2019-01-01

Bayesian Analysis of Ordinal Outcomes Through Latent Variable Approach

Benard Owusu Dechi
University of Texas at El Paso, benardowusudechi@gmail.com

BAYESIAN ANALYSIS OF ORDINAL OUTCOMES WITH LATENT VARIABLE APPROACH

BENARD OWUSU DECHI

Master's Program in Mathematical Sciences

APPROVED:

Naijun Sha, Ph.D., Chair

Rosen Ori, Ph.D.

Thompson Sarkodie-Gyan, Ph.D.

Charles Ambler, Ph.D.
Dean of the Graduate School

to my

mother and father

with love and gratitude

BAYESIAN ANALYSIS OF ORDINAL OUTCOMES WITH LATENT VARIABLE APPROACH

by

BENARD OWUSU DECHI

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Master's Program in Mathematical Sciences

THE UNIVERSITY OF TEXAS AT EL PASO

May 2019

Acknowledgments

My heartfelt gratitude belongs to the Almighty God for His great provisions throughout my degree. I want to express my sincere appreciation to my supervisor, Dr. Naijun Sha of the Department of Mathematical Sciences at the University of Texas at El Paso, for his academic advice, supervision, encouragement and constant support. He is always readily available to provide clear explanations when I dont seem to make headway.

I also wish to extend my gratitude to my committee members, Dr. Ori Rosen of the Department of Mathematical Sciences and Professor Thompson Sarkodie-Gyan, of the Electrical and Computer Engineering. Their comments and guidance were valuable to the completion of this work.

Abstract

Modeling and predicting of ordinal outcomes have become essential study to many statisticians due to the numerous forms of data encountered in real life, which has such format. Many authors have proposed variant methods in modeling these type of data either in classical approaches (McCullagh, 1980a) or from the Bayesian perspective (Albert and Chib, 1993) (Cowles et al., 1996).

A commonly adopted way of modeling ordinal data is via an underlying continuous latent variable. That is to say that the observed ordinal outcomes have a correspondence with the latent variable through some set of cutoff points. Thus, it can be established that the probability of an ordinal outcome is equivalent to a continuous latent variable falling into an interval on the real line. Sometimes, there exist some difficulties in the estimation of these cutoff categories. (Albert and Chib, 1993) proposed an ordinal probit model in a Bayesian framework, in-cooperating a vague prior on the cutoff point parameters.

However, in this work, we try to establish a correspondence between the cutoff categories through the Dirichlet distribution via a reasonable transformation. The Gibbs sampling approach is used to estimate these parameters from their posterior distribution. We then compare our result after prediction to the well known Polytomous Ordinal Logistic Regression. It tends out that, our method yields more parsimonious results as compared to the POLR model.

Key Words: Latent Variable, Gibbs Sampling, Ordinal Outcomes, POLR Model, and Dirichlet Distribution.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, modeling and predicting of ordinal outcomes have become an essential study for many mathematicians and statisticians. Various forms of data encountered in real life have some natural ordering. For instance, in social and economic sciences, we usually come across ordinal outcomes. Sometimes, the magnitude of the order is readily not available. For example, the level of education consists of high school, bachelors degree, masters degree, and doctoral degree. This variable can be viewed as an ordinal variable but with no scale or magnitude between each order or category (Sirisrisakulchai and Sriboonchitta, 2016).

There are numerous and well-known methods which are used in the analysis of ordinal outcomes. One of which is the Ordinal Logistic Regression (OLR). In the case where the outcome is binary, the binary logistic regression is used. However, since this paper deals with multinomial outcomes, we would concentrate on the Polytomous Ordinal Logistic Regression (POLR) and compare it to our method, i.e., using Bayesian approach in predicting ordinal outcomes through some continuous latent variable.

1.1 The POLR Model

Ordinal outcomes are usually modeled by logistic regression. They can also be modeled by nominal regression model (Johnson and Wichern, 1992). However, to yield a more parsimonious and easily interpretable model is to use the Polytomous Ordinal Logistic Regression (POLR) which accounts for the natural ordering of the categories. (Walker and Duncan, 1967) initially proposed the cumulative logit model for modeling ordinal outcomes but later (McCullagh, 1980b) called it the proportional odds model. Given a multinomial

response variable Z with categorical outcomes denoted $j = 1, 2, , ..., J-1$, if $\boldsymbol{x}$ is a p-dimensional vector of variates, $\boldsymbol{\beta} = (\beta_1, \beta_2, ...\beta_{p-1})'$ then the dependence of Z on $\boldsymbol{x}$ is given as:

$$P(Z_i \leq j \mid \boldsymbol{x}_i) = \frac{\exp(\gamma_j + \boldsymbol{x}_i'\boldsymbol{\beta})}{1 + \exp(\gamma_j + \boldsymbol{x}_i'\boldsymbol{\beta})},\ j = 1, 2, 3, ..., J-1 \tag{1.1}$$

This can be rewritten in logit form as:

$$\text{logit}(\pi_j) = \log\Big[\frac{\pi_j}{1-\pi_j}\Big] = \gamma_j + \boldsymbol{x}_i'\boldsymbol{\beta}$$

$$\log\left[\frac{P(Z_i \leq j \mid \boldsymbol{x})}{P(Z_i > j \mid \boldsymbol{x})}\right] = \gamma_j + \boldsymbol{x}'\boldsymbol{\beta},\ j = 1, 2, 3, ..., J-1. \tag{1.2}$$

where $\pi_j = P(Z_i \leq j)$ is the cumulative probability for the event $(Z_i \leq j)$ and γ_j are unknown intercept of parameters satisfying the condition $\gamma_1 \leq \gamma_2 \leq, ..., \leq \gamma_J$. The Proportional Odds Model (POM) for ordinal logistic regression models the cumulative probabilities $P(Z_i \leq j)$ rather than the specific category probabilities $P(Z_i = j)$ as in the nominal logistic regression.

1.2 Outline of Thesis

The remaining parts of the thesis is organized in this manner. Chapter two provides a literature review on some Bayesian analysis of ordinal outcomes and other traditional known methods such as the Polytomous Ordinal Logistic Regression. Chapter three talks about the methodology, specifically the Gibbs sampling and other derivations. Chapter four outlines the simulation studies and the results from the estimates of parameters from the Gibbs sampling and the various prediction techniques.

Chapter 2

Literature Review

2.1 Introduction

This chapter presents a literature review on some Bayesian methods used in the analysis of ordinal outcomes. The section also outlines some weaknesses and strengths of some of these methods. Ordinal data is usually classified into several groups where there exists a natural ordering among the groups. There has been much increase in modeling these type of data, especially where there exists some correlation in the ordered categories. Usually, the data is either treated as continuous or reduced to binary outcomes, with the purpose to lower the complexity in its analysis and presentation. In the quest to implement such analysis to the data, leads to the underestimation of the variance and loss of information.

Many authors have proposed variant methods in modeling these type of data either in classical approaches (McCullagh, 1980a) or from the Bayesian perspective (Albert and Chib, 1993) (Cowles et al., 1996). A commonly adopted way of modeling ordinal data is via an underlying continuous latent variable. The model thus relies on the assumption that the latent variable has a correspondence to the ordinal variable based on some interval. That is to say that the observed ordinal outcomes have a relationship with the latent variable through some set of cutoff points. Thus , it can be established that the probability of an ordinal outcome is equivalent to a continuous latent variable falling into an interval on the real line.

(Agresti, 1996) proposed that the latent variable linear models imply cumulative link models. He realized that, with many ordinal variables, it is realistic to regard the observed response as a crude measurement of some continuous latent variable. His method is very

similar to that of (Albert and Chib, 1993). (Agresti, 1996) also made inferences for multinomial models using the Bayesian approach. He argued that it is common to use diffuse normal priors on the effect parameters. Moreover, for any link function, Bayesian model fitting uses MCMC with the product of the chosen prior densities and the multinomial likelihood function for the model. For cumulative link models, the prior distributions for the intercept parameters should take into account the ordering constraint by rightly truncating the priors that would be used without the constraints. In his illustration of the Bayesian approach, he modeled the mental impairment data file, where he used a relatively flat normal prior for the parameters, with mean 0 and standard deviation 10. The posterior mean estimates are based on a long run of the MCMC process.

(Sirisrisakulchai and Sriboonchitta, 2016) also used a Bayesian analysis approach to develop a parametric model to investigate the effect of a binary treatment variable on an ordinal outcome of interest through a latent variable. (Kwon et al., 2007) also used the probit model in identifying biomarkers from mass spectrometry data with ordinal outcomes, a method similar to that adopted in this work with the difference existing in finding the cut off boundaries estimation. One of the issues often associated with ordinal data modeling is the slow convergence rate of the samples generated from the MCMC process. (Albert and Chib, 1993) proposed an ordinal probit model in a Bayesian framework, in-cooperating a vague prior on the cutoff point parameters. The problem encountered in his approach is the slow convergence of the Gibbs sampling implementation for a large sample size.

In avoiding such issues in the estimation of cutoff point parameters jointly with other parameters, (Zhou, 2006) in her work proposed a mixture model which can model the ordinal property of the data without the need to estimate these parameters. However, in our setting, we try to establish a correspondence between the cutoff categories through the Dirichlet distribution via a reasonable transformation. The Gibbs sampling approach is used to estimate these parameters from their posterior distribution.

2.2 Gibbs Sampling

Gibbs sampling is one MCMC algorithm that repeatedly samples from the conditional distribution of one variable of the target distribution given all of the other variables. It is applicable when the joint distribution is not explicitly known or is complicated to sample from directly, but quite easy to sample from the conditional distributions of each variable. Samples are obtained by running through all the posterior conditionals, with one random variable at a time. Since random values initiate the algorithm, the samples simulated at early iterations may not necessarily be a representation of the actual posterior distribution.

However, the theory of MCMC guarantees that the stationary distribution of the samples generated using this approach is the target joint posterior that we are interested in (Gilks et al., 1995). Thus, the MCMC algorithm is run for a large number of iterations with the belief that convergence to the desired posterior distribution would be obtained. Since samples from the early iterations are not from the target posterior, it is essential to discard them. The immediately discarded samples from the iterations are often referred to as the "burn-in" period.

