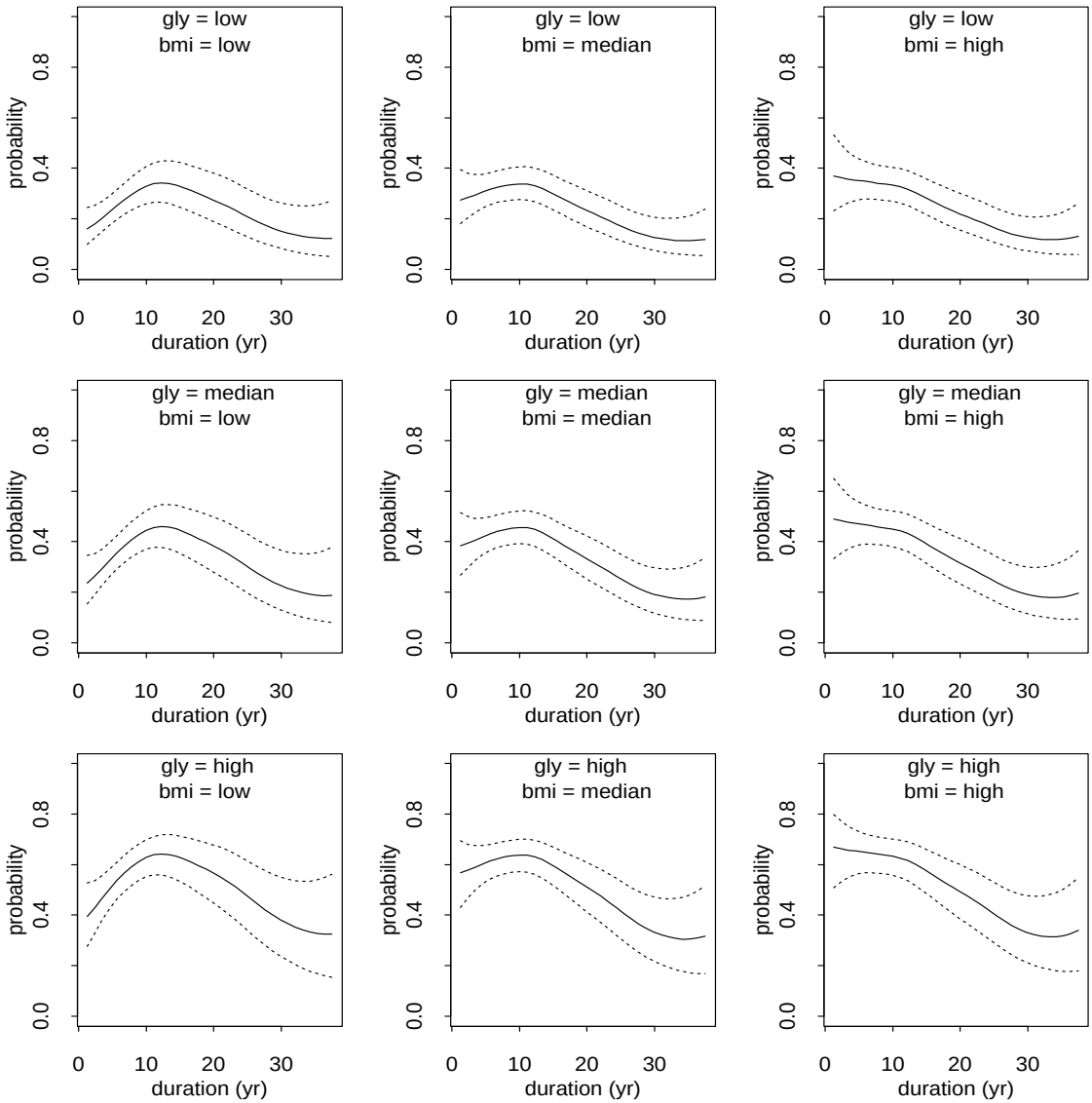


Figure 21: Wesdr Example—Cross sections of estimated probability of progression as a function of **duration** with their 90% Bayesian confidence intervals, at three levels of **gly** and three levels of **bmi**. Low, median and high denote .25, .5 and .75 percentiles. Method: GRKPACK.

For this simulated data set, we also examine the performance of the Bayesian Confidence Interval. The posterior variance and covariance derived in this Chapter is used to calculate the posterior standard deviation of the fitted surface when we use the OneStepRGACV method. For the iterative UBR method, the posterior standard deviation is calculated by using the formula derived in Wang (1994). Wang's formula is derived by following the approach in Wahba (1983). For OneStepRGACV, the coverage rate of 95% C.I is 94% and the coverage of 90% C.I. is 87%. For iterative UBR method, the coverage of 95% C.I is 96% and the coverage of 90% C.I. is 88%. This indicates the performance of the Bayesian Confidence for the OneStepRGACV method is close to that of the Bayesian Confidence Interval for the iterative UBR method. Also the coverage rates of both methods are close to the nominal levels.

Finally, we use the OneStepRGACV and approximate smoothing spline method to refit the model and data which appeared in Wahba *et al.* (1995). The number of representative points is 50. $\sigma = 1$ and 5 replicates are used in the calculation of OneStepRGACV. The results are presented in

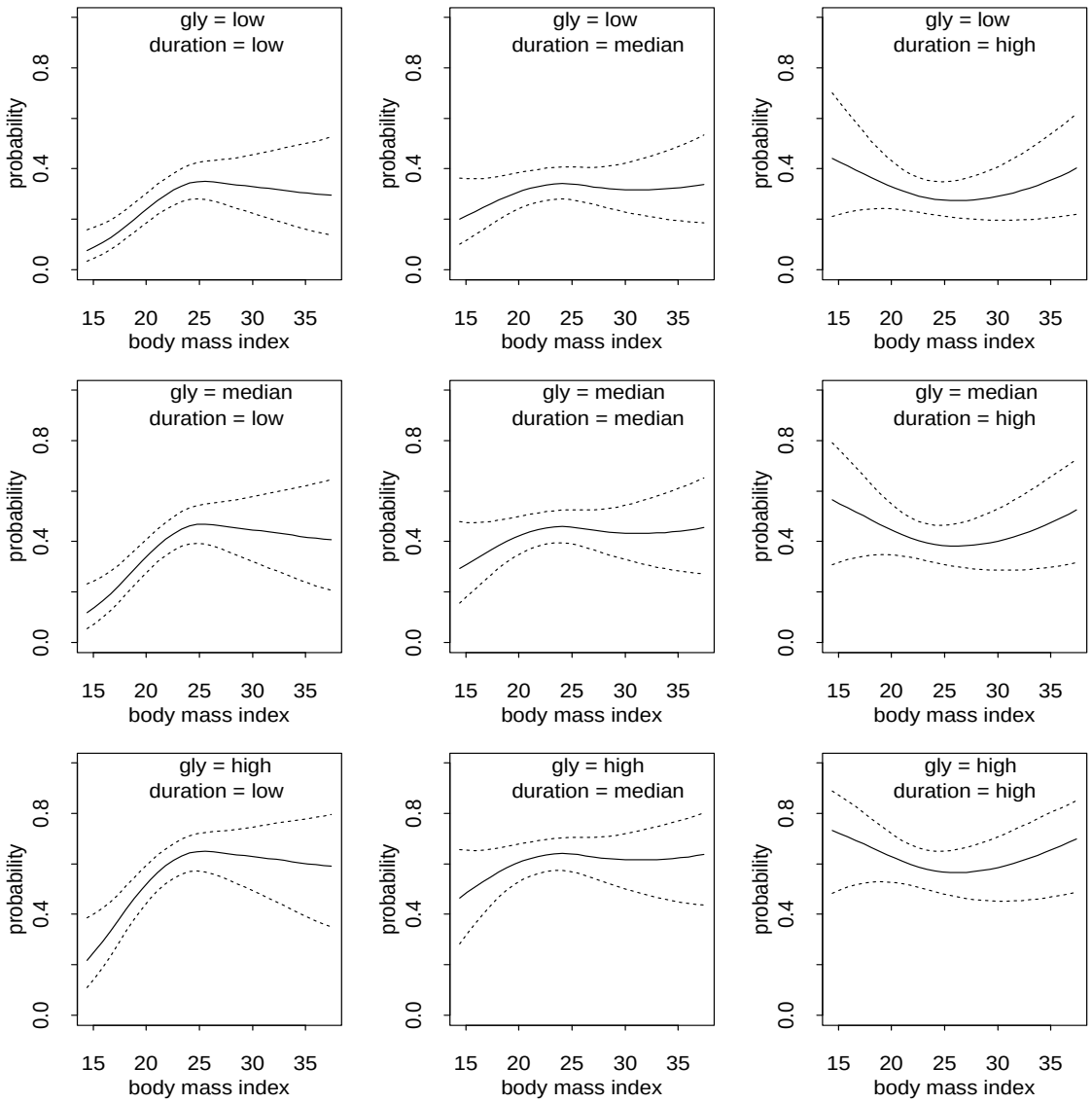


Figure 22: Wesdr Example—Cross sections of estimated probability of progression as a function of **bmi** with their 90% Bayesian confidence intervals, at three levels of **gly** and three levels of **duration**. Low, median and high denote .25, .5 and .75 percentiles. Method: GRKPACK.

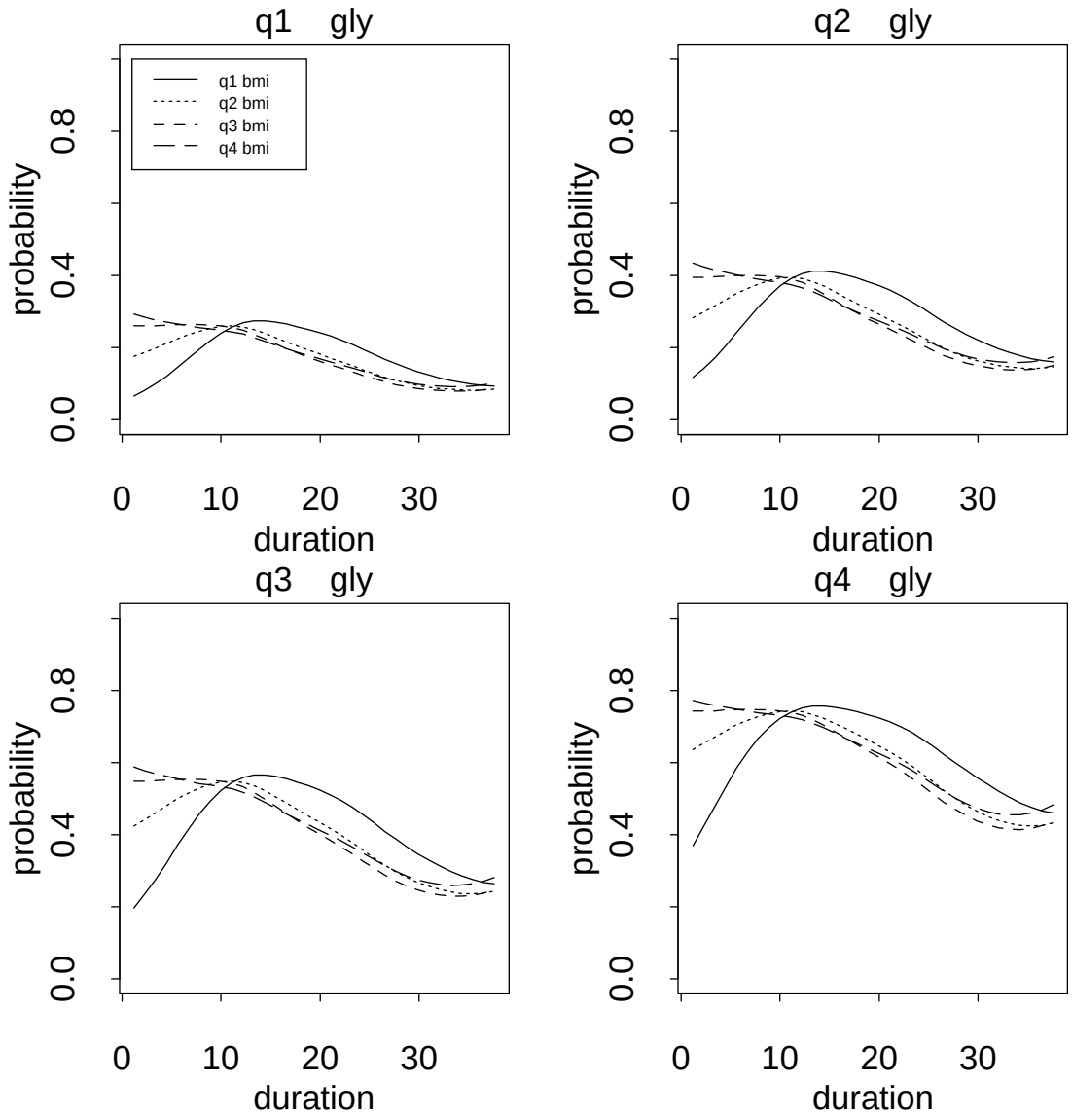


Figure 23: Wesdr Example—Cross sections of estimated probability of progression as a function of **duration**, at four levels of **gly** and four levels of **bmi**. q1, q2, q3 and q4 are the quantiles at $0.125, 0.375, 0.625$ and 0.875. Method: GRKPACK.