The logic behind the implementation of the MCMC sampling is that we can estimate any desired expectation by ergodic averages. That is, we can calculate any statistic of the posterior distribution as far as we have N sufficiently simulated samples from that distribution, i.e:

$$E\big[h(\theta)\big]_{\mathcal{P}} \approx \frac{1}{N}\sum_{i=1}^{N} h(\theta^{(i)}) \tag{2.1}$$

where $\mathcal{P}$ is the posterior distribution of interest, and $h(\theta^{(i)})$ is the i^{th} simulated sample from $\mathcal{P}$. For instance, we can estimate the mean by

$$E\big[\theta\big]_{\mathcal{P}} = \frac{1}{N}\sum_{i=1}^{N} \theta^{(i)} \tag{2.2}$$

Chapter 3

Methodology

This chapter outlines the details of the methodology used in analysis of the work. It outlines the formulation of the ordered probit model, how the prior settings is done, posterior distribution, and how the correspondence between the cutoff points and the Dirichlet distribution is obtained. It also outlines the Gibbs sampling implementation and the various prediction procedures used in this work.

3.1 Ordered Probit Model

Let $(\boldsymbol{Z}_{n\times 1}, \boldsymbol{X}_{n\times p})$ denote the observed data, where $\boldsymbol{Z}_{n\times 1}$ is the vector of ordered categorical outcomes and $\boldsymbol{X}_{n\times p}$ is the matrix of covariates. Each outcome Z_i is associated with a vector $(p_{i,0}, ..., p_{i,J-1})$, where $p_{i,j} = P(Z_i = j)$ is the probability that subject i falls in the ordered j class. The probabilities $p_{i,j}$ can be related to the linear predictor $\boldsymbol{x}_i'\boldsymbol{\beta}$. We assume that there exists a latent continuous random variable Y_i, such that

$$Y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \epsilon_i,\ \epsilon_i \sim N(0,1) \tag{3.1}$$

The correspondence between the $Z_{i's}$ and the latent variable Y_i can be defined as

$$Z_i = j \text{ if } \delta_{j-1} < Y_i < \delta_j,\ j = 1, 2, ..., J, \text{ and } i = 1, 2, ..., n \tag{3.2}$$

where the boundaries are unknown and $-\infty = \delta_0 < \delta_1 < ... < \delta_J = \infty$.

3.2 Prior Settings

We assume that our prior $\boldsymbol{\beta}$ follows a Multivariate Normal distribution with hyperparameters $\boldsymbol{\beta}_0$ and $\boldsymbol{\Sigma}_0$ i.e. $\boldsymbol{\beta} \sim MVN(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$. For our prior distribution $\pi(\boldsymbol{\delta})$, we establish a correspondence between the $\boldsymbol{\delta}$ and the Dirichlet distribution via a reasonable transformation as shown in 3.5. To initialize our delta values for the Gibbs sampling, we randomly generate values from the uniform distribution according to the number of categories and sort it in ascending order to ensure the ordering in the cut off points of $\boldsymbol{\delta}$. The estimates are then computed from the Gibbs sampling after a reasonable number of iterations.

3.2.1 The Dirichlet and the Delta Correspondence

The Dirichlet distribution which we denote $\text{Dir}(\alpha_1, ..., \alpha_J)$, is parameterized by positive scalars $\alpha_j > 0$ for $j = 1, ..., J$, where $J \geq 2$. The support of the Dirichlet distribution is the $(J-1)$ dimensional simplex S_J; that is, all J dimensional vectors form a valid probability distribution. The probability density of $p = (p_1, ..., p_J)$ with $\sum p_j = 1$ is given by

$$\pi(p_1, ..., p_J; \alpha_1, ..., \alpha_J) = \frac{\Gamma(\sum_{j=1}^{J} \alpha_j)}{\prod_{j=1}^{J} \Gamma(\alpha_j)} \prod_{j=1}^{J} p_j^{\alpha_{j-1}} \tag{3.3}$$

The Dirichlet distribution is the multivariate generalization of the beta distribution. It is often used as the prior distribution in Bayesian inference and it is the conjugate prior of the categorical distribution and multinomial distribution (Tu, 2014). The relationship between the Dirichlet and the unknown cut off categories $\boldsymbol{\delta}$ is obtained as:

$$\begin{aligned} P(\delta < \delta_1) &= F(\delta_1) = p_1 \\ P(\delta_1 < \delta < \delta_2) &= F(\delta_2) - F(\delta_1) = p_2 \\ \vdots \qquad & \qquad \vdots \\ P(\delta_{J-1} < \delta < \delta_J) &= F(\delta_J) - F(\delta_{J-1}) = p_J \end{aligned}$$

We now find a distribution for the $\boldsymbol{\delta}$ using the transformation above i.e

$$\pi_{\delta}(\boldsymbol{\delta}) = \pi_{p}(\boldsymbol{p}) \times |\text{Jacobian}| \tag{3.4}$$

$$\pi(\boldsymbol{\delta}) = f_{P_1,\cdots,p_{J-1}}[F(\delta_1), F(\delta_2) - F(\delta_1), \cdots, F(\delta_{J-1}) - F(\delta_{J-2})] \times |\text{Jacobian}|$$

$$\text{Jacobian} = \begin{pmatrix} \frac{\partial p_1}{\partial \delta_1} & \frac{\partial p_1}{\partial \delta_2} & \frac{\partial p_1}{\partial \delta_3} & \cdots & \frac{\partial p_1}{\partial \delta_{J-1}} \\ \frac{\partial p_2}{\partial \delta_1} & \frac{\partial p_2}{\partial \delta_2} & \frac{\partial p_2}{\partial \delta_3} & \cdots & \frac{\partial p_2}{\partial \delta_{J-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial p_{J-1}}{\partial \delta_1} & \frac{\partial p_{J-1}}{\partial \delta_2} & \frac{\partial p_{J-1}}{\partial \delta_3} & \cdots & \frac{\partial p_{J-1}}{\partial \delta_{J-1}} \end{pmatrix}$$

$$\text{Jacobian} = \begin{pmatrix} f(\delta_1) & 0 & 0 & 0 & \cdots & 0 \\ -f(\delta_1) & f(\delta_2) & 0 & 0 & \cdots & 0 \\ 0 & -f(\delta_2) & f(\delta_3) & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & -f(\delta_{J-2}) & f(\delta_{J-1}) \end{pmatrix}$$

since the Jacobian matrix is a lower triangular matrix, its determinant is the product of the diagonals i.e.

$$|\text{Jacobian}| = \prod_{j=1}^{J-1} f(\delta_j)$$

By substituting into the Dirichlet distribution with ignoring the normalizing constant, we obtain the joint pdf of $\boldsymbol{\delta}$ as;

$$\begin{aligned} \pi(\boldsymbol{\delta}) = {} & F(\delta_1)^{\alpha_1 - 1}[F(\delta_2) - F(\delta_1)]^{\alpha_2 - 1} \times [F(\delta_3) - F(\delta_2)]^{\alpha_3 - 1} \\ & \times \cdots \times [F(\delta_{J-1}) - F(\delta_{J-2})]^{\alpha_{J-1} - 1} \times [1 - F(\delta_{J-1})]^{\alpha_J - 1} \prod_{j=1}^{J-1} f(\delta_j) \end{aligned} \tag{3.5}$$

3.2.2 Hyperparameter Settings

From the assumption that our prior $\boldsymbol{\beta}$ follows a Multivariate Normal distribution with hyperparameters $\boldsymbol{\beta}_0$ and $\boldsymbol{\Sigma}_0$ i.e. $\boldsymbol{\beta} \sim MVN(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$. We initialize $\boldsymbol{\beta}_0$ to be a $p \times 1$ zero

vector whereas $\boldsymbol{\Sigma}_0 = c\text{I}$. The value c is somewhat set large to increase the variability in the prior distribution of $\boldsymbol{\beta}$. The choice of $F(.)$ for $\boldsymbol{\delta}$ can be any cumulative density function whose domain lies in $(-\infty, \infty)$. We choose $F(.)$ from the normal distribution with $N(0, 10)$. For the shape and scale parameters, we fix the α's such that $\alpha = 1$, which makes Beta(1,1) eventually to be Unif(0,1).

3.3 Posterior Distribution

We perform some Bayesian inference, by updating our prior beliefs with information from the data to obtain the following posterior distribution

$$\pi(\boldsymbol{\beta}, \boldsymbol{\delta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) \propto L(\boldsymbol{\beta}, \boldsymbol{\delta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})\pi(\boldsymbol{\beta})\pi(\boldsymbol{\delta}) \tag{3.6}$$

$$\text{where } L(\boldsymbol{\beta}, \boldsymbol{\delta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) = \prod_{i=1}^{n} f_{Y_i}(y_i) \times I(\delta_{j-1} < y_i < \delta_j) \tag{3.7}$$

$$\text{with } Y_i|\boldsymbol{x}_i \sim N(\boldsymbol{x}_i{}'\boldsymbol{\beta}, 1).$$

The conditional posterior for $\boldsymbol{\beta}$ is:

$$\pi(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) \propto L(\boldsymbol{\beta}, \boldsymbol{\delta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})\pi(\boldsymbol{\beta}) \tag{3.8}$$

$$\text{i.e } (\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{Y}), \boldsymbol{Z}) \sim N(\tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\Sigma}})$$

It can be shown that the posterior mean and the posterior variance of $\boldsymbol{\beta}$ are respectively

$$\tilde{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{\Sigma_0^{-1}})^{-1}(\boldsymbol{X}'\boldsymbol{Y} + \boldsymbol{\Sigma_0^{-1}}\boldsymbol{\beta}_0) \tag{3.9}$$

$$\tilde{\boldsymbol{\Sigma}} = (\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{\Sigma_0^{-1}})^{-1} \tag{3.10}$$

The conditional posterior for $\boldsymbol{\delta}$ is:

$$\pi(\boldsymbol{\delta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\beta}) \propto \pi(\boldsymbol{\delta}) \times \prod_{i=1}^{n} \prod_{j=1}^{J} I(\delta_{j-1} < y_i < \delta_j) \tag{3.11}$$

Since the distribution has no close form, as a result we compute the conditional distribution for each of the deltas (δ_j) so that we can easily sample from it. If we let $\boldsymbol{\delta}_{(-j)}$ be the vector $\boldsymbol{\delta}$ without the $j'th$ element, that is
$\boldsymbol{\delta}_{(-j)} = (\delta_1, ..., \delta_{j-1}, \delta_{j+1}, ..., \delta_{J-1})$. Then the conditional posterior is:

$$\pi(\delta_j \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{\delta}_{(-j)}) \propto [F(\delta_j) - F(\delta_{j-1})]^{\alpha_j - 1} [F(\delta_{j+1}) - F(\delta_j)]^{\alpha_{j+1} - 1} f(\delta_j)$$
$$\times I(c_{j,1} < \delta_j < c_{j,2}),\ j = 1, 2, ..., J-1 \quad (3.12)$$

where $c_{j,1} = \max\{y_i, i = 1, 2, ..., n : z_i = j\}, c_{j,2} = \min\{y_i, i = 1, 2, ..., n : z_i = j + 1\}$, with$j = 1, 2, ..., J-1$ and notice that $F(\delta_0) = 0$ and $F(\delta_J) = 1$ i.e $F(-\infty) = 0$ and $F(\infty) = 1$.

It can be shown that conditionally, δ_j is a random variable whose transformed $F(\delta_j)$ is distributed as a scaled $[F(\delta_{j+1}) - F(\delta_{j-1})]$ and shifted $F(\delta_{j-1})$ Beta(α_j, α_{j+1}) truncated at the interval $[F(c_{j,1}), F(c_{j,2})]$, i.e

$$F(\delta_j) \sim [F(\delta_{j+1}) - F(\delta_{j-1})]\ \text{Beta}(\alpha_j, \alpha_{j+1}) + F(\delta_{j-1}) \quad (3.13)$$

3.4 Gibbs Sampling Implementation

The logic in Gibbs sampling is to generate posterior samples by sweeping through each variable to sample from its conditional distribution with the rest of the variables fixed to their current values.

Given the $(k-1)^{\text{th}}$ step, the Gibbs sampling for the next k^{th} iteration in our case is implemented as follows:

$$\begin{cases} (Y_i^{(k)} | \boldsymbol{\beta}^{(k-1)}, \boldsymbol{\delta}^{(k-1)}) \sim N(\boldsymbol{x}_i{}' \boldsymbol{\beta}^{(k-1)},\ 1) \text{ with } \delta_{j-1}^{(k-1)} < Y_i^{(k)} < \delta_j^{(k-1)} \text{ if } Z_i = j \\ (\boldsymbol{\beta}^{(k)} | \boldsymbol{Y}^{(k)}, \boldsymbol{\delta}^{(k-1)}) \sim N(\tilde{\boldsymbol{\beta}}^{(k)}, \tilde{\boldsymbol{\Sigma}}) \\ (F(\delta_j^{(k)}) | \boldsymbol{\delta}_{(-j)}^{(k)}, \boldsymbol{Y}^{(k)}, \boldsymbol{\beta}^{(k)}) \sim [F(\delta_{j+1}^{(k-1)}) - F(\delta_{j-1}^{(k)})]\ \text{Beta}(\alpha_j, \alpha_{j+1}) + F(\delta_{j-1}^{(k)}) \\ \qquad\qquad \text{with } F(\delta_j^{(k)}) \in\ [F(c_{j,1}), F(c_{j,2})] \end{cases} \quad (3.14)$$

Where $i = 1, 2, ..., N,\ k = 1, 2, ..., M,$ and $j = 1, 2, ..., J-1$. This process continues until convergence (i.e. the distribution of the sample values behave as if they were actually sampled from the true posterior joint distribution). With discarding the immediate samples, we may compute several parameter estimates from the posterior distribution such as the

posterior mean, variance, median and mode may be computed. For instance, we compute the estimates of the posterior means $\hat{\boldsymbol{Y}}$, $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\delta}}$ for the purpose of our analysis.

3.5 Data Ellipsoid Plots

We visualize the data sets to examine the structural differences existing between the variables across the various groups. The main idea as in (Dempster, 1969) is that for a p-dimensional sample, $\boldsymbol{X}_{n\times p}$, the $p \times p$ covariance matrix $\boldsymbol{S}$ can be represented by the p-dimensional concentration or data ellipsoid, D_c of size (radius) c. This is defined as the set of all points $\boldsymbol{X}$ satisfying

$$D_c(\bar{\boldsymbol{X}}, \boldsymbol{S}) = \{\boldsymbol{X} : (\boldsymbol{X} - \bar{\boldsymbol{X}})^T \boldsymbol{S}^{-1} (\boldsymbol{X} - \bar{\boldsymbol{X}}) \leq c^2\} \tag{3.15}$$

Clearly, we see from the quadratic form in 3.15 that it corresponds to the set of points whose squared Mahalanobis distances $D^2(\boldsymbol{X}) = (\boldsymbol{X} - \bar{\boldsymbol{X}})^T \boldsymbol{S}^{-1} (\boldsymbol{X} - \bar{\boldsymbol{X}})$, from the centroid of the sample, $\bar{\boldsymbol{X}} = (\bar{X}_1, \bar{X}_2, ..., \bar{X}_p)$, are less or equal to c^2. For a multivariate normal variable, the data ellipsoid approximates a contour with constant density in their joint distribution. (Friendly et al., 2013) elaborates more on the properties of data ellipsoids and their use to interpret a wide variety of problems and applications in multivariate linear models.

The Skull data set provides an example where there exist substantial differences among the means of groups, but little evidence for heterogeneity of their covariance matrices. Since the location and variability within groups may be an essential measure for prediction, we explore to visualize and understand the heterogeneity and the variations in means within groups of the data.

The covEllipses function in r is used to plot the data set to have a visual display of the data. This method is useful when looking at the data ellipses for all pairs of variables in a scatter plot matrix format (Friendly and Sigal, 2018). The data ellipsoid plot for each of the data set in shown in chapter four.

3.6 Model Prediction

We now outline different methods to make predictions based on our estimates from the Bayesian approach. In each case of our prediction, the data set is partitioned into train and test set, and with the test set, we make predictions for the new X to identify which category Z it falls through the latent variable Y. In section 3.7, a detailed explanation of the data partitioning for cross validation is given.

3.6.1 Point Estimate Through the Latent Variable

Various techniques are used in making prediction which yields almost the same result. Predictions are made by the point estimates of the latent variable using the relation $\hat{Y}_{new} = \boldsymbol{x}'_{\text{new}}\hat{\boldsymbol{\beta}}$. We determine $\hat{Z}_{\text{new}}$ using:

$$\hat{Z}_{\text{new}} = j \text{ if } \hat{\delta}_{j-1} < \hat{Y}_{\text{new}} < \hat{\delta}_j \tag{3.16}$$

The $\hat{Z}_{\text{new}}$ is compared to the true Z to calculate the misclassification error rate.

3.6.2 Prediction Via Probabilities

Another method of prediction based on probability is performed. In this scenario, we find the probability of each Y falling in the estimated intervals based on $(\hat{\boldsymbol{\delta}})$ and obtain the category which has the maximum probability, i.e.

$$P(\hat{Z}_{\text{new}} = j) \ = P(\hat{\delta}_{j-1} < \hat{Y}_{\text{new}} < \hat{\delta}_j) \tag{3.17}$$

$$\therefore \ \hat{Z}_{\text{new}} = \underset{j}{\text{argmax}} \ P(\hat{Z}_{\text{new}} = j), \ j = 1, 2, ..., J \tag{3.18}$$

Using this prediction approach, we compare the predicted $\hat{Z}$ to the true Z and then report the misclassification error rate.

3.6.3 Prediction by Weighted Average

Another form of prediction used very similar to the later i.e (prediction via probabilities) is by computing the weighted average of $\hat{Z}$, where the weights in this case are the indices of the individual categorical probabilities. This is computed as:

$$E\{\hat{Z}_{\text{new}}\} = \sum_{j=1}^{J} jp_j \tag{3.19}$$

$$\text{where } P(\hat{Z}_{\text{new}} = j) \ = P(\hat{\delta}_{j-1} < \hat{Y}_{\text{new}} < \hat{\delta}_j) = p_J$$

After which we choose an integer which is most close to $E\{\hat{Z}\}$ as our predicted category.

3.7 Data Partitioning for Cross-Validation

To ensure that at least all the samples from within each category has a chance of being used in training and testing the model; a systematic approach is used to partition the data to cross-validate the model. For any given data with some number of categories, we randomly and equally select subsamples from each category and combine it to form the testing set, and the rest is used as the training set. We choose the next subsamples again from each group which has not been previously selected and combine it to form the testing set with the remaining samples forming the training set.

We repeat this partitioning approach until all the samples from each category are exhausted. At each stage of partition, we record the error rate from the prediction. The average of the error rates is then computed. For instance, in our simulation, we generate ninety observations which have three categories with thirty samples in each category, five samples from each category is selected and concatenated to form the testing set and the rest forms the training set. The partitioning is repeated for the next five samples from each category and continues until all the samples from each category are exhausted. Similar partitioning approach is used for the real data. The summary of the results for each of the data set is shown in chapter four.

3.8 MCMC Diagnostics

We perform some diagnostics that would help determine how well the Gibbs sampler mixes and how efficient or reliable our estimates are. There are many ways of which these diagnostics may be done. We stick to the use of the trace plots. The trace plot is the simplest tool for visualizing the convergence of a Markov chain. It gives the plot of the values generated from the Markov chain versus the iteration number. It also helps to give a visual idea about the number of samples that must be discarded as burn-in periods in the chain. It is sometimes said that we are aiming for the trace to look like a hairy caterpillar which would mean that the parameters of the model explore well in the parameter space.

The autocorrelation is another way to check for convergence between the samples returned by our MCMC. The lag-k autocorrelation is the correlation between every sample and the sample k steps before. This autocorrelation should become smaller as k increases, i.e., the samples are considered to be independent. If, on the other hand, autocorrelation remains high for higher values of k, this indicates a high degree of correlation between our samples and slow mixing.

Chapter 4

Simulation and Real Data Application

4.1 Introduction

This chapter outlines the procedure adopted in the simulation of the data set. It shows the MCMC diagnostics of the Gibbs sampler implementation to see how well the model mixes in the parameter space. It outlines the comparison of the Bayesian method with the POLR method. The section as well talks about the misclassification error rates from the cross-validations of each data set.