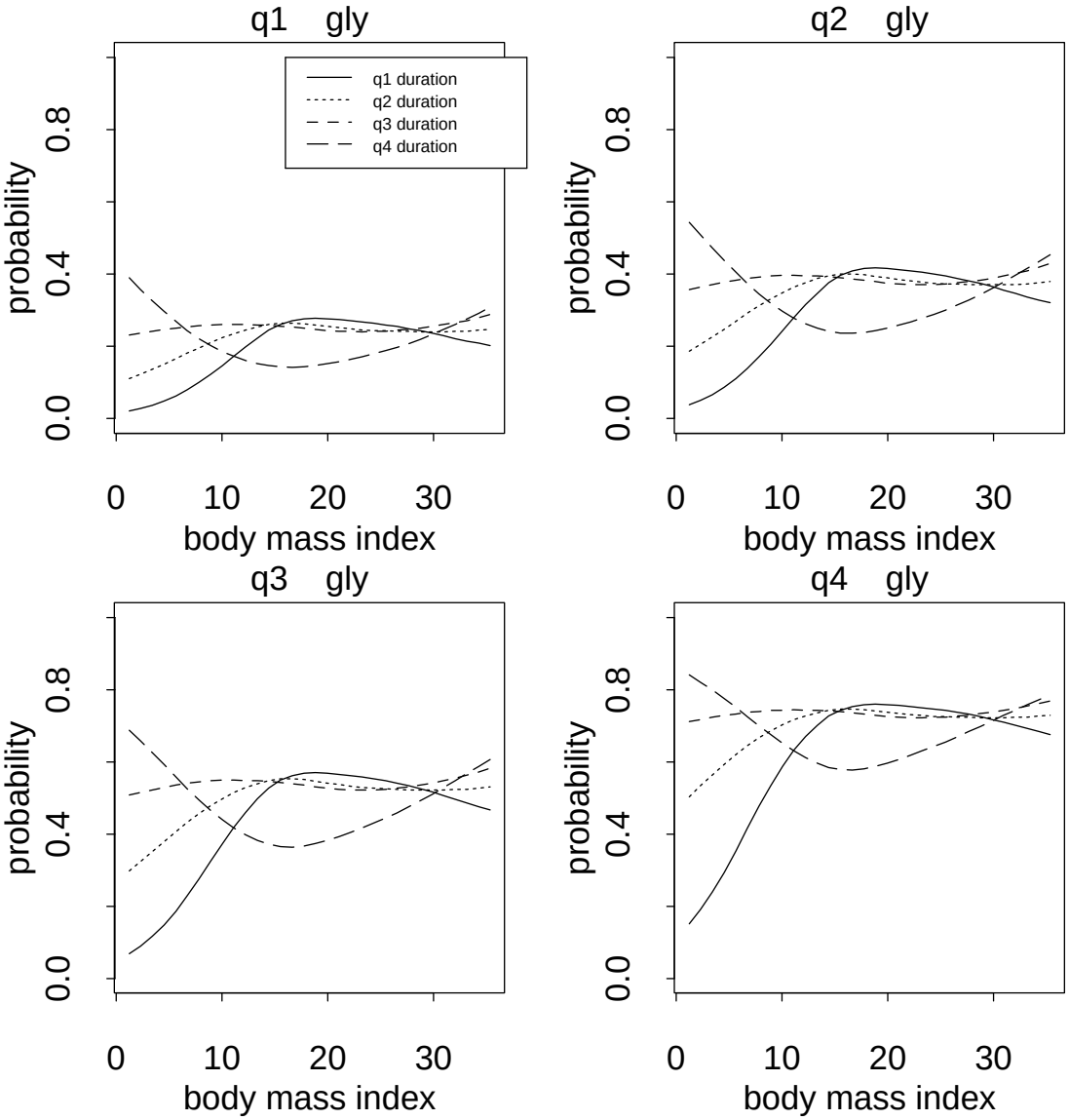


Figure 24: Wesdr Example—Cross sections of estimated probability of progression as a function of **bmi**, at four levels of **gly** and four levels of **duration**. q1, q2, q3 and q4 are the quantiles at $0.125, 0.375, 0.625$ and 0.875. Method: GRKPACK.

Figures 25 to 29. These figures looks almost the same as those appeared in Wahba *et al.* (1995). Again, We might conclude that by using the OnestepRGACV and fewer basis functions, we can obtain the estimate much faster than and get similar fit as the iterate UBR method when the iterated method gives us a good fit.

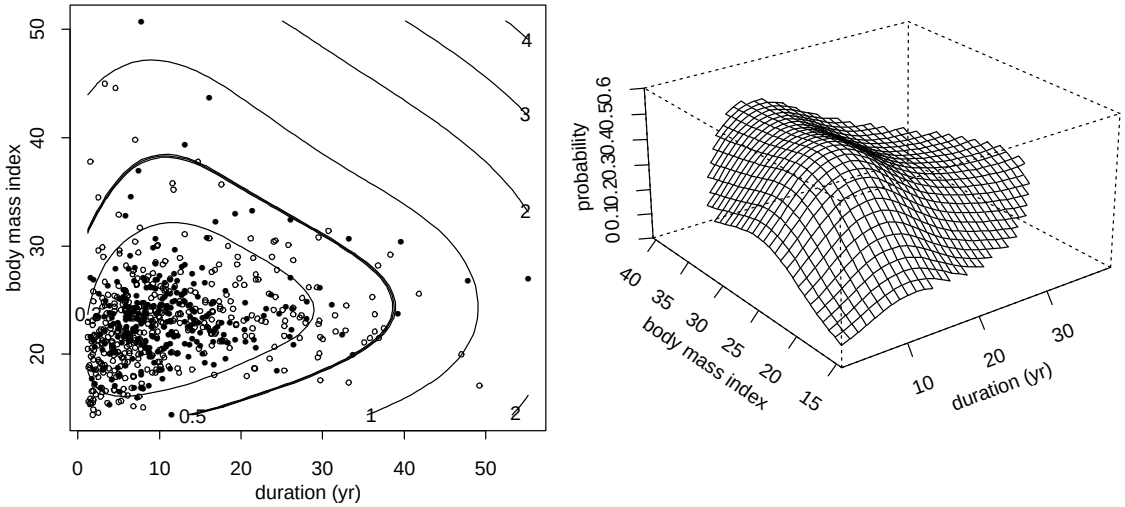


Figure 25: Wesdr Original Data—Left: data and contours of constant posterior standard deviation at the median **glycosylated hemoglobin** as a function of duration of **duration** and **bmi**. A solid point indicates a progression and a circle indicates a non-progression. Right: estimated probability in the defined region, as a function of **duration** and **bmi** at the median value of **glycosylated hemoglobin**. Method: OneStepRGACV.

4.2 Polychotomous Case

4.2.1 Fitting Polychotomous Response Data by Individual Fitting

We will discuss how we can use the fast algorithm developed for binary data to model the polychotomous response data.

Given $X = t$, the conditional class probabilities satisfy

$$p_0(t) + p_1(t) + \cdots + p_k(t) = 1. \tag{4.2.1}$$

Let $q_i = \sum_{l \neq i} p_l(t)$, then we will have $p_i(t) + q_i(t) = 1$. Notice that $p_i(t)$ corresponding to the conditional probability of a subject in the ith class given the covariate information $X = t$ while $q_i(t)$ denotes the probability that a subject is not in the ith class. As a result, we can use the algorithm developed for binary data to estimate the conditional probability $p_i(t)$. Denote

$$Z_i = \begin{cases} 1 & \text{if} \quad Y = i, \\ 0 & \text{otherwise,} \end{cases} \qquad i = 0, 1, \cdots, k. \tag{4.2.2}$$

By doing this, we have the random vector $(X, Z_0, \cdots Z_k)$ which is equivalent to the random pair (X, Y). Hence, we have $p_i(t) = P(Z_i = 1 | X = t)$. The random variables $Z_0, \cdots, Z_k$ have the following constraint,

$$Z_0 + \cdots + Z_k = 1 \tag{4.2.3}$$

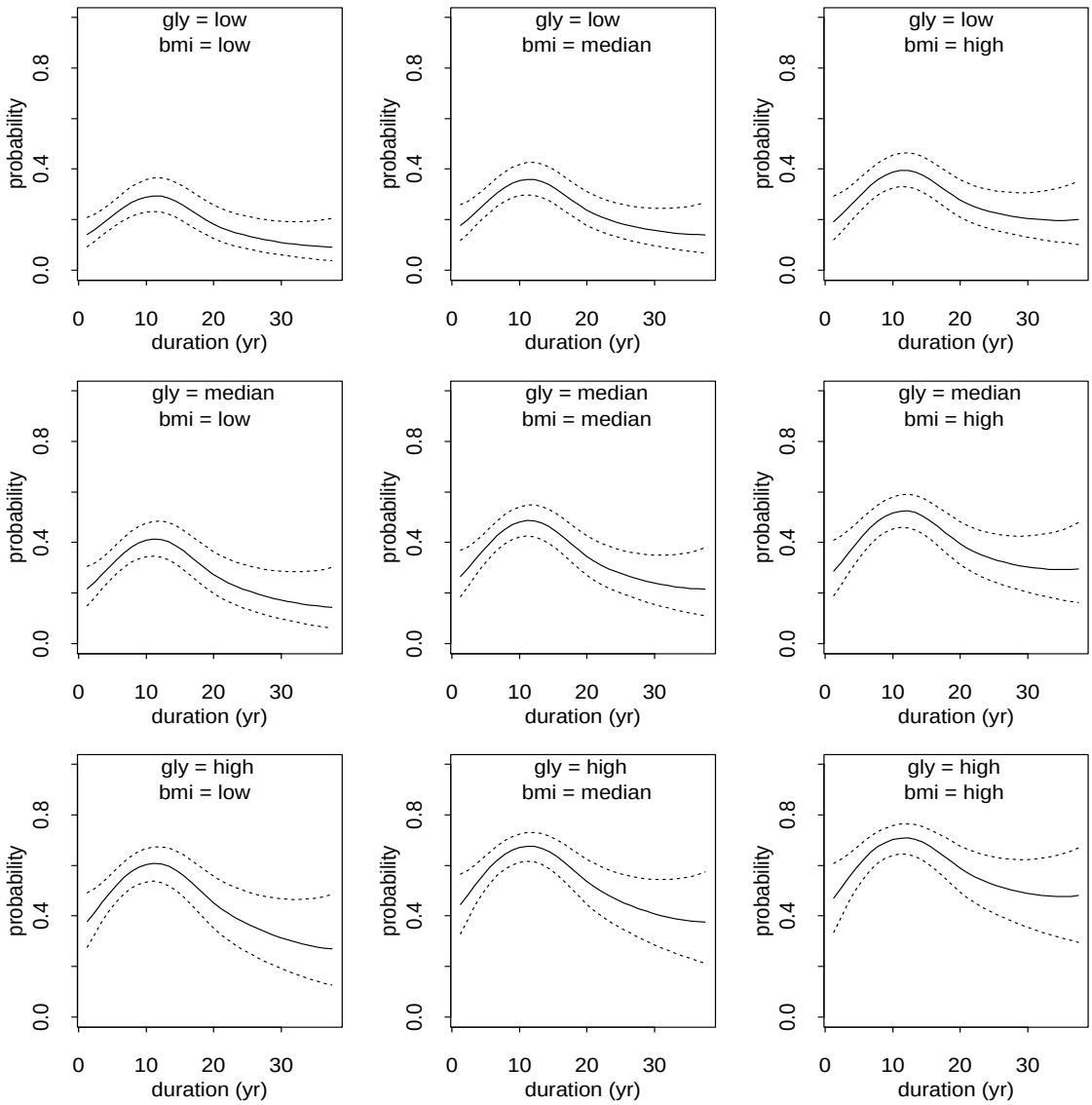


Figure 26: Wesdr Original Data—Cross sections of estimated probability of progression as a function of **duration** with their 90% Bayesian confidence intervals, at three levels of **gly** and three levels of **bmi**. Low, median and high denote .25, .5 and .75 percentiles. Method: OneStepRGACV.