4.2 Data Simulation

Our simulation is done from the multivariate normal distribution. We simulate two different data sets. Since we are dealing with ordinal data analysis, we use a different vector of means to ensure that the simulated data within each category are well separated. The variance-covariance matrices are set differently from each group using distinct correlation matrices. The vector of means for the three categories are shown below:

$$\boldsymbol{\mu}_1 = [3, 2, 4, 1]', \ \boldsymbol{\mu}_2 = [3, -2, 4, -1]', \ \boldsymbol{\mu}_3 = [-3, -2, -4, -1]'$$

The variance-covariance matrix used for the simulation is:

$$\boldsymbol{\Sigma} = \sigma^2[(1-\rho)\mathbf{I} + \rho \times \mathbf{1} \times \mathbf{1}']$$

where $\sigma = 2$, and ρ is the correlation with $\boldsymbol{\rho} = (0.1, 0.5, 0.9)$, $\mathbf{I}$ is the identity matrix and $\mathbf{1}$ is vector of ones. This is basically done to control the variability within each group. For the purpose of our analysis, we simulate data that consists of four independent variables and a target or dependent variable which has three categories. Each category has a sub-sample size of thirty making a total sample size of ninety.

4.2.1 Data Ellipsoid Plot for First Simulated Data

The data ellipsoid plot for the first simulated data is shown below to visualize the structural differences existing between the variables across the various groups.

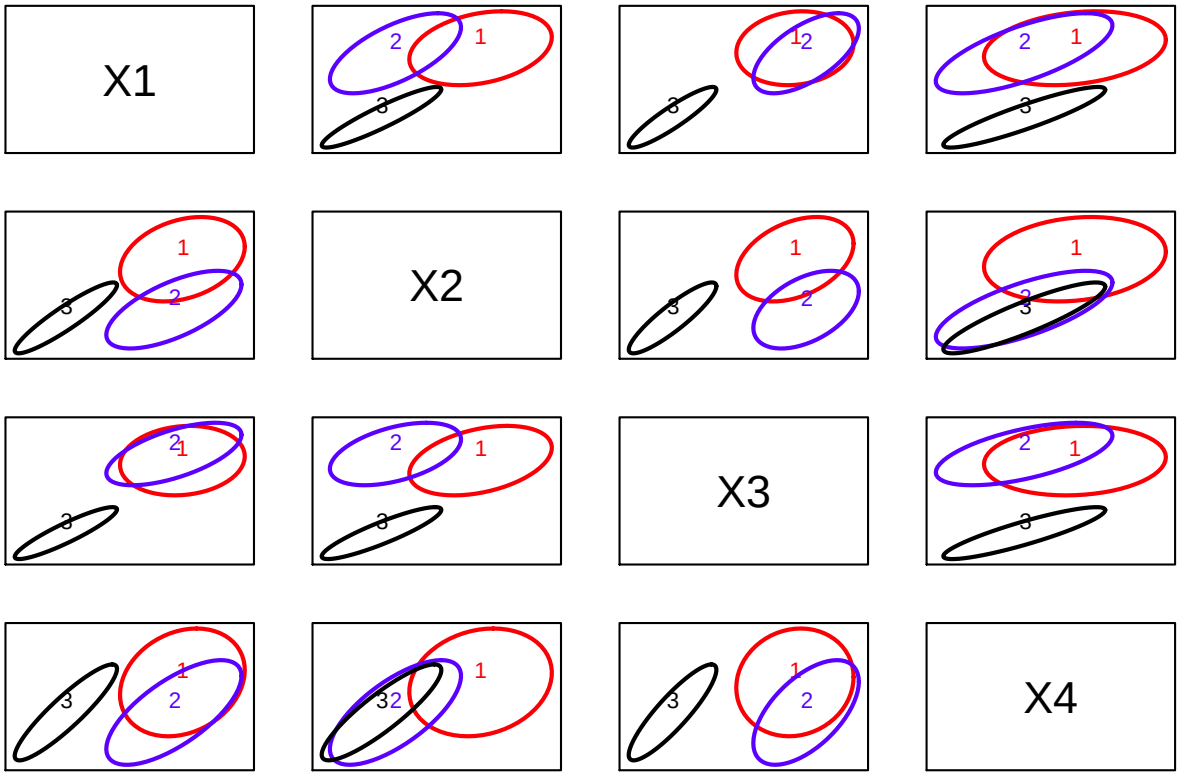

Figure 4.1: Data Ellipsoid Plot for First Simulated Data

From the above, we see that the variables are relatively separated from each other with

the first and second category overlapping each other in all pairs. Thus we should expect to see a quite low misclassification error rate after prediction with the misclassifications coming from the first and second categories.

4.2.2 Trace Plots of Parameters from First Simulated Data

The trace plots of the parameters from the posterior distribution of the simulated data are shown below. It could be seen that the $\beta's$ have a quiet good mixing rate as compared to the $\delta's$.

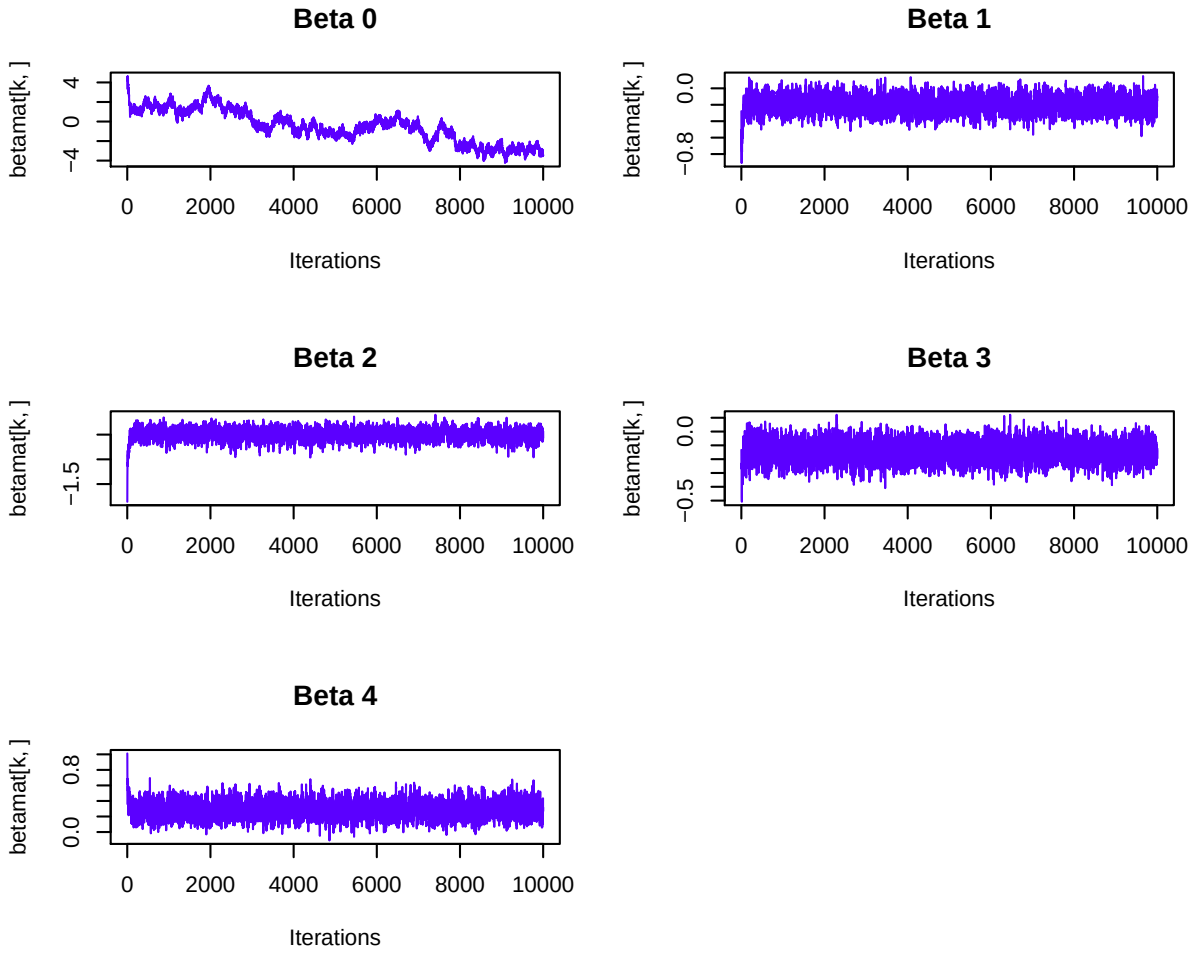

Figure 4.2: Trace Plot of $\beta's$ from First Simulated Data

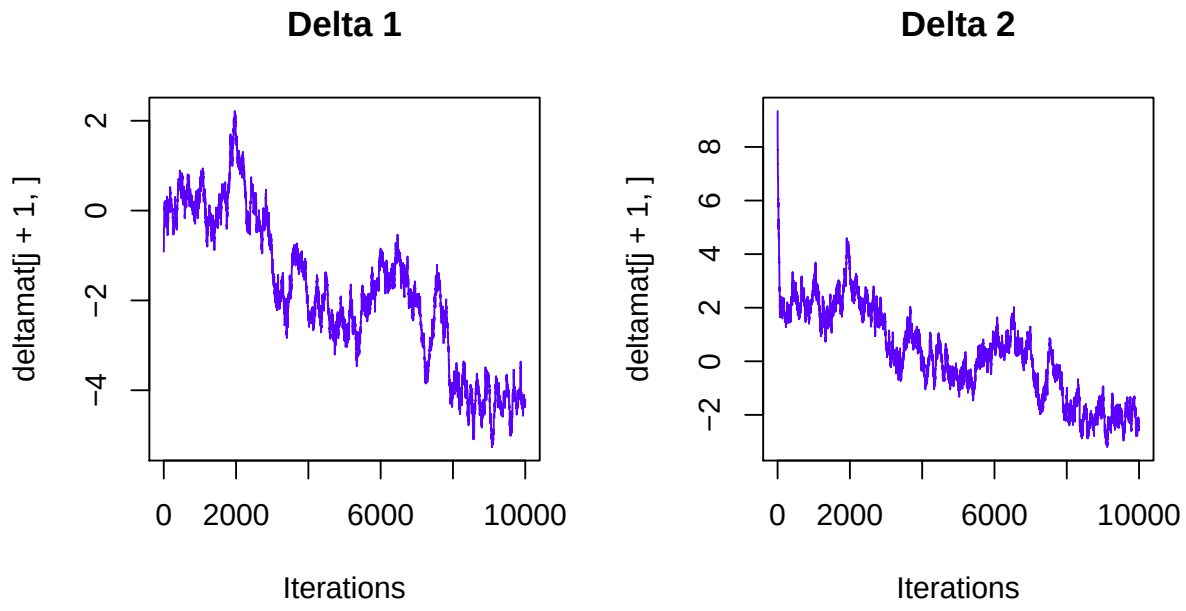

Figure 4.3: Trace Plot of $\delta's$ from First Simulated Data

4.2.3 Parameter Estimates for First Simulated Data

From our posterior distribution 3.6, we estimate the parameters of the conditional posteriors i.e. $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\delta}}$, via Gibbs sampling, starting from a set of arbitrary values of parameters and the estimates of their mean and standard deviations are obtained after a reasonable number of iterations. The parameter estimates of the simulated data from the last CV iterations of the Gibbs sampling are shown below.