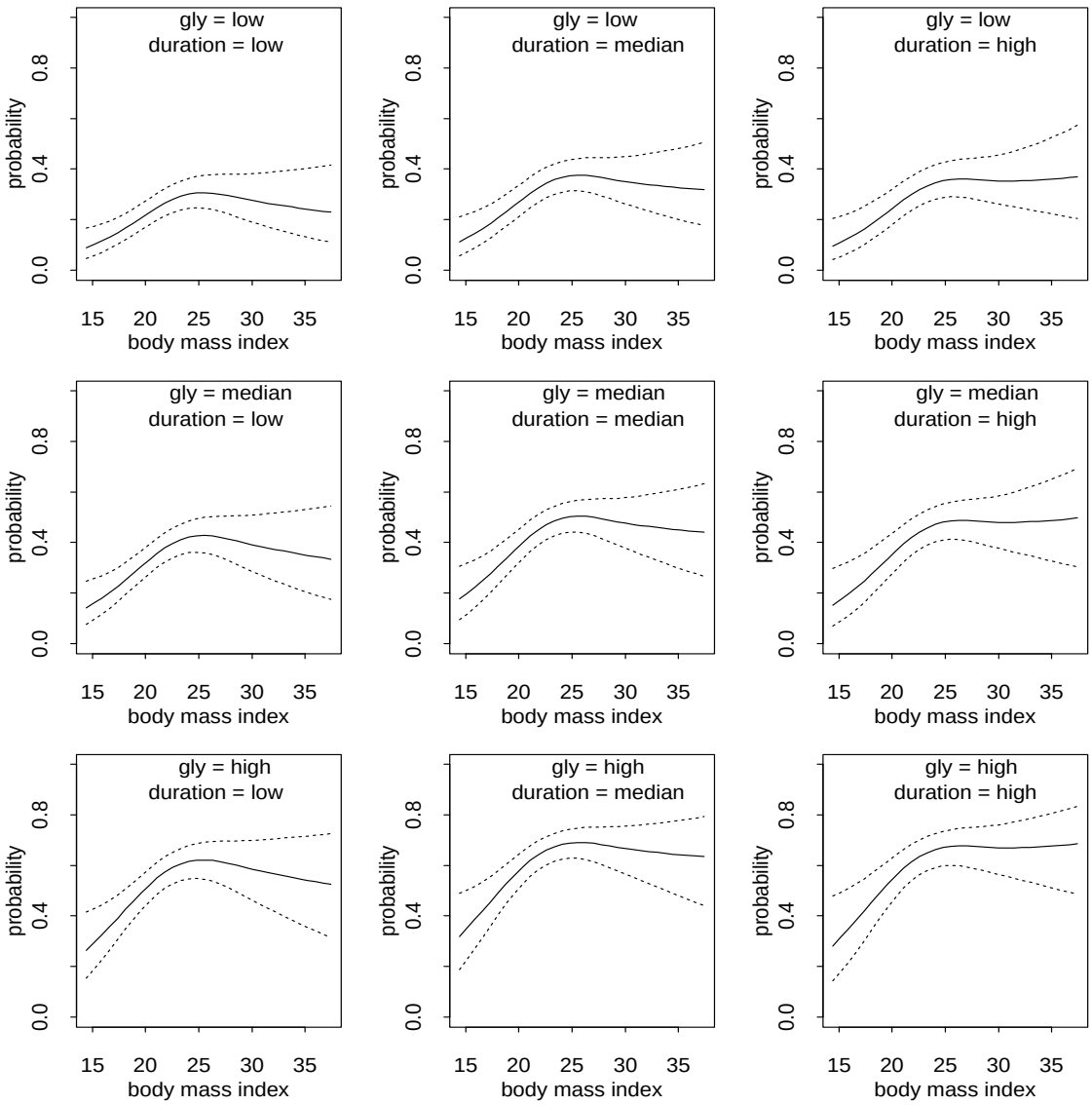


Figure 27: Wesdr Original Data—Cross sections of estimated probability of progression as a function of **bmi** with their 90% Bayesian confidence intervals, at three levels of **gly** and three levels of **duration**. Low, median and high denote .25, .5 and .75 percentiles. Method: OneStepRGACV.

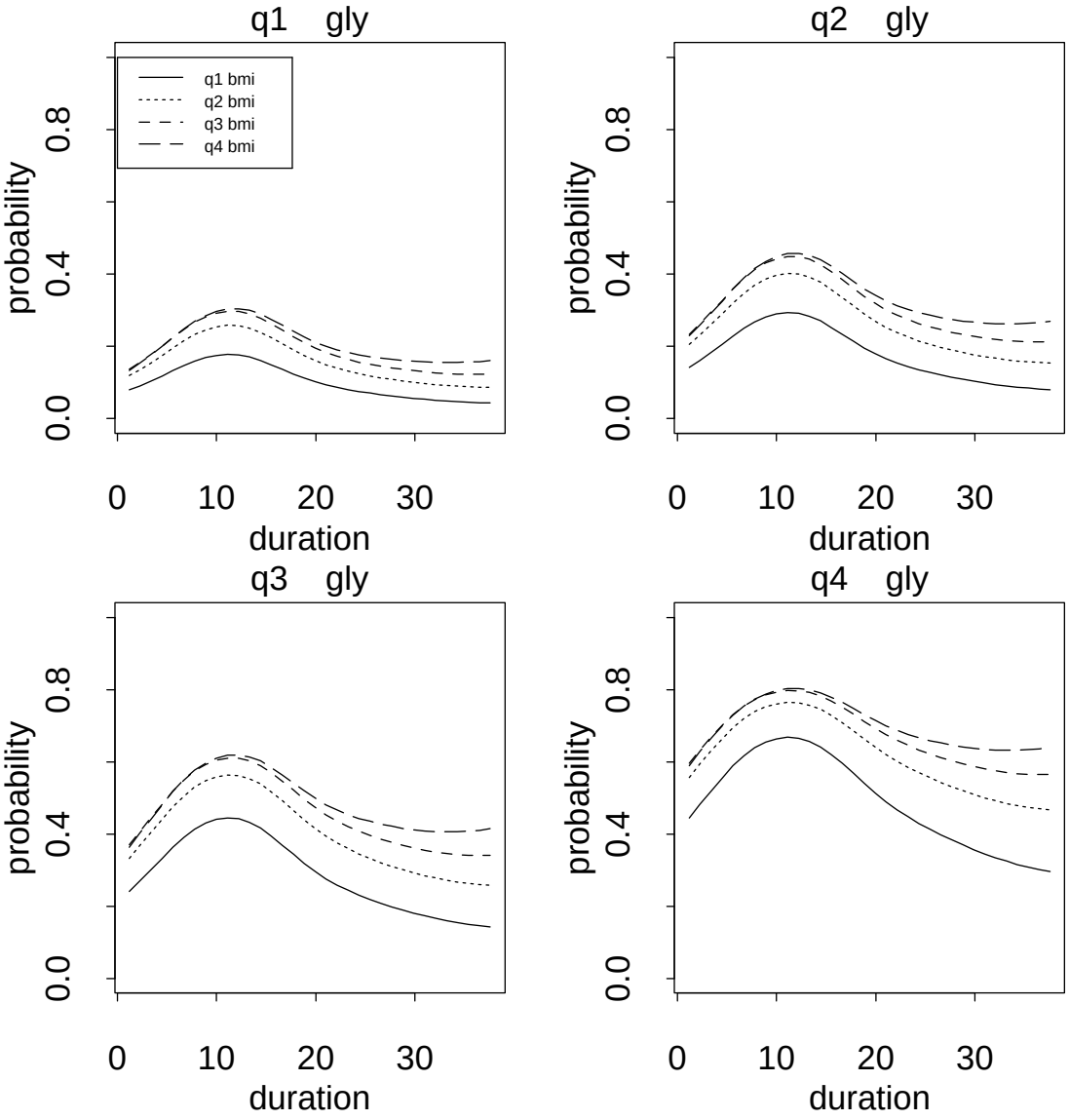


Figure 28: Wesdr Original Data—Cross sections of estimated probability of progression as a function of **duration**, at four levels of **gly** and four levels of **bmi**. q1, q2, q3 and q4 are the quantiles at $0.125, 0.375, 0.625$ and 0.875. Method: OneStepRGACV.

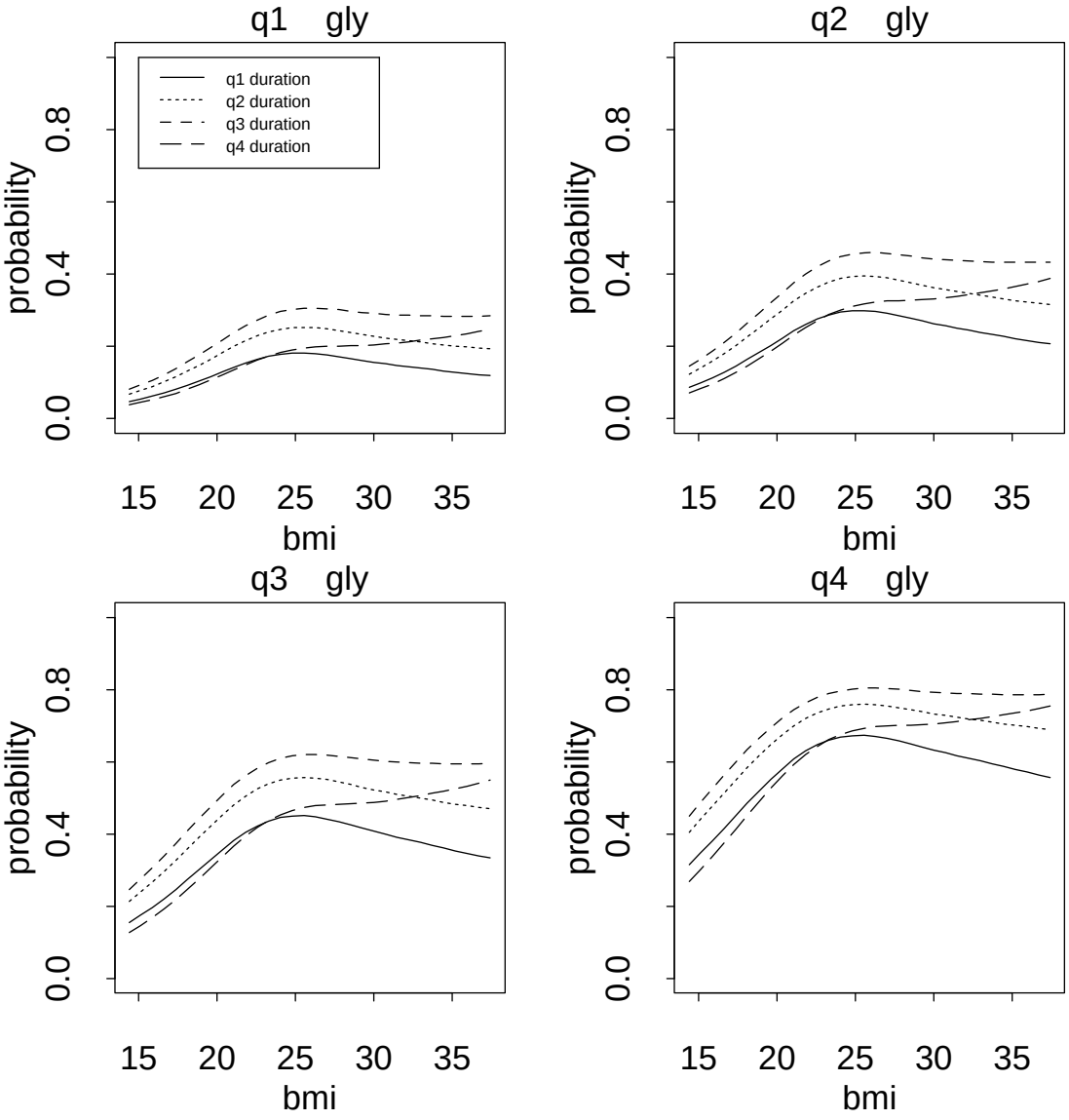


Figure 29: Wesdr Original Data—Cross sections of estimated probability of progression as a function of **bmi**, at four levels of **gly** and four levels of **duration**. q1, q2, q3 and q4 are the quantiles at $0.125, 0.375, 0.625$ and 0.875. Method: OneStepRGACV.

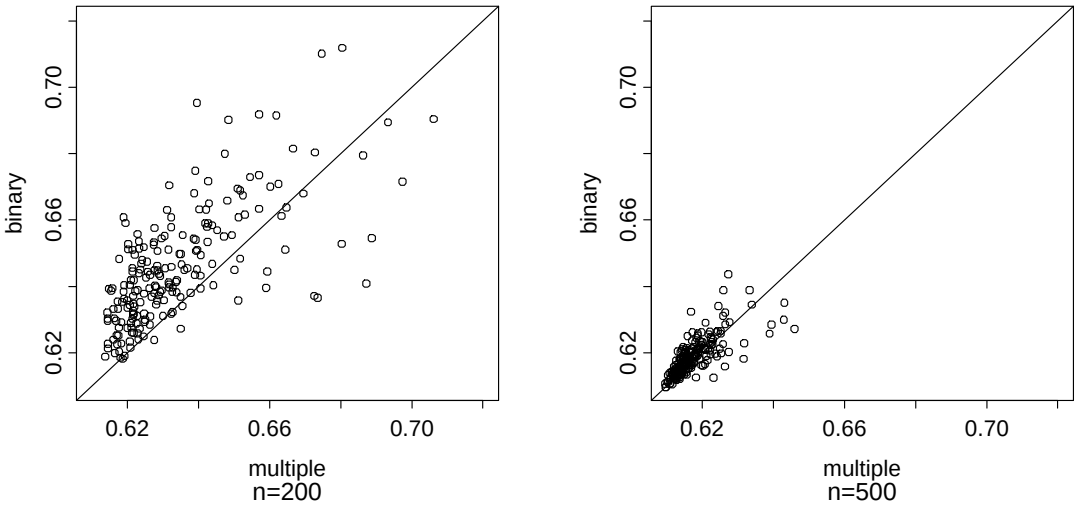


Figure 30: CKL Comparison based on 200 runs.

Similarly for the observations, we denote

$$z_{ij} = \begin{cases} 1 & \text{if} \quad y_j = 1, \\ 0 & \text{otherwise} \end{cases} \qquad i = 0, \cdots, k \quad \text{and} \quad j = 1, \cdots, n. \tag{4.2.4}$$

Note that $\{(X_j, Z_{ij}), j = 1, \cdots, n\}$ is sufficient for $\{p_i(x_j), j = 1, \cdots, n\}$, so the conditional class probability can be estimated from the observed data set $\{(x_j, z_{ij}), j = 1, \cdots, n\}$. Letting $f_i(t) = log(p_i(t)/(1 - p_i(t))$, we can estimate $f_i(t)$ by minimizing the following penalized problem,

$$\sum_{j=1}^{n} [-z_{ij} f_i(x_j) + log(1 + e^{f_i(x_j)})] + \frac{n}{2} J_{\lambda_i}(f_i).$$

This can be done by using the OneStepRGACV method developed earlier in this chapter. Denote the estimates obtained through this way be $\hat{p}_0(t), \cdots, \hat{p}_k(t)$. In order for the estimates to satisfy the constraint (4.2.1), we set the final estimates as follows,

$$\tilde{p}_i(t) = \frac{\hat{p}_i(t)}{\hat{p}_0(t) + \cdots + \hat{p}_k}, \quad i = 1, \cdots, k. \tag{4.2.5}$$

We conduct Monte Carlo simulations to see how the individual fitting by binary data algorithm compared with the penalized polychotomous regression. We apply this approach to the two simulation examples in chapter 3. The comparison is presented in term of Comparative Kullback-Leibler Distance. Figure 30 is for the univariate case, and Figure 31 is for the multivariate case. We use 'multiple' to stand for the penalized polychotomous method proposed in chapter 3 and 'binary' for the individual fitting. From the plots, the penalized polychotomous method seems to perform a little bit better. However, the performances are very close when we have reasonable large sample size. We can expect the performance will get closer as the sample size gets larger.

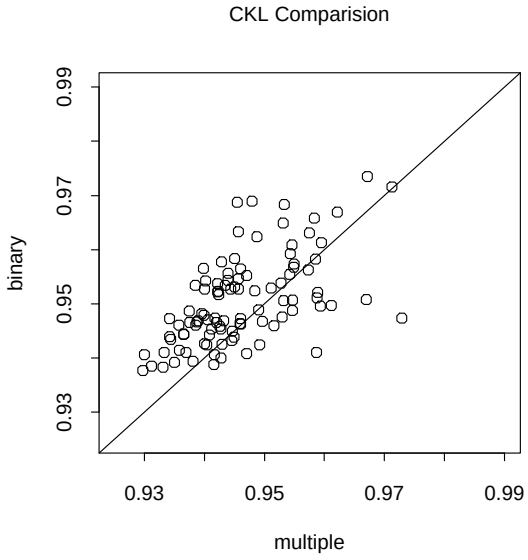


Figure 31: CKL Comparison based on 100 runs. $n = 500$.

4.2.2 Randomized GACV for Penalized Polychotomous Regression

Another possibility to apply the penalized Polychotomous Regression to large data set is to use approximate smoothing spline and select the smoothing parameters by some criteria similar to the OneStepRGACV.

We will first extend the derivation of GACV in Xiang and Wahba (1996) to the Penalized Polychotomous Regression.
Our object is the Kullback-Leibler distance or the Comparative Kullback-Leibler distance between the estimate and the true functions.

$$CKL(\lambda) = \frac{1}{n}\sum_{j=1}^{n}\{-p'(x_j)f_\lambda(x_j) + b(f_\lambda(x_j))\}. \tag{4.2.6}$$

The CKL depends on the true functions which are unknown, so an estimate of the CKL is needed. Define the ordinary, or leaving-out-one cross validation function $CV(\lambda)$,

$$\begin{array}{rcl} CV(\lambda) & = & \frac{1}{n}\sum_{j=1}^{n}[-y_j^{'}f_\lambda^{\perp j}(x_j) + b(f_\lambda(x_j))] \\ & = & \frac{1}{n}\sum_{j=1}^{n}[-y_j^{'}f_\lambda(x_j) + b(f_\lambda(x_j))] + \frac{1}{n}\sum_{j=1}^{n} y_j^{'}(f_\lambda(x_j) - f_\lambda^{\perp j}(x_j)) \end{array}, \tag{4.2.7}$$

where $y_j = (y_{1j}, \cdots, y_{kj})^T$, $f_\lambda(t) = f_{1\lambda}, \cdots, f_{k\lambda})^T$ and $f_\lambda^{\perp j}$ is the minimizer of penalized polychotomous likelihood (2.2.1) with the jth data point omitted. $CV(\lambda)$ can be expected to be at least roughly unbiased for the $CKL(\lambda)$. For any fixed λ, in order to evaluate $CV(\lambda)$, we have to get n leaving-out-one estimates $f_\lambda^{\perp j}(x_j)$, $j = 1, \cdots, n$. In general, it will be very expensive to compute $f_\lambda^{\perp j}$. Hence, using $CV(\lambda)$ is almost infeasible. We will introduce an approximate for $CV(\lambda)$ via several first order Taylor series expansions.

Notice that

$$\begin{array}{l} \sum_{j=1}^{n} y_j^{'}(f_\lambda(x_j) - f_\lambda^{\perp j}(x_j)) \\ = \sum_{i=1}^{k}\sum_{j=1}^{n} y_{ij}(f_{i\lambda}(x_j) - f_{i\lambda}^{\perp j}(x_j)) \\ \sum_{i}^{k}\sum_{j}^{n} y_{ij}\frac{f_{i\lambda}(x_j) \perp f_{i\lambda}^{-j}(x_j)}{y_{ij} \perp p_{i\lambda}^{-j}(x_j)} \frac{y_{ij} \perp p_{i\lambda}(x_j)}{1 \perp \frac{p_{i\lambda}(x_j) - p_{i\lambda}^{-j}(x_j)}{y_{ij} - p_{i\lambda}(x_j)}}, \end{array}$$

and

$$\frac{p_{i\lambda}(x_j) - p_{i\lambda}^{\perp j}(x_j)}{y_{ij} - p_{i\lambda}^{\perp j}(x_j)} \approx b_{ii}^{''}(f(x_j))\frac{f_{i\lambda}(x_j) - f_{i\lambda}^{\perp j}(x_j)}{y_{ij} - p_{i\lambda}^{\perp j}(x_j)}.$$

Hence we have

$$CV(\lambda) \approx \frac{1}{n}\sum_{j=1}^{n}[-y_j^{'}f_\lambda^{\perp j}(x_j) + b(f_\lambda(x_j))] + \frac{1}{n}\sum_{i=1}^{k}\sum_{j=1}^{n}\frac{y_{ij}(y_{ij} - p_{i\lambda}(x_j))}{\frac{y_{ij} \perp p_{i\lambda}^{-j}(x_j)}{f_{i\lambda}(x_j) \perp f_{i\lambda}(x_j)} - b_{ii}^{''}(f_\lambda(x_j))}. \tag{4.2.8}$$

We can see from the right side of (4.2.8) that the calculation of $CV(\lambda)$ will focus on the calculation of

$$\frac{y_{ij} - p_{i\lambda}(x_j)}{f_{i\lambda}(x_j) - f_{i\lambda}^{\perp j}(x_j)}. \tag{4.2.9}$$

To avoid the calculation of (4.2.9), we will develop an approximation for it. Before obtaining an approximation for this ratio, we need to generalize the leaving-out-one lemma of Graven and Wahba (1979).

Lemma 4.10 *(leaving-out-one lemma) Let*

$$I_\lambda(f,y) = -l(y_j, f(x_j)) - \sum_{l\neq j} l(y_l, f(x_l)) + \frac{n}{2} J_\lambda(f).$$

Suppose $h_\lambda(j,z,\cdot)$ is the minimizer in $\mathcal{H}$ of $I_\lambda(f,z)$ where

$$z = (y_1, \cdots, y_{j\perp 1}, z, y_{j+1}, \cdots, y_n)^T.$$

Then $h_\lambda(j, p^{\perp j}(x_j), \cdot) = f^{\perp j}(\cdot)$ where $f^{\perp j}(\cdot)$ is the minimizer of

$$-\sum_{l\neq j} l(y_l, f(x_l)) + \frac{n}{2} J_\lambda(f)$$

and $p^{\perp j}(\cdot)$ is the probability function corresponding to $f^{\perp j}(\cdot)$.

Proof First define

$$y^{\perp j} = (y_1, \cdots, y_{j\perp 1}, p^{\perp j}(x_j), y_{j+1}, \cdots, y_n)^T$$

and

$$-l(p^{\perp j}(x_j), \tau) = -[p^{\perp j}(x_j)]'\tau + b(\tau).$$

We will show that

$$-l(p^{\perp j}(x_j, f^{\perp j}(x_j) \leq -l(p^{\perp j}(x_j), f(x_j)). \tag{4.2.10}$$

$$\frac{\partial l(p^{\perp j}(x_j),\tau)}{\partial \tau} = -p^{\perp j}(x_j) + \frac{\partial b(\tau)}{\partial \tau}$$

and using the fact

$$\frac{\partial^2 b(\tau)}{\partial \tau^T \partial \tau} > 0$$

implies that $-l(p^{\perp j}(x_j,\tau)$ achieves its minimum for $\frac{\partial b(\tau)}{\partial \tau} = p^{\perp j}(x_j)$. So (4.2.10) holds since

$$\frac{\partial b(\tau)}{\partial \tau}|_{\tau = f^{-j}(x_j)} = p^{\perp j}(x_j).$$

Then, for any f

$$\begin{aligned} &I_\lambda(f, y^{\perp j}) = -l(p^{\perp j}(x_j), f(x_j)) - \textstyle\sum_{l\neq j} l(y_l, f(x_l)) + \frac{n}{2} J_\lambda(f) \\ &\geq -l(p^{\perp j}(x_j), f^{\perp j}(x_j)) - \textstyle\sum_{l\neq j} l(y_l, f(x_j)) + \frac{n}{2} J_\lambda(f) \\ &\geq -l(p^{\perp j}(x_j, f^{\perp j}(x_j)) - \textstyle\sum_{l\neq j} l(y_l, f^{\perp l}(x_l)) + \frac{n}{2} J_\lambda(f^{\perp j}). \end{aligned}$$

So $h_\lambda(j, p^{\perp j}(x_j), \cdot) = f^{\perp j}(\cdot)$. **Q.E.D.**

From this lemma, we can see that replacing $y_j = (y_{1j}, \cdots, y_{kj})^T$ by $p_\lambda^{\perp j}(x_j)$, the minimizer of I_λ with respect to $f(\cdot)$ will be $f_\lambda^{\perp j}(\cdot)$. From Chapter 2, we know that if $(f_{1\lambda}, \cdots, f_{k\lambda})^T$ is a minimizer of I_λ, $f_{i\lambda}$ is in a certain linear space of dimension at most n, and then $J_i(f_{i\lambda})$ can be written as a quadratic form in its values at x_j. With some abuse of notation we will write below $J_i(f_i) = f_i^T \Sigma_i f_i = c_i^T Q_i c_i$ where in this context we are letting $f_i = (f_i(x_1), \cdots, f_i(x_n))^T$. Hence, I_λ can be written as follows,

$$I_\lambda(f,y) = -\sum_{j=1}^{n} \{\sum_{i=1}^{k} y_{ij} f_i(x_j) + log(1 + \sum_{i=1}^{k} e^{f_i(x_j)})\} + \frac{n}{2} \sum_{i=1}^{k} \lambda_i f_i^T \Sigma_i f_i. \tag{4.2.11}$$