Parameters	β_0	β_1	β_2	β_3	β_4	δ_1	δ_2
Estimate	-1.0	-0.20	-0.5	-0.14	0.29	-2.32	-0.25
SD	1.42	0.09	0.11	0.07	0.1	1.43	1.41

Table 4.1: Parameter Estimates of First Simulated Data

4.2.4 Cross Validated Error Rates for First Simulated Data

The cross-validated misclassification error rates for the simulated data is tabulated below. The reported error rates based on the type of the prediction method adopted, and this compared to the results from the POLR method.

Prediction Type	**Error Rate**
Point Estimates	0.3
By Probabilities	0.3
By Weighted Average	0.3
POLR Method	0.3

Table 4.2: Cross Validated Misclassifiaction Error Rates for First Simulated Data

The confusion matrix table for the first simulated data of both the Bayesian and the POLR method is shown below.

Actual	1	2	3
1	19	10	0
Predicted 2	9	18	4
3	2	2	26

Table 4.3: Bayesian Confusion Matrix for 1st Simulated Data

Actual	1	2	3
1	19	10	0
Predicted 2	9	18	4
3	2	2	26

Table 4.4: POLR Confusion Matrix for 1st Simulated Data

4.3 Application to Second Simulated Data

We perform another simulation where we to try make the separation among the variables within the groups as wide as possible as compared to the first simulation to see how consistent our method compares to the POLR model with respect to the error rates. The vector of means used this time around is shown below:

$$\boldsymbol{\mu}_1 = [5, 1, 4, 6]', \ \boldsymbol{\mu}_2 = [3, -2, -4, -1]', \ \boldsymbol{\mu}_3 = [-5, 7, 4, -10]'$$

The variance-covariance matrix is the same as used in the first simulation above. Since our concern is much of the location and how it affects the error rates for prediction.

4.3.1 Data Ellipsoid Plot for Second Simulated Data

The data ellipsoid plot for the second simulated data is shown below to visualize the structural differences existing between the variables across the various groups.

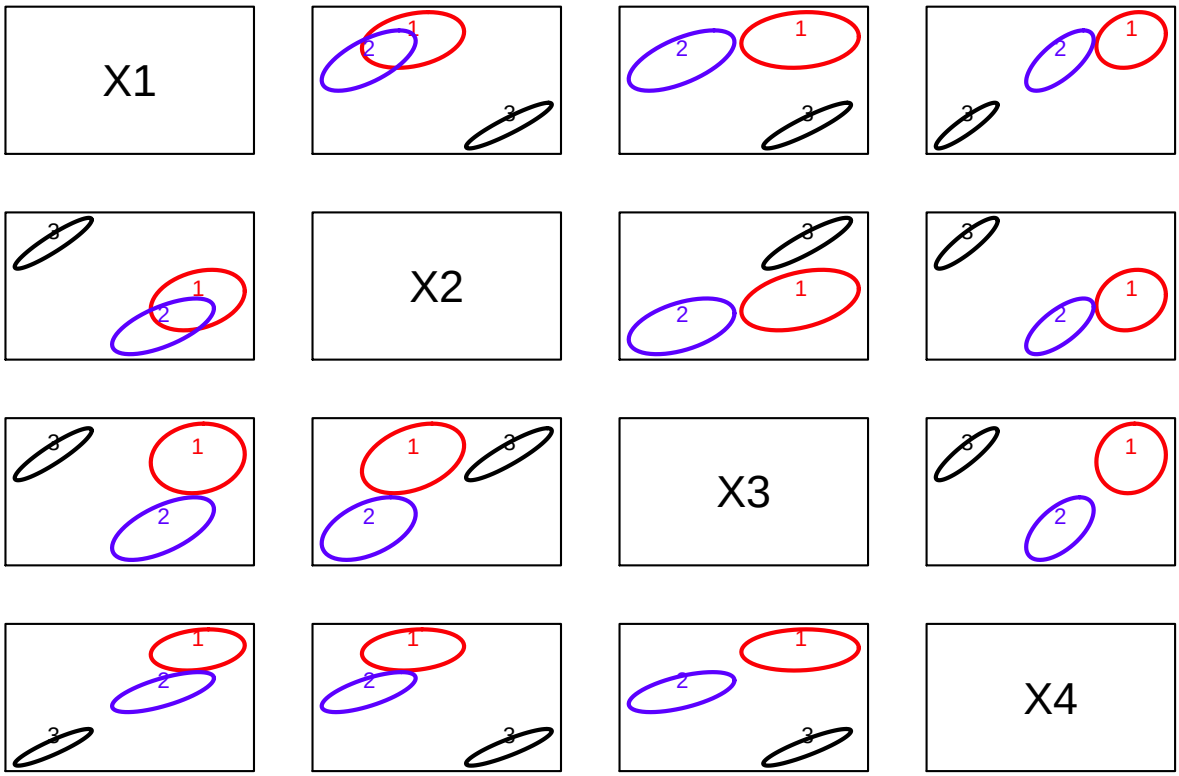

Figure 4.4: Data Ellipsoid Plot for Second Simulated Data

From the above ellipsoid plot, we see that the variables are well separated from each other as compared to the first simulated data, with a small portion of the first category overlapping the second category. The third category is much separated from the other two

groups. As such, we should expect less misclassification error rate as compared to the first simulated data.

4.3.2 Parameter Estimates for Second Simulated Data

The parameter estimates of the second simulated data from the last CV iterations of the Gibbs sampling are shown below.

Parameters	β_0	β_1	β_2	β_3	β_4	δ_1	δ_2
Estimate	-2.14	0.17	1.97	-1.35	-2.53	-8.29	15.26
SD	2.31	0.47	0.61	0.48	0.53	2.87	4.11

Table 4.5: Parameter Estimates of Second Simulated Data

4.3.3 Cross-Validated Error Rates for Second Simulated Data

The cross-validated misclassification error rates for the other simulated data is tabulated below. The reported error rates are based on the type of prediction using the Bayesian approach and compared to the output from the POLR method. As we expected, the reported error rates here are less as compared to the first simulated data. This is evidenced by the data ellipsoid plot in figure (4.4) above which has much separation within groups of the variables as compared to the first simulated data. The error rate using the Bayesian method here outperforms the POLR method.

Prediction Type	**Error Rate**
Point Estimates	0.01
By Probabilities	0.01
By Weighted Average	0.01
POLR Method	0.08

Table 4.6: Cross-Validated Misclassifiaction Error Rates for Second Simulated Data

The confusion matrix table for the second simulated data is shown below

Actual	1	2	3
1	30	1	0
Predicted 2	0	29	0
3	0	0	30

Table 4.7: Bayesian Confusion Matrix for 2nd Simulated Data

Actual	1	2	3
1	28	1	0
Predicted 2	2	27	2
3	0	2	28

Table 4.8: POLR Confusion Matrix for 2nd Simulated Data

4.4 Real Data Application

For the purpose of this work, we apply the analysis to the well-known Iris flower data set and the measurements of male Egyptian skull data set. These two different data sets are used due to their differences in their mean and variation structure within their groups. We shall explore further to see the details of these variations and how it affects the model prediction.

4.4.1 Iris Data Description

We use the well known Iris flower data set as our benchmark for the analysis. The Iris flower data is a multivariate data set that was introduced by the British statistician Ronald Fisher in his work (Fisher, 1936). It is sometimes called Anderson's Iris data set because Edgar Anderson collected the data to quantify the morphological variation of Iris flowers of three related species. Two of which were collected in the Gasp Peninsula "all from the same pasture, picked and measured at the same time by the same person with the same apparatus"(Anderson, 1935). The data set consists of fifty samples from each of three species of Iris (Iris Setosa, Iris Versicolor and Iris Versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