Let $Y_i = (y_{i1}, \cdots, y_{in})^T$, $Y_i^{\perp j} = (y_{i1}, \cdots, y_{i(j\perp 1)}, p_i^{\perp j}(x_j), y_{i(j\perp 1)}, \cdots, y_{in})^T$ and $Y^{\perp j} = ((Y_1^{\perp j})^T, \cdots, (Y_k^{\perp j})^T)^T$. Because (f_λ, Y) and $(f_\lambda^{\perp j}, Y^{\perp j})$ are two local minimizers of $I_\lambda(f, y)$, we have

$$\frac{\partial I_\lambda(f,y)}{\partial f_i}|_{f=f_\lambda, y=Y} = 0,$$

and

$$\frac{\partial I_\lambda(f,y)}{\partial f_i}|_{f=f_\lambda^{-j}, y=Y^{-j}} = 0,$$

hence

$$\frac{\partial I_\lambda(f,y)}{\partial f_i}|_{f_i=f_{i\lambda}, f_l=f_{l\lambda}^{-j}, y_i=Y_i, y_l=Y_l^{-j}} \approx \frac{\partial I_\lambda(f,y)}{\partial f_i}|_{f=f_\lambda, y=Y} = 0.$$

Also, we have

$$\frac{\partial^2 I_\lambda}{\partial f_i^T \partial f_i} = W_i + n\lambda_i \Sigma_i,$$

$$\frac{\partial^2 I_\lambda}{\partial y_i^T \partial y_i} = 0,$$

and

$$\frac{\partial^2 I_\lambda}{\partial y_i \partial f_i^T} = -I_{n\times n}.$$

Hence, using a Taylor expansion we have

$$0 \approx (W_i + n\lambda_i\Sigma_i)(f_{i\lambda} - f_{i\lambda}^{\perp j}) - (Y_i - Y_i^{\perp j}),$$

thus

$$f_{i\lambda} - f_{i\lambda}^j \approx (W_i + n\lambda_i\Sigma_i)^{\perp 1}(Y_i - Y_i^{\perp j}),$$

so

$$\begin{pmatrix} f_{i\lambda}(x_1) - f_{i\lambda}^{\perp j}(x_1) \\ \vdots \\ f_{i\lambda}(x_j) - f_{i\lambda}^{\perp j}(x_j) \\ \vdots \\ f_{i\lambda}(x_n) - f_{i\lambda}^{\perp j}(x_n) \end{pmatrix} \approx (W_i + n\lambda_i\Sigma_i)^{\perp 1} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ y_{ij} - p_{i\lambda}^j(x_j) \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{4.2.12}$$

Defining $H_i = (W_i + n\lambda_i\Sigma_i)^{\perp 1}$, and h_i^{jj} be the diagonal element of H_i, we will have

$$\frac{f_{i\lambda}(x_j) - f_{i\lambda}^{\perp j}(x_j)}{y_{ij} - p_{i\lambda}^{\perp j}(x_j)} \approx h_i^{jj}. \tag{4.2.13}$$

Using (4.2.13) to calculate the $CV(\lambda)$ we will have an approximate formula for the cross validation,

$$ACV(\lambda) = \frac{1}{n}L + \frac{1}{n}\sum_i^k \sum_{j=1}^n \frac{y_{ij}(y_{ij} - p_{i\lambda}(x_j))}{(h_i^{jj})^{\perp 1} - w_{ij}}. \tag{4.2.14}$$

If we replace h_i^{jj} by $\frac{1}{n}tr(H_i)$ and replace $h_i^{jj}w_{ij}$ by $\frac{1}{n}tr(W_i^{1/2}H_iW_i^{1/2})$, we have the generalized form of the approximate cross-validation as follows,

$$\begin{aligned} GACV(\lambda) = \quad & \tfrac{1}{n}\textstyle\sum_{j=1}^{n}\{-\sum_{i=1}^{k} y_{ij}f_{i\lambda}(x_j) + log(1+\sum_{i=1}^{k} e^{f_{i\lambda}(x_j)})\} \\ & +\tfrac{1}{n}\textstyle\sum_{i=1}^{k}\sum_{j=1}^{n}\frac{tr(H_i)y_{ij}(y_{ij}\perp p_{i\lambda}(x_j))}{n\perp tr(W_i^{1/2}H_iW_i^{1/2})}. \end{aligned} \tag{4.2.15}$$

We can see that the GACV formula will reduce to the formula for binary case when $k = 1$. As mentioned earlier, the computation of H_i will be numerical unstable when the sample size is large. Numerical method should be sought to overcome this difficulty if we want to use this approach in practice. Considering the disturbance $\epsilon \sim N(\underline{0}, \sigma^2 I_{nk})$ and letting $\epsilon = (\epsilon_1^T, \cdots, \epsilon_k^T)^T$, we will have $E(\epsilon_i^T H_i \epsilon_i) = \sigma^2 Tr(H_i)$ and $E(\epsilon_i^T W_i H_i \epsilon_i) = \sigma^2 Tr(W_i^{1/2} H_i W_i^{1/2})$. Hence, we can use $\epsilon_i^T H_i \epsilon_i/\sigma^2$ to estimate $Tr(H_i)$ and $\epsilon_i^T W_i H_i \epsilon_i/\sigma^2$ to estimate $Tr(W_i^{1/2} H_i W_i^{1/2})$.

Let

$$f_{i\lambda}^{y+\epsilon,1} = f_{i\lambda} - (\frac{\partial^2 I_\lambda}{\partial f_i^T \partial f_i}(f_\lambda, y+\epsilon))^{\perp 1}\frac{\partial I_\lambda}{\partial f_i}(f_\lambda, y+\epsilon).$$

By observing that

$$\frac{\partial I_\lambda}{\partial f_i}(f_\lambda, y+\epsilon) = -\epsilon_i + \frac{\partial I_\lambda}{\partial f_i}(f_\lambda, y) = -\epsilon_i,$$

and

$$[\frac{\partial^2 I_\lambda}{\partial f_i^T \partial f_i}]^{\perp 1} = [\frac{\partial^2 I_\lambda}{\partial f_i^T \partial f_i}(f_\lambda, y)]^{\perp 1} = H_i,$$

we have

$$f_{i\lambda}^{y+\epsilon,1} - f_{i\lambda} = H_i\epsilon_i. \tag{4.2.16}$$

Thus $Tr(H_i)$ can be estimated by $\epsilon_i^T(f_{i\lambda}^{y+\epsilon,1} - f_{i\lambda})/\sigma^2$, and $Tr(W_i^{1/2}H_iW_i^{1/2})$ can be estimated by $\epsilon_i^T W_i(f_{i\lambda}^{y+\epsilon,1} - f_{i\lambda})/\sigma^2$.

By replacing $Tr(H_i)$ and $Tr(W_i^{1/2}H_iW_i^{1/2})$ with their randomized estimates and use $\epsilon^T\epsilon/nk$ to estimate σ^2, we have a randomized version of GACV function for the penalized polychotomous regression,

$$\begin{aligned} ranGACV(\lambda) = \quad & \tfrac{1}{n}\textstyle\sum_{j=1}^{n}\{-\sum_{i=1}^{k} y_{ij}f_{i\lambda}(x_j) + log(1+\sum_{i=1}^{k} e^{f_{i\lambda}(x_j)})\} \\ & +\tfrac{1}{n}\textstyle\sum_{i=1}^{k}\sum_{j=1}^{n}\frac{\epsilon_i^T(f_{i\lambda}^{y+\epsilon,1}\perp f_{i\lambda})y_{ij}(y_{ij}\perp p_{i\lambda}(x_j))}{\epsilon_i^T\epsilon_i/k\perp\epsilon_i^T W_i(f_{i\lambda}^{y+\epsilon,1}\perp f_{i\lambda})} \end{aligned} \tag{4.2.17}$$

This will reduce to the OneStepRGACV formula for binary data when $k = 1$. To reduce the variance of the $ranGACV$, we may draw R independent replicates $\epsilon^{(r)}, r = 1, \cdots, R$ and obtain an R-replicate version randomized GACV,

$$\begin{aligned} ranGACV_R(\lambda) = \quad & \tfrac{1}{n}\textstyle\sum_{j=1}^{n}\{-\sum_{i=1}^{k} y_{ij}f_{i\lambda}(x_j) + log(1+\sum_{i=1}^{k} e^{f_{i\lambda}(x_j)})\} \\ & +\tfrac{1}{nR}\textstyle\sum_{r=1}^{R}\sum_{i=1}^{k}\sum_{j=1}^{n}\frac{(\epsilon_i^{(r)})^T(f_{i\lambda}^{y+\epsilon^{(r)},1}\perp f_{i\lambda})y_{ij}(y_{ij}\perp p_{i\lambda}(x_j))}{(\epsilon^{(r)})^T\epsilon^{(r)}/k\perp(\epsilon_i^{(r)})^T W_i(f_{i\lambda}^{y+\epsilon,1}\perp f_{i\lambda})}. \end{aligned} \tag{4.2.18}$$

For fixed λ, we can iterate the block one step SOR-newton until it converges to get a solution f_λ, and evaluate $ranGACV(\lambda)$. Then we can find the minimizer $\hat{\lambda}$ of $ranGACV(\lambda)$, and use $f_{\hat{\lambda}}$ as our estimate. In order to apply this to large data set, the approximate smoothing spline method should also be used. Besides, for the polychotomous problem we have more smoothing parameters than the binary case. Before we can apply this in practice, we should put some effort in reducing the number of smoothing parameters or investigating some efficient way to find the minimizer of the $ranGACV$ function.

Chapter 5

Application to Wisconsin Epidemiological Study of Diabetic Retinopathy

5.1 Introduction

In this chapter, we use data from the Wisconsin Epidemiology Study of Diabetic Retinopathy (WESDR) to demonstrate the penalized polychotomous regression method.

The study area is composed of 11 counties in southern Wisconsin. Diabetic persons were identified by a review of the records of 452 of the 457 physicians providing primary care to diabetic persons in the period July 1, 1979, through June 30, 1980. A two part sample of 2990 diabetic patients was selected on July 1, 1980, for the examination phase of study. The first part consisted of all persons whose conditions were diagnosed before 30 years of age and who were taking insulin, referred to as younger-onset persons ($N = 1210$). The second part consisted of a probability sample stratified by duration of diabetes of persons diagnosed by a physician as having diabetes at or after age 30 years and confirmed by a random or postprandial serum glucose level of at least 11.1 mmol/L (200 mg/dL) or a fasting serum glucose level of at least 7.8 mmol/L (140 mg/dL) on at least two occasions, referred to as older-onset persons ($N = 1780$). Of the older-onset group, 824 were taking insulin and 956 were not taking insulin. The sampled persons were invited to participate in the examination phase of the study from 1980 to 1982. Baseline examinations were obtained for 996 (82.3%) younger-onset and 1370(77.0%) older-onset persons.

One of the original aims of the Wisconsin Epidemiologic Study of Diabetic Retinopathy was to examine mortality in the population. Thus, all sampled persons are contacted annually by telephone to determine vital status. In addition, designated contact persons, relatives, and physicians are contacted, and newspaper obituaries are reviewed daily. In all cases, an attempt is made to obtained an exact or approximate date of death. Annually, a request is made to Wisconsin Center for Health Statistics, Section of Vital Statistics, for death certificate information of these persons. In addition, persons who are not known to be dead but have been unavailable for follow-up are submitted for matching against the death records. Wisconsin death records through March 1995 have been searched. Information on persons who have moved out of Wisconsin and are suspected of being dead and persons who are unavailable for follow-up is submitted to the National Death Index for matching against national death data. When a match is made, a copy of the death certificate is obtained from the appropriate state.

All medical conditions on the Wisconsin death certificate were coded by trained nosologists in the Wisconsin Division of Health using the International Classification of Diseases, Ninth Revision (ICD-9). The underlying cause of death was selected by the Automated Classification of Medical Entities computer program. Out-of-state certificates were coded and processed in the same manner. The cause-specific mortality analysis of the present investigation is based on the underlying cause of death.

For this study population, diabetes and heart disease are among the several major causes for mortality. As an example, we will employ the penalized polychotomous regression method to investigate the associate between the risk factors and the cause-specific mortality such as dying of diabetes and dying of heart disease.

5.2 Estimate the Risks of Cause-specific Mortality by Penalized Polychotomous Regression

We are going to investigate the how the risk factors. We only consider older onset without taking insulin group in this analysis. Based on the previous investigation by other researchers and some preliminary analysis using multiple logistic regression, we decide to include the following variables in our analyses:

1. **Age:** age in years at the time of baseline examination;
2. **Glycosylated hemoglobin:** a measure of hyperglycemia;
3. **Systolic blood pressure** in mmHg;

We are concerned about the cause-specific mortality. Specifically, the participants will belong to one of the following categories:

1. Die of diabetes;
2. Die of heart disease;
3. Die of other cause other than diabetes and heart disease;
4. Still alive

Three kinds of mortality will be considered:

1. **5 years mortality:** only those patients who died within 5 years from baseline examination are considered to be death while those patients died after 5 years or still alive are considered to be alive;
2. **10 years mortality:** only those patients who died within 10 years from the baseline examination are considered to be death;
3. **12 years mortality:** only those patients who died within 12 years from the baseline examination are considered to be death.