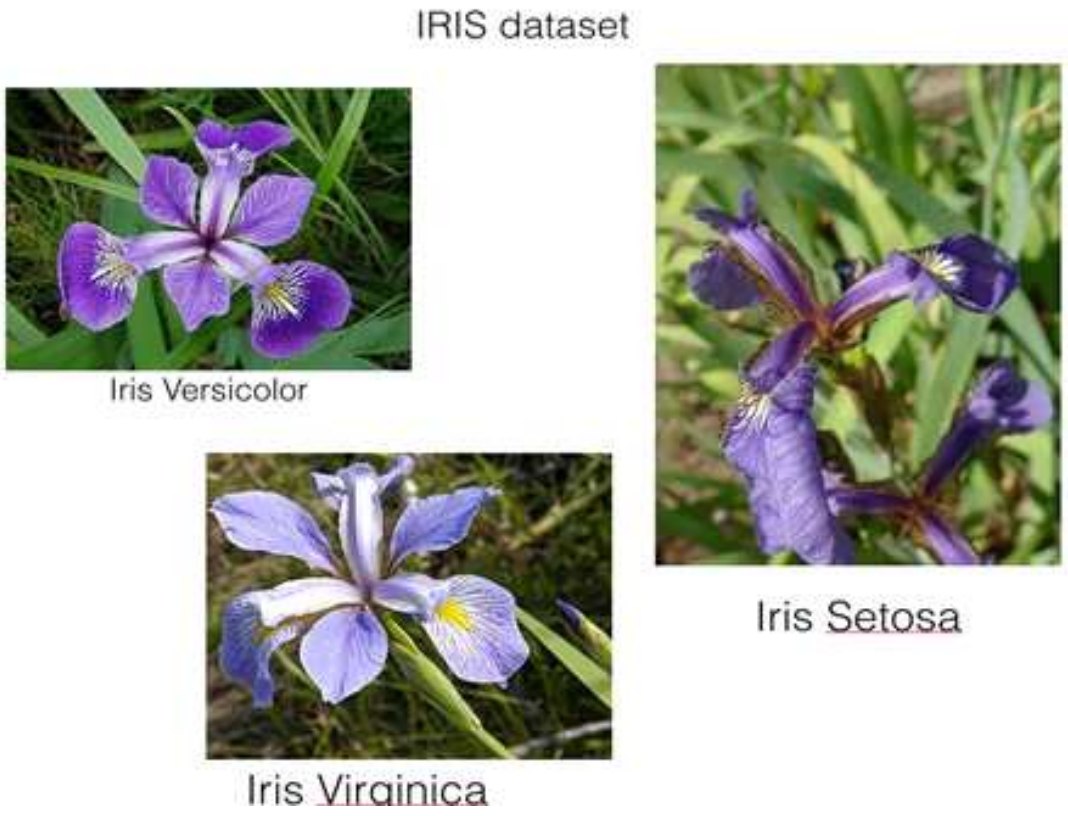

Figure 4.5: The Three Different Species of Iris Flower

4.4.2 Data Ellipsoid Plot for Iris Data

The data ellipsoid plot for the Iris data is also shown below to visualize the structural differences existing between the measurements of the parts of the flowers across the various group of species.

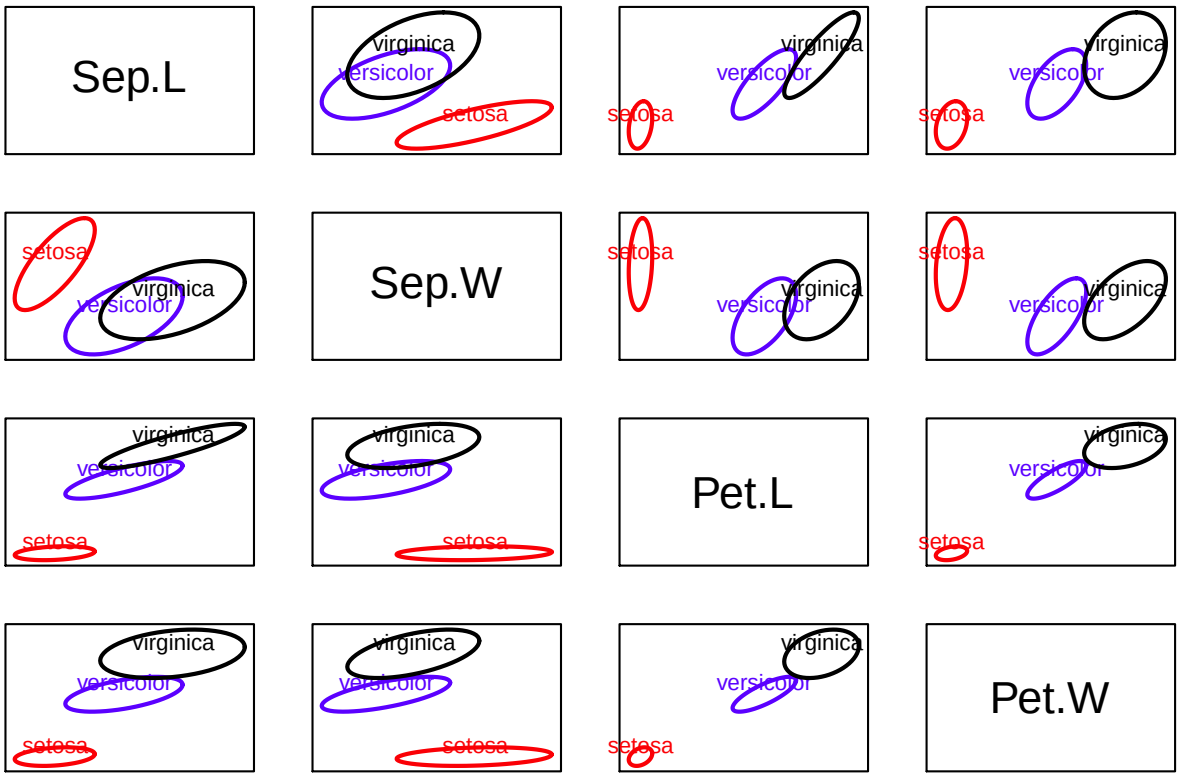

Figure 4.6: Data Ellipsoid Plot for Iris Data

From the above, we see that the measurements are well separated across the various species with only a few of the Versicolor and Virginica overlapping in some cases. The Setosa species is far separated from the other two species. Thus we should expect to see less misclassification error rate after prediction, with most of the misclassified categories coming from the Virginica and the Versicolor species.

4.4.3 MCMC Diagnostics for Iris Data

We again make some MCMC diagnostics for the Gibbs sampler on the real (Iris) data using the trace plot as it was done for the simulated data. We see from the trace plots that the $\beta's$ mix well whereas the $\delta's$ have a slow mixing effect, as the samples from the iterations

exhibit some serial dependencies.

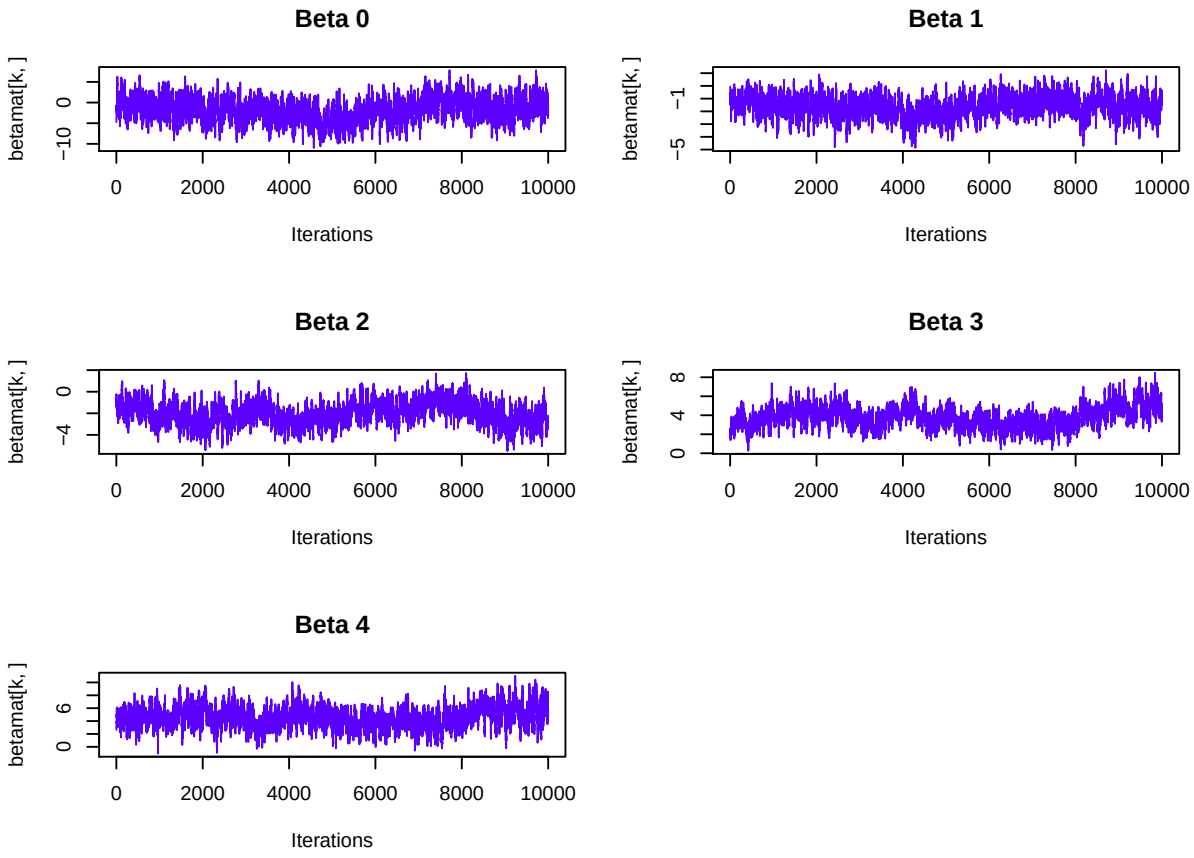

Figure 4.7: Trace Plot of $\beta's$ from Iris Data

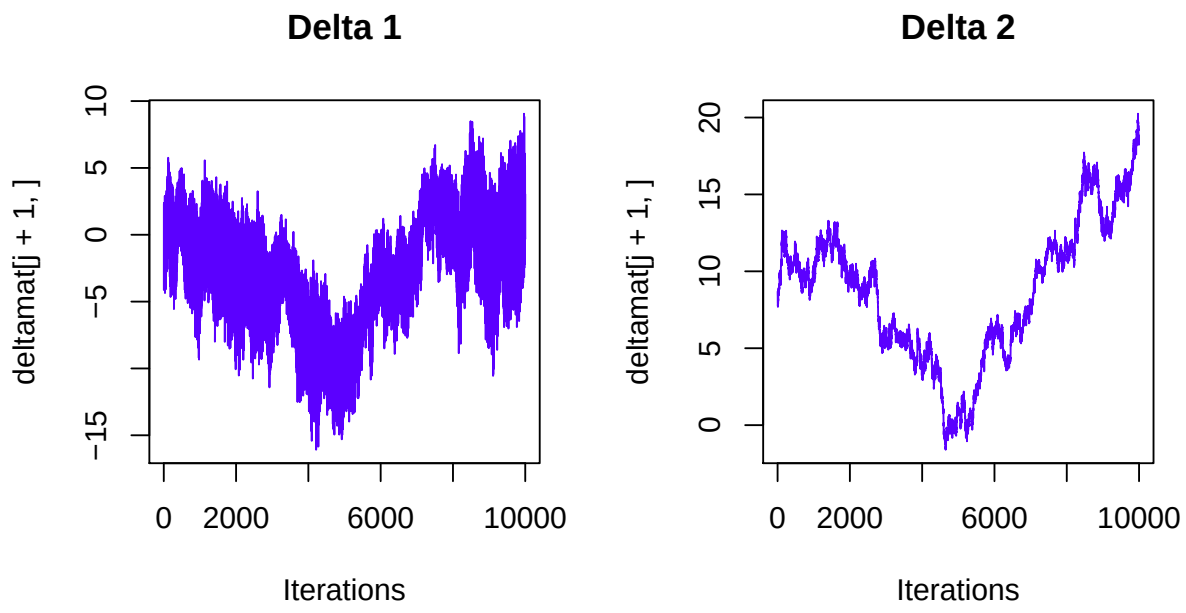

Figure 4.8: Trace Plot of $\delta's$ from Iris Data

4.4.4 Parameter Estimates of Iris Data

The parameter estimates of the Iris data from the last CV iterations of the Gibbs sampling are shown below.