By deleting the incomplete observations, we summarize the data in Table 1, 2 and 3. Table 1 is for **5 years mortality**, Table 2 is for **10 years mortality** and Table 3 is for **12 years mortality**. The values in the columns under the 'gly', 'sp' and 'age' are the corresponding means for each group.

The polychotomous response for the data set is defined as follows. The patients who died of diabetes as category 1, those died of heart disease as category 2 (these two causes are most commonly fond in diabetes patients), those died of other causes as category 3 and the rest as category 0. The penalized polychotomous regression method is used to build the models for the **5 years mortality**, **10 years mortality** and **12 years mortality** respectively. The covariates considered are:

Table 1: 5 years mortality summary

cause	N	gly	sp	age
diabetes	12	11.23	157	73.17
heart disease	99	11.01	157.78	73.85
other causes	83	10.30	150.14	75.43
alive	452	10.12	145.96	65.24

Table 2: 10 years mortality summary

cause	N	gly	sp	age
diabetes	23	10.95	159.30	70.43
heart disease	155	10.65	155.23	73.06
other causes	164	10.31	152.12	73.25
alive	304	10.07	142.33	62.43

1. **gly1:** glycosylated hemoglobin level at the baseline examination;
2. **sp1:** systolic blood pressure measured at the baseline examination;
3. **age:** age at the baseline examination.

Let

$$f_1(age, gly1, sp1) = log(p_1(age, gly1, sp1)/p_4(age, gly1, sp1)),$$
$$f_2(age, gly1, sp1) = log(p_2(age, gly1, sp1)/p_4(age, gly1, sp1)),$$
$$f_3(age, gly1, sp1) = log(p_3(age, gly1, sp1)/p_4(age, gly1, sp1)).$$

The functional ANOVA decomposition for f_1 is as follows,

$$f_1(age, gly1, sp1) = \mu_1 + h_1(age) + g_1(gly1) + g_2(sp2) + g_{12}(gly1, sp1)$$

and decomposition for f_2 and f_3 are similar. We fit these functions by the penalized polychotomous method proposed in chapter 3. The estimates are plotted in Figure 32 to Figure 37. From the plots, we can see that

1. For those patients with age less than 55, the diabetes is the leading cause of death;
2. For those patients with very high systolic blood pressure ($> 200mmHg$) at baseline examination, heart disease seems to be the leading cause of death within 5 years from baseline.

Table 3: 12 years mortality summary

cause	N	gly	sp	age
diabetes	25	10.96	158.92	69.75
heart disease	172	10.64	154.23	72.54
other causes	188	10.31	151.22	72.85
alive	260	10.00	141.89	61.44

Also, the effect of glycosylated hemoglobin level always turns out to be linear which is consistent with other analysis on the WESDR data.

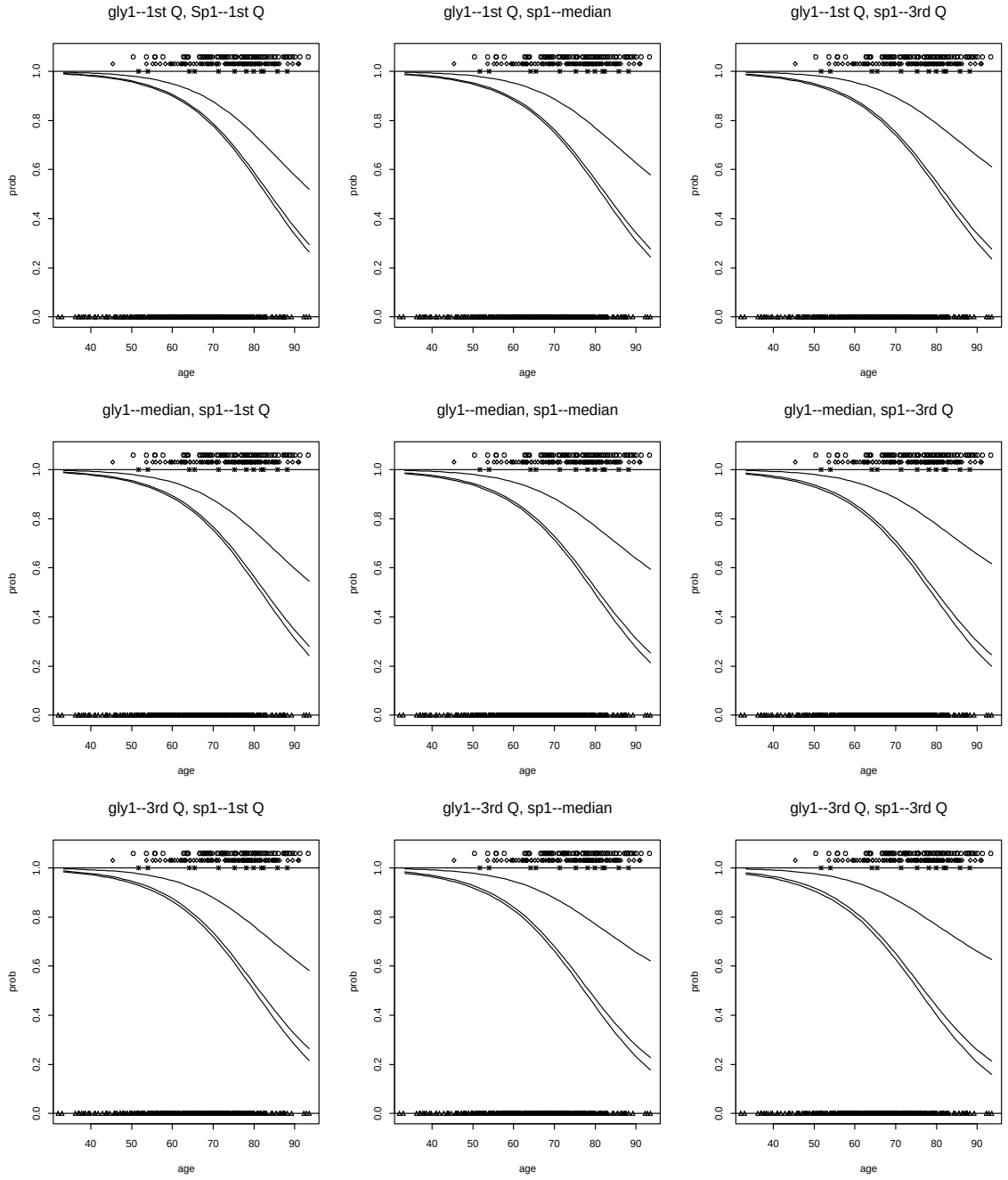


Figure 32: Cross-section plot of probability surfaces. In each plot, the differences between adjacent curves (from bottom to top) are probabilities for: alive, diabetes, heart attack, other cause respectively. The points imposed are in the same order. Older onset without taking insulin, those who died after 5yrs from baseline are considered to be alive. $n = 646$.

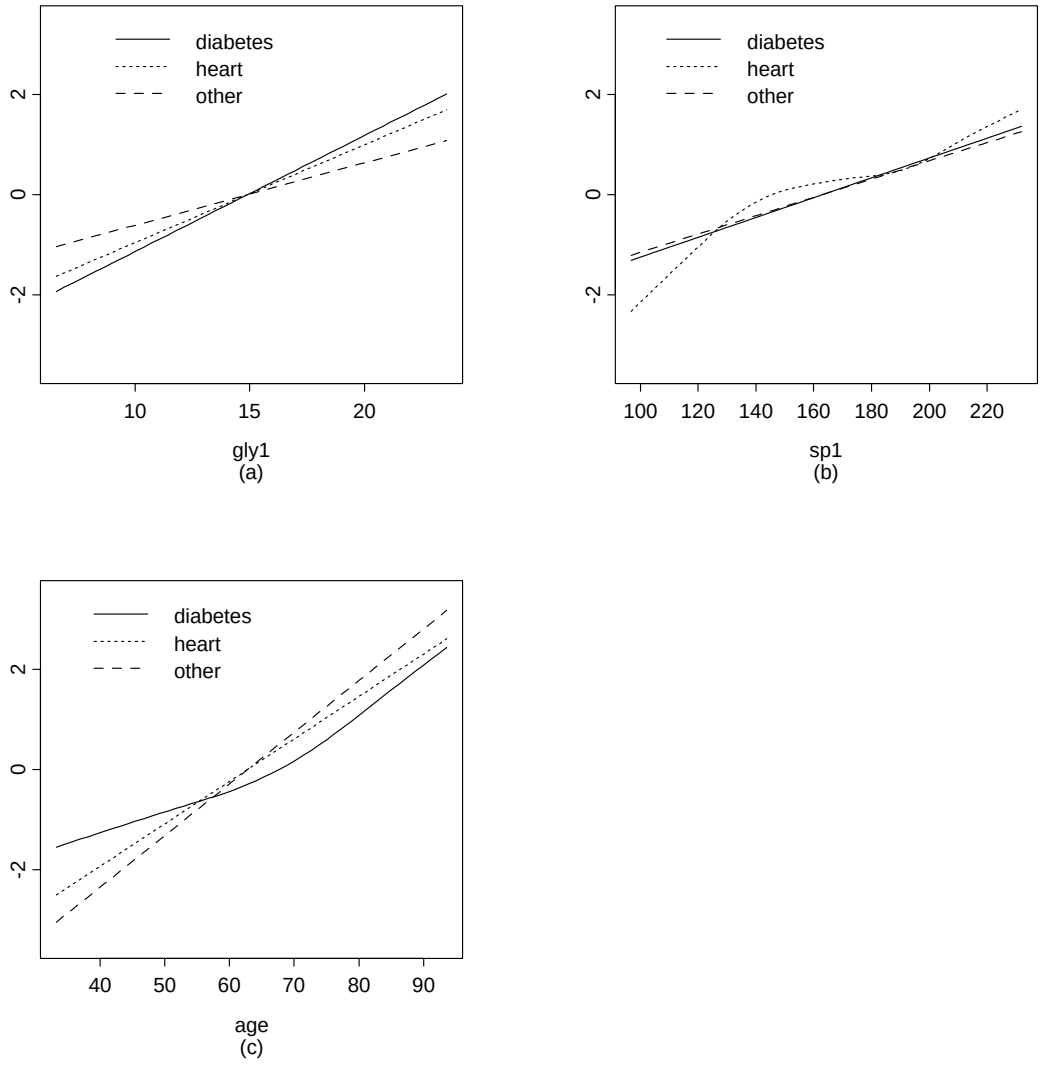


Figure 33: Main-effect plots in logit scale (y-axis corresponding to the value of logit function). Older onset without taking insulin group. Those who died after 5yrs from baseline are considered to be alive. $n = 646$.

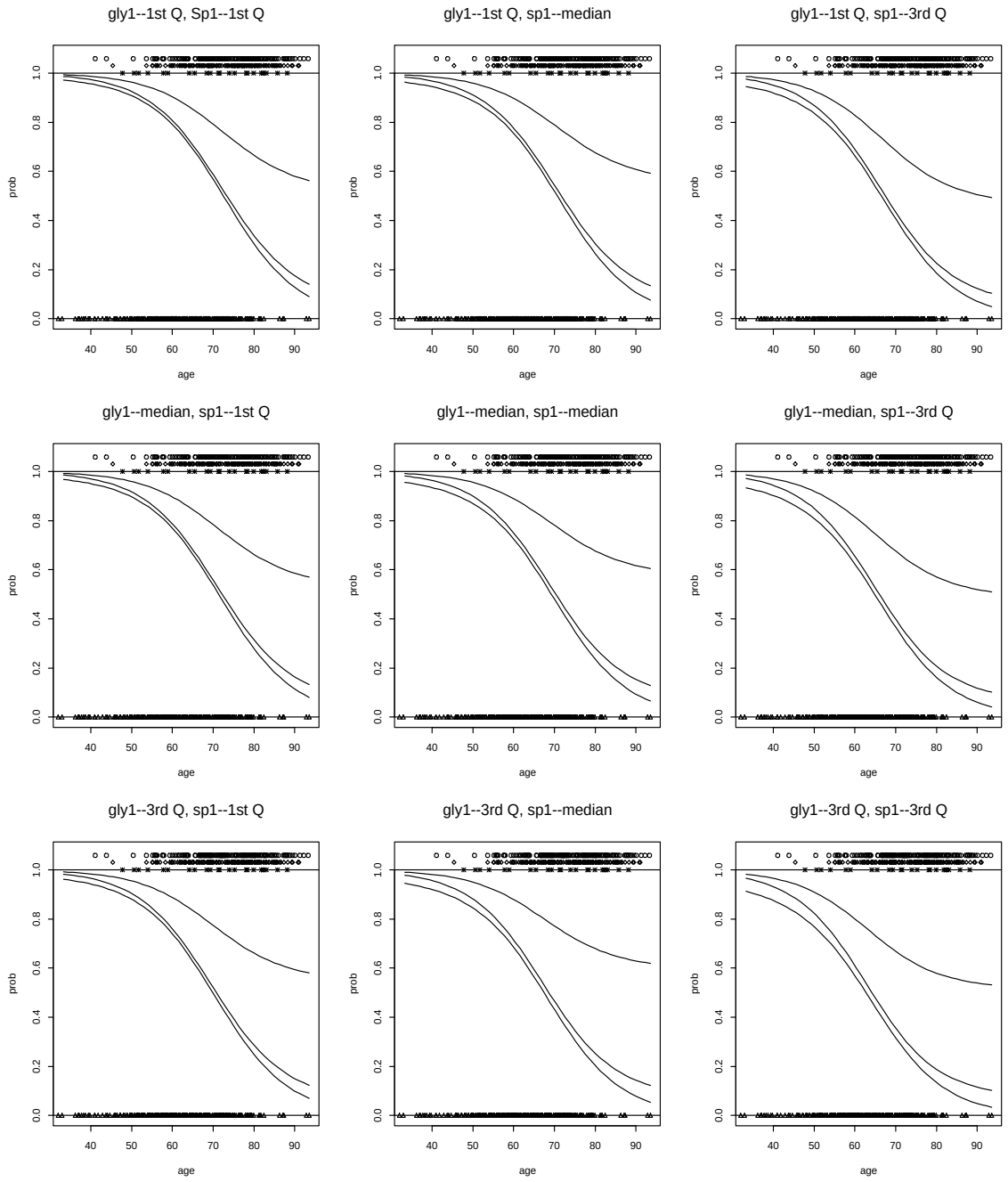


Figure 34: Cross-section plot of probability surfaces. In each plot, the differences between adjacent curves (from bottom to top) are probabilities for : alive, diabetes, heart attack, other cause respectively. The points imposed are in the same order. Older onset without taking insulin, those who died after 10yrs from baseline are considered to be alive. $n = 646$.

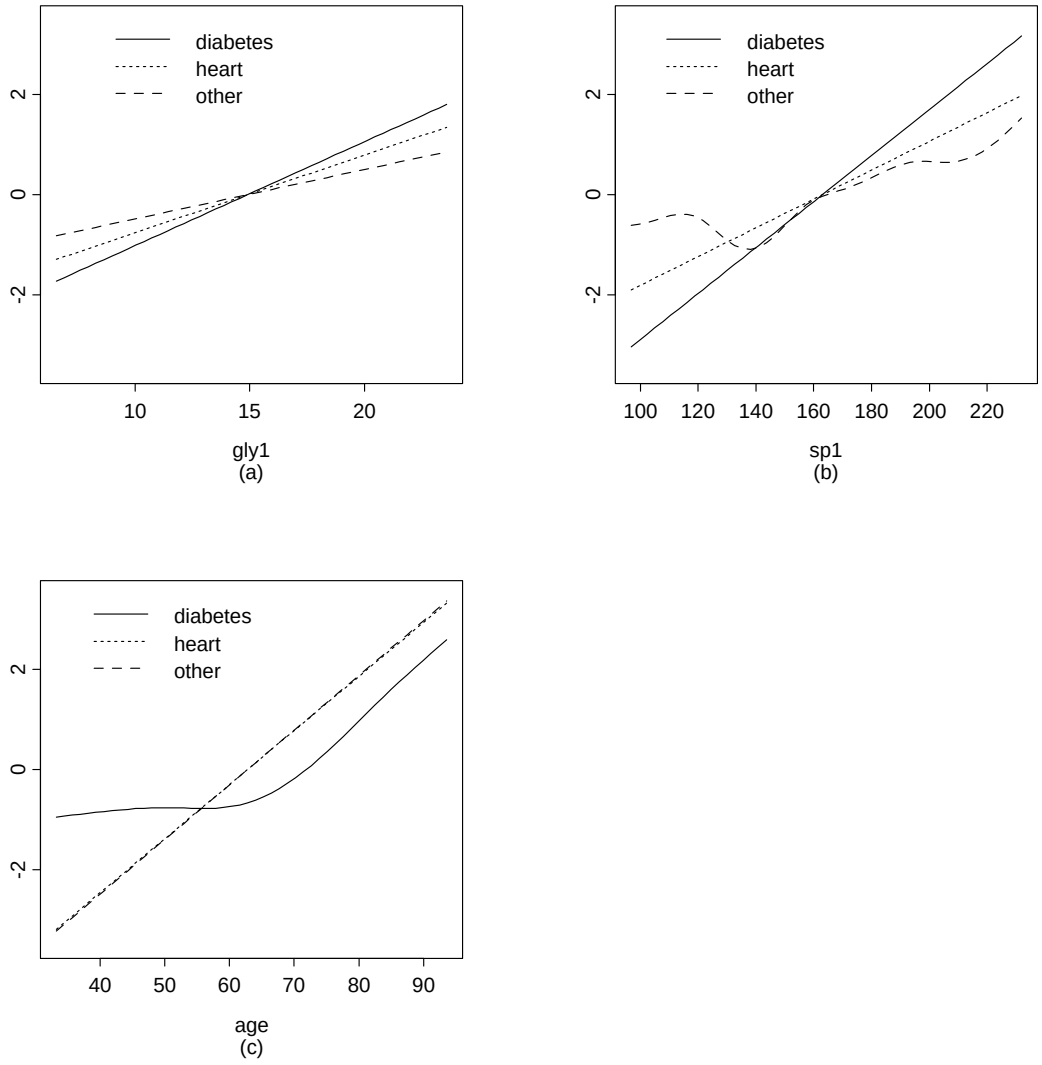


Figure 35: Main-effect plots in logit scale (y-axis corresponding to the value of logit function). Older onset without taking insulin group. Those who died after 10yrs from baseline are considered to be alive. $n = 646$.

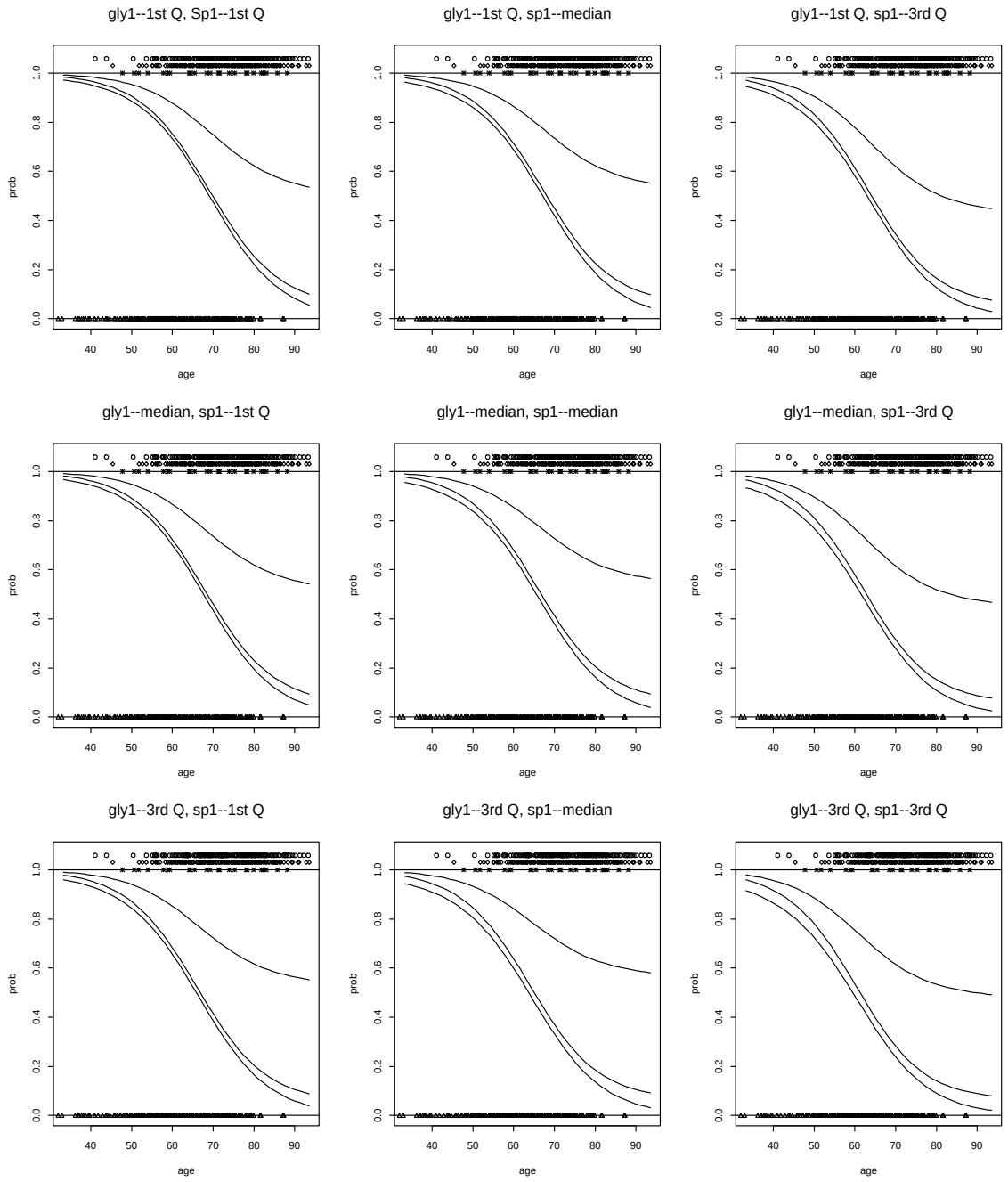


Figure 36: Cross-section plot of probability surfaces. In each plot, the differences between adjacent curves (from bottom to top) are probabilities for: alive, diabetes, heart attack, other cause respectively. The points imposed are in the same order. Older onset without taking insulin, those who died after 12yrs from baseline are considered to be alive. $n = 646$.

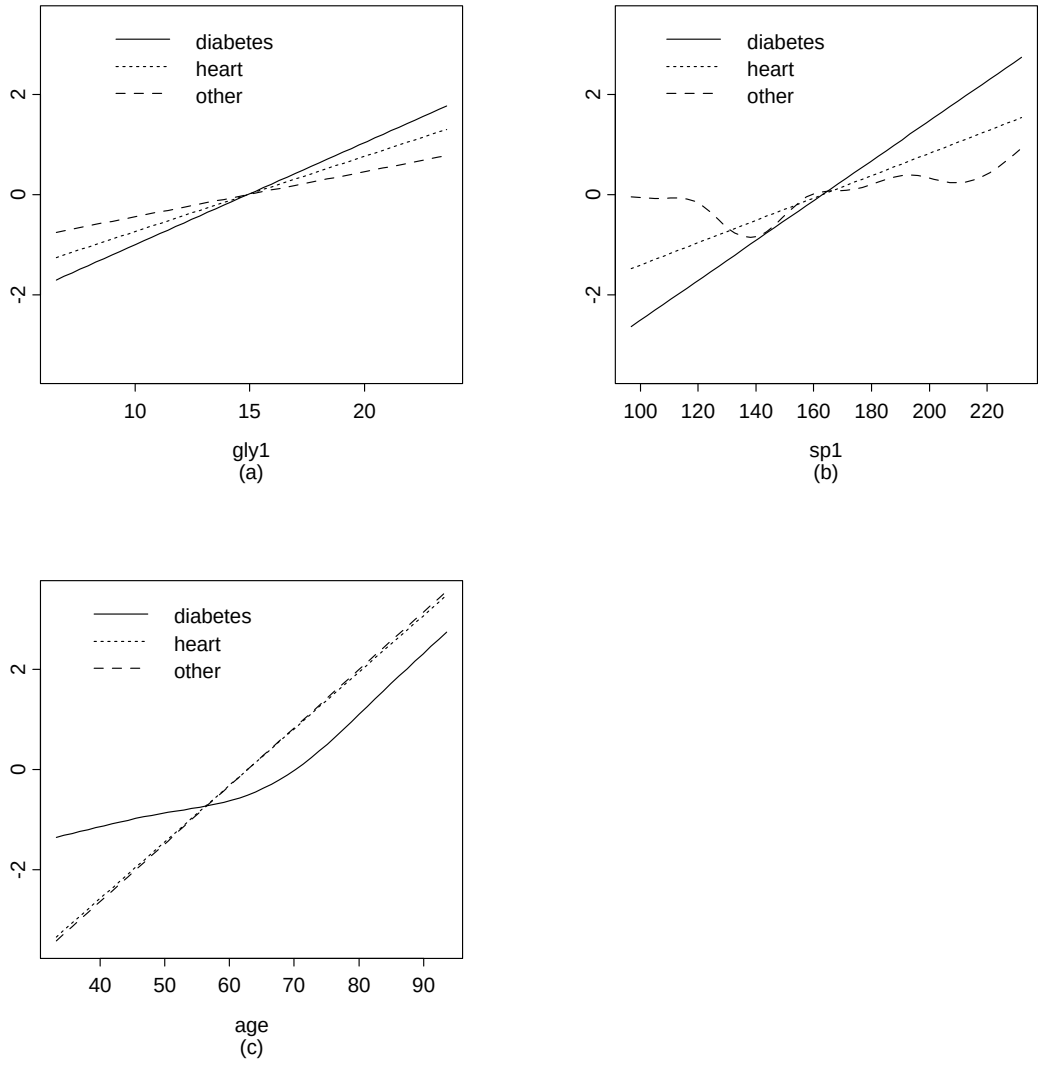


Figure 37: Main-effect plots in logit scale (y-axis corresponding to the value of logit function). Older onset without taking insulin group. Those who died after 12yrs from baseline are considered to be alive. $n = 646$.