Parameters	β_0	β_1	β_2	β_3	β_4	δ_1	δ_2
Mean	-2.0	-1.65	-2.04	-3.78	4.55	-3.24	8.10
SD	2.84	0.86	1.07	1.17	1.70	4.38	4.92

Table 4.9: Parameter Estimates of Iris Data

4.4.5 Cross Validated Error Rates For Iris Data

The cross-validated misclassification error rates for the Iris data is tabulated below. The reported error rates based on the three types of the predictions from the Bayesian approach

as well as the POLR method is displayed in the table below.

Prediction Type	Error Rate
Point Estimates	0.013
By Probabilities	0.013
By Weighted Average	0.013
POLR Method	0.03

Table 4.10: Cross-Validated Misclassifiaction Error Rates For Iris Data

The summary of the table of confusion matrices for both the Bayesian and the POLR methods of the Iris data set is shown below, where 1, 2, and 3 represent the Setosa, Versicolor, and Virginica species respectively.

Actual Species	1	2	3
1	50	0	0
Predicted Species 2	0	48	0
3	0	2	50

Table 4.11: Bayesian Confusion Matrix for Iris Data

Actual Species	1	2	3
1	50	0	0
Predicted Species 2	0	47	1
3	0	3	49

Table 4.12: POLR Confusion Matrix for Iris Data

4.5 Skull Data Description

The skull data set was obtained from R embedded in the package "HSAUR." The data consists of four physical measurements in millimeters of 150 male Egyptian skulls from five epochs (periods) (Thomson and Randall-MacIver, 1905). Period 1 (4000 BC), period 2 (3300 BC), Period 3 (1850 BC), Period 4 (c200BC), and Period 5 (cAD150). The measures are maximal breadth (mb), basibregmatic height (bh), basialveolar length (bl), and nasal height (nh) of each skull. Researchers claim that a change in skull measurement is as

a result of the time duration. Systematic changes over time could indicate interbreeding among migrant populations (or the influence of other factors). The interest in this analysis, however, lies in the ability to predict well which period these measurements fall within. The figure below gives a label for these measurements of a typical skull.

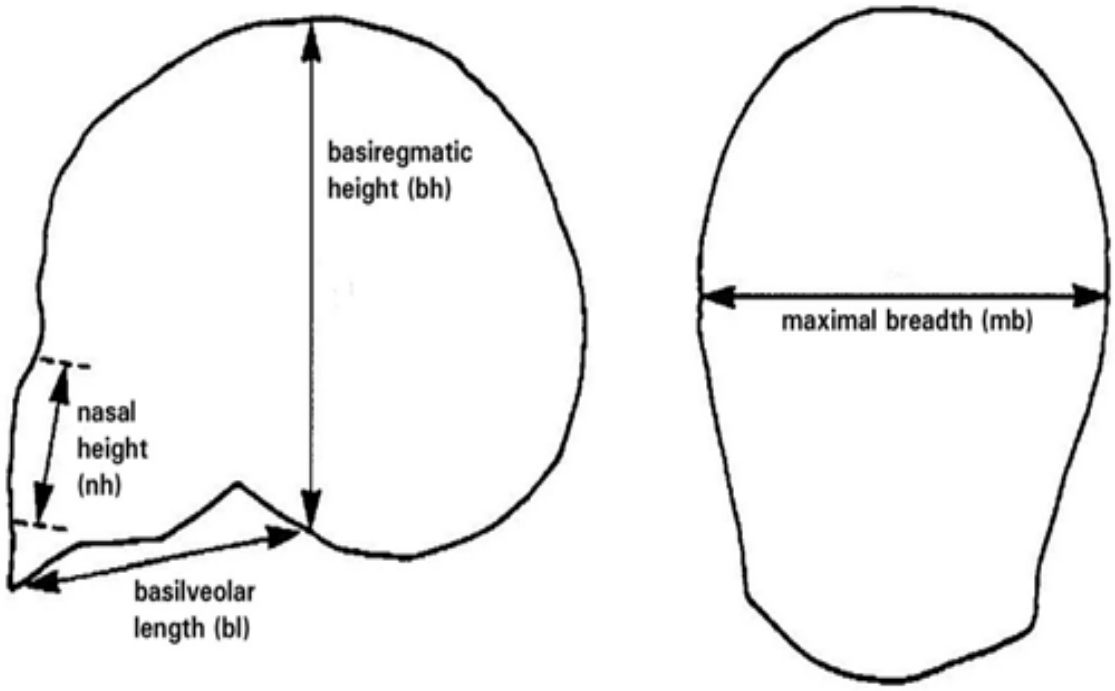

Figure 4.9: A Labelled Male Egyptian Skull

4.5.1 Data Ellipsoid Plot for Skull Data

The data ellipsoid plot for the skull data is also shown below to visualize the structural differences existing between the measurements of the parts of the skull across the various group of periods.

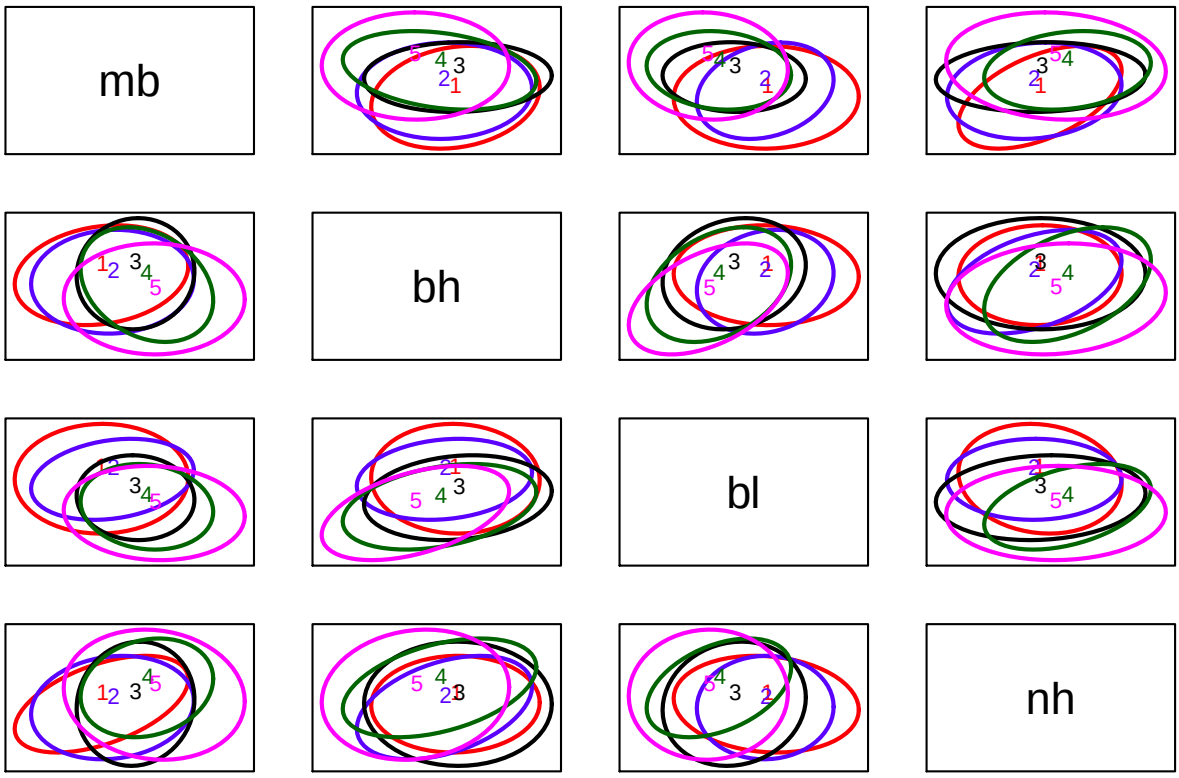

Figure 4.10: Data Ellipsoid Plot for Skull Data

As seen by the data ellipsoid plot above, the measurements of the skull overlap each other across the various periods with no evidence of separations of their means. The measurements tend to cluster around a common centroid. In this case, we should expect to see higher misclassification error rate.

4.5.2 MCMC Diagnostics for Skull Data

As evidenced from the trace plots, the $\beta's$ and $\delta's$ mix or explore well in the parameter space.