Chapter 6

Concluding Remarks

6.1 Summary

We have proposed nonparametric models using smoothing spline ANOVA for modeling data with polychotomous response. We obtained the estimates by solving a minimization problem involving the penalized likelihood. A block one step SOR-Newton-Raphson method is used to solve this minimization problem. We use GCV and the unbiased risk method to choose smoothing parameters at each update. Our simulations indicate that the method will give us a good estimate most of the time for moderate data sets. We successfully applied this method to a medical data set. The disadvantage is that we can not apply this method to large data sets. Also, the convergence of this method is not guaranteed due to the way we choose the smoothing parameters although we did not experience any fail of convergence in our simulations and example.

We also proposed a fast algorithm to model the data with binary response (special case of polychotomous response). The randomized GACV we derived is shown to be a good proxy of the true CKL from the simulations. An approximate scheme is also proposed to speed up the computation in case of large data set. Simulations show that this method outperforms the iterated UBR method proposed by Gu.

To overcome the computational difficulties for large data sets with polychotomous responses, we proposed two methods. By transforming the polychotomous response data into several binary data sets, we can use the fast algorithm for binary data and obtain the final estimate by combining the estimate from each binary data sets. The disadvantage of this methods is that we don't have functional ANOVA decomposition for the final estimates. Simulations show that the performance of this method is close to the penalized polychotomous regression and the results are expected to get closer when the sample size gets larger. Alternatively, by following the derivation of randomized GACV for binary data we derived a randomized GACV for the penalized polychotomous regression problem. Combining with the approximate smoothing spline, this approach is expected to produce the solution much faster.

6.2 Future Research

Hypothesis testing and model selection are important for data analysis. The approximate posterior variance or covariance can be used to construct the confidence interval for the smoothing spline estimates. The performance and the interpretation of these confidence interval remain to be investigated.

Theoretical results can provide insight into and justification for the proposed methods. Large sample properties like asymptotic consistency, convergence rate, strong or weak consistency for the randomized GACV are desirable.

Bibliography

[1] N. Aronszajn. Theory of reproducing kernels. *Trans. Am. Math. Soc.*, 68:337–404, 1950.

[2] D. Barry. Nonparametric Bayesian regression. *Ann. Statist.*, 14:934–953, 1986.

[3] Colin B. Begg and Robert Gray. Calculation of polychotomous logistic regression parameters using individualized regressions. *Biometrika*, 71:11–18, 1984.

[4] S. Bose. Classification using splines. Technical Report 541, Department of Statistics, The Ohio State University, 1994.

[5] S. Le Cessie and J. C. Van Houwelingen. Logistic regression for correlated binary data. *Appl. Statist.*, 43:95–108, 1994.

[6] Z. Chen. Fitting multivariate regression functions by interaction spline models. *J. Roy. Stat. Soc. B*, 55:473–491, 1993.

[7] D. Cox. Approximation of method of regularization estimators. *Ann. Math. Statist.*, 16:694–713, 1988.

[8] D. Cox and Y. Chang. Iterated state space algorithms and cross validation for generalized smoothing splines. Technical Report 49, University of Illinois, Dept. of Statistics, Champaign, IL, 1990.

[9] D. Cox and F. O'Sullivan. Asymptotic analysis of penalized likelihood and related estimators. *Ann. Statist.*, 18:1676–1695, 1990.

[10] P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math.*, 31:377–403, 1979.

[11] J.H. Friedman. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, 1996.

[12] D. Girard. A fast 'Monte Carlo cross validation' procedure for large least squares problems with noisy data. Technical Report RR 687-M, IMAG, Grenoble, France, 1987.

[13] D. Girard. Asymptotic optimality of the fast randomized versions of GCV and C_L in ridge regression and regularization. *Ann. Statist.*, 19:1950–1963, 1991.

[14] Robert J. Glynn and Bernard Rosner. Accounting for the correlation between fellow eyes in regression analysis. *Arch Ophthalmol.*, 110:381–387, 1992.

[15] P. Green and B. Yandell. Semi-parametric generalized linear models. In R. Gilchrist, editor, *Lecture Notes in Statistics, Vol. 32*, pages 44–55. Springer, 1985.

[16] C. Gu. RKPACK and its applications: fitting smoothing spline models. In *Proceedings of the Statistical Computing Section*, pages 42–51. American Statistical Association, 1989. Code available thru `netlib`.

[17] C. Gu. Adaptive spline smoothing in non-Gaussian regression models. *J. Amer. Statist. Assoc.*, 85:801–807, 1990.

[18] C. Gu. Cross-validating non-Gaussian data. *J. Comput. Graph. Stats.*, 1:169–179, 1992.

[19] C. Gu. Penalized likelihood regression: a Bayesian analysis. *Statistica Sinica*, 2:255–264, 1992.

[20] C. Gu, D.M. Bates, Z. Chen, and G. Wahba. The computation of generalized cross-validation functions through householder tridiagonalization with applications to the fitting of interaction spline models. *SIAM Journal on Matrix analysis and Application*, 10:457–480, 1989.

[21] C. Gu and C. Qiu. Smoothing spline density estimation:theory. *Ann. Statist.*, 21:27–234, 1993.

[22] C. Gu and G. Wahba. Minimizing GCV/GML scores with multiple smoothing parameters via the Newton method. *SIAM J. Sci. Statist. Comput.*, 12:383–398, 1991.

[23] C. Gu and G. Wahba. Semiparametric analysis of variance with tensor product thin plate splines. *J. Royal Statistical Soc. Ser. B*, 55:353–368, 1993.

[24] C. Gu and G. Wahba. Smoothing spline ANOVA with component-wise Bayesian "confidence intervals". *J. Computational and Graphical Statistics*, 2:97–117, 1993.

[25] T. Hastie and R. Tibshirani. *Generalized additive models.* Chapman and Hall, 1990.

[26] T. Hastie and R. Tibshirani. Classification by pairwise coupling. Technical report, Department of Statistics, Stanford University and University of Toronto, 1996.

[27] D. Hosmer and S. Lemeshow. *Applied Logistic Regression.* New York: Wiley, 1989.

[28] S.T. Jensen, S. Johansen, and S.L. Lauritzen. Globally convergent algorithms for maximizing a likelihood function. *Biometrika*, 78:867–877, 1991.

[29] R. Klein, B. E. K. Klein, S. E. Moss, M. D. Davis, and D. L. DeMets. Glycosylated hemoglobin predicts the incidence and progression of diabetic retinopathy. *Journal of the American Medical Association*, 260:2864–2871, 1988.

[30] James Koehler and Art Owen. Computer experiments. In S. Ghosh and C.R. Rao, editors, *Handbook of Statistics, 13: Design and Analysis of Experiments*, pages 261–308. North-Holland, 1996.

[31] C. Kooperberg, S. Bose, and C.J. Stone. Polychotomous regression. *JASA*, 92:117–127, 1997.

[32] J. T. Kent K.V. Mardia and J.M. Bibby. *Multivariate Analysis.* London: Academic Press, 1979.

[33] R.A. Olshen L. Breiman, J.H. Friedman and C. J. Stone. *Classification and Regression Trees.* Wadsworth, 1984.

[34] J. Leeuv. Block-relaxation algorithm in statistics. Technical report, Department of Statistics, UCLA, 1995.

[35] E.L. Lehmann. *Theory of Point Estimation.* New York: Wiley, 1983.

[36] K. C. Li. Asymptotic optimality of C_L and generalized cross validation in ridge regression with application to spline smoothing. *Ann. Statist.*, 14:1101–1112, 1986.

[37] Kung-Yee Liang and Scott L. Zeger. A class of logistic regression models for multivariate binary time series. *JASA*, 84:447–451, 1989.

[38] P. McCullagh and J. Nelder. *Generalized Linear Models.* Chapman and Hall, London, 1983.

[39] J.M. Ortega and W.C. Rheinboldt. *Iteration Solution of Nonlinear Equations in Several Variables.* Academic Press, 1970.

[40] F. O'Sullivan. *The analysis of some penalized likelihood estimation schemes.* PhD thesis, Dept. of Statistics, University of Wisconsin, Madison, WI, 1983. Technical Report 726.

[41] B. Ripley. Neural networks and related methods for classification. *J. Roy. Statist. Soc.*, 56:409–456, 1994.

[42] Bernard Rosner. Multivairate methods in ophthalmology with application to other paired-data situations. *Biometrics*, 40:1025–1035, 1984.

[43] R. Klein S. Moss and B. Klein. Cause-specific mortality in a population-based study of diabetes. *American Journal of Public Health*, 81:1158–1162, 1991.

[44] J. Sacks, S.B. Schiller, and W.J. Welch. Designs for computer experimens. *Technometrics*, 31:41–47, 1989.

[45] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Designs and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[46] S. Schechter. Relaxation methods for convex problems. *SIAM J. Numer. Anal.*, 5:601–612, 1968.

[47] B.W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *J. Roy. Stat. Soc. Ser. B*, 47:1–52, 1985.

[48] R. Tibshirani and M. LeBlanc. A strategy for binary description and classification. *J. Comp. Graph Statist.*, 1:3–20, 1992.

[49] R. S. Varga. *Matrix Iterative Analysis.* Prentice-Hall, 1984.

[50] M.A. Villalobos and G. Wahba. Multivariate thin plate spline estimators for the posterior probabilities in the classification problem. *Commun. Statist.-Theor. Meth.*, 12:1449–1479, 1983.

[51] G. Wahba. Improper priors, spline smoothing and the problem of guarding against model errors in regression. *J. Roy. Stat. Soc. Ser. B*, 40:364–372, 1978.

[52] G. Wahba. Spline bases, regularization, and generalized cross validation for solving approximation problems with large quantities of data. In W. Cheney, editor, *Approximation Theory III*, pages 905–912. Academic Press, 1980.

[53] G. Wahba. Bayesian "confidence intervals" for the cross-validated smoothing spline. *J. Roy. Stat. Soc. Ser. B*, 45:133–150, 1983.

[54] G. Wahba. *Spline Models for Observational Data.* SIAM, 1990. CBMS-NSF Regional Conference Series in Applied Mathematics, v. 59.

[55] G. Wahba. Multivariate function and operator estimation, based on smoothing splines and reproducing kernels. In M. Casdagli and S. Eubank, editors, *Nonlinear Modeling and Forecasting, SFI Studies in the Sciences of Complexity, Proc. Vol XII*, pages 95–112. Addison-Wesley, 1992.

[56] G. Wahba, D.R. Johnson, F. Gao, and J. Gong. Adaptive tuning of numerical weather prediction models: Randomized gcv in three- and four-dimensional data assimilation. *Monthly Weather Review*, 11:3358–3369, 1995.

[57] G. Wahba, Y. Wang, C. Gu, R. Klein, and B. Klein. Smoothing spline anova for exponential families, with application to the wisconsin epidemiological study of diabetic retinopathy. *The Annals of Statistics*, 23:1865–1895, 1995.

[58] Y. Wang. *Smoothing spline analysis of Variance of Data from Exponential Families.* PhD thesis, TR 928, University of Wisconsin-Madison, Madison, WI, 1994.

[59] Y. Wang. Grkpack: Fitting smoothing spline analysis of variance models to data from exponential families. *Commun. Statist. Simulation and Computation*, 26:765–782, 1997.

[60] W. Wong. Estimation of the loss of an estimate. Technical Report 356, Dept. of Statistics, University of Chicago, Chicago, Il, 1992.

[61] D. Xiang. Model fitting and testing for non-gaussian data with a large data set. Technical Report 957, Ph D thesis, Department of Statistics, University of Wisconsin-Madison, 1996.

[62] D. Xiang and G. Wahba. A generalized approximate cross validation for smoothing splines with non-gaussian data. *Statistics Sinica*, 6:675–692, 1996.

[63] B. Yandell. Algorithms for nonlinear generalized cross-validation. In T.J. Boardman, editor, *Computer Science and Statistics: 18th Symposium on the Interface.* American Statistical Association, Washington, DC, 1986.