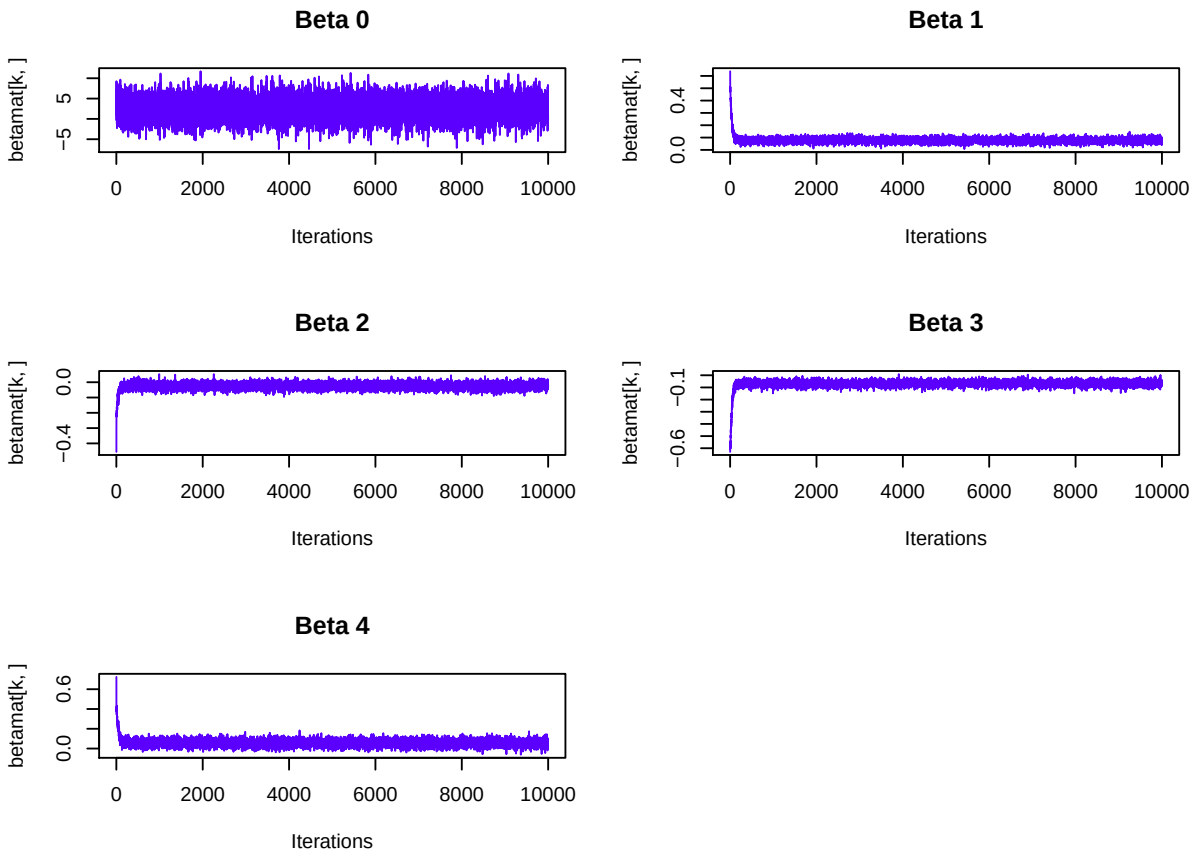

Figure 4.11: Trace Plot of $\beta's$ for Skull Data

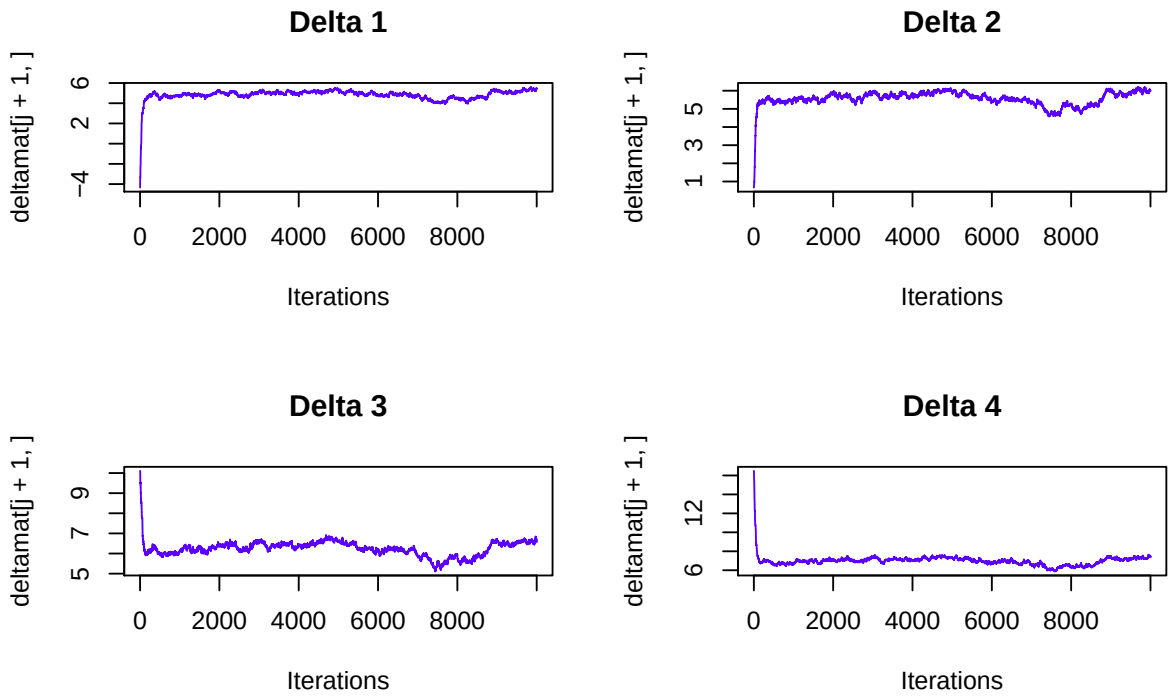

Figure 4.12: Trace Plot of $\delta's$ for Skull Data

Below is the cross-validation misclassification error rates obtained from the skull data.

Prediction Type	**Error Rate**
Point Estimates	0.68
By Probabilities	0.69
By Weighted Average	0.68
POLR Method	0.71

Table 4.13: Cross-Validated Misclassifiaction Error Rates for Skull Data

The confusion matrix table for the skull data set is shown below.

Actual	1	2	3	4	5
1	16	12	3	3	2
2	5	7	5	3	3
Predicted 3	4	6	7	7	5
4	3	3	11	7	9
5	2	2	4	10	11

Table 4.14: Bayesian Confusion Matrix for Skull Data

Actual	1	2	3	4	5
1	14	13	3	2	2
2	5	6	5	4	3
Predicted 3	5	6	7	8	5
4	3	3	11	6	9
5	2	2	4	10	11

Table 4.15: POLR Confusion Matrix for Skull Data

The skull data which has less separability in the means of the variables as compared to the Iris data, yielded a misclassification error rate of 0.68 using the Bayesian approach and 0.71 using the POLR model. The higher error rates for the skull data in both prediction approach is not surprising as we see from how the measurements overlap each other across the groups. This justifies the reliability of the model to classify or predict correctly if the variables are well separated across the given categories. In the skull data set, we could logically infer that the measurements of these male Egyptian skulls do not change significantly across the given periods.

Chapter 5

Summary and Conclusions

5.1 Conclusions

The study aims to use the Bayesian approach to predict ordinal outcomes through some latent variable, and comparison of the results is made to the well-known Polytomous Ordinal Logistic Regression model (POLR). As already discussed, the challenge that arises in the modeling of ordinal data is the estimation of the cutoff point parameters, which links a continuous latent variable to the ordinal outcomes. The method proposed by (Albert and Chib, 1993) where they in-cooperated a vague prior on the cutoff parameters has a slow convergence rate for a large sample size.

The procedure adopted in this work is very similar to that of (Albert and Chib, 1993) and (Kwon et al., 2007), however, in our case we used an informative prior, and we established a correspondence between these cutoff parameters and the ordinal outcomes via the Dirichlet distribution. We implement the Gibbs sampling to estimate the parameters from their full conditional posteriors. We then compare the results from our method to the POLR model. In terms of predictability in this work, our method overall outperforms the traditional well known POLR model as seen in the various reported misclassification error rates.

References

Agresti, A. (1996). Multicategory logit models. *An introduction to categorical data analysis*, pages 173–196.

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679.

Anderson, E. (1935). The irises of the gaspe peninsula. *Bulletin of the American Iris society*, 59:2–5.

Cowles, M. K., Carlin, B. P., and Connett, J. E. (1996). Bayesian tobit modeling of longitudinal ordinal clinical trial compliance data with nonignorable missingness. *Journal of the American Statistical Association*, 91(433):86–98.

Dempster, A. P. (1969). Elements of continuous multivariate analysis. Technical report.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

Friendly, M., Monette, G., and Fox (2013). Elliptical insights: understanding statistical methods through elliptical geometry. *Statistical Science*, 28(1):1–39.

Friendly, M. and Sigal, M. (2018). Visualizing tests for equality of covariance matrices. *arXiv preprint arXiv:1805.05756*.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC.

Johnson, R. and Wichern, D. (1992). Applied multivariate statistical methods. *Prentice Hall, Englewood Cliffs, NJ*.

Kwon, D., Tadesse, M. G., Sha, N., Pfeiffer, R. M., and Vannucci, M. (2007). Identifying biomarkers from mass spectrometry data with ordinal outcome. *Cancer Informatics*, 3:19 – 28.

McCullagh, P. (1980a). Regression models for ordinal data. *Journal of the royal statistical society. Series B (Methodological)*, pages 109–142.

McCullagh, P. (1980b). Regression models for ordinal data.

Sirisrisakulchai, J. and Sriboonchitta, S. (2016). Causal effect for ordinal outcomes from observational data: Bayesian approach. *Thai Journal of Mathematics*, pages 63–70.

Thomson, A. and Randall-MacIver, D. (1905). *The Ancient Races of the Thebaid: Being an Anthropometrical Study of the Inhabitants of Upper Egypt from the Earliest Prehistoric Times to the Mohammedan Conquest, Based Upon the Examination of Over 1,500 Crania*. Clarendon Press.

Tu, S. (2014). The dirichlet-multinomial and dirichlet-categorical models for bayesian inference. *Computer Science Division, UC Berkeley*.

Walker, S. H. and Duncan, D. B. (1967). Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179.

Zhou, X. (2006). *Bayesian inference for ordinal data*. PhD thesis, Rice University.

Appendix A

Derivation of Posterior Mean and Variance of $\boldsymbol{\beta}$

$$\text{From } \; Y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \epsilon_i, \; \epsilon_i \sim N(0,1), \quad Y|X \sim N(\boldsymbol{x}_i'\boldsymbol{\beta}, 1) \tag{5.1}$$

assuming that $\boldsymbol{\beta} \sim MVN(\boldsymbol{\beta}_0, \boldsymbol{\Sigma}_0)$ and also

$\boldsymbol{\beta}$ has a normal prior then the posterior is:

$$\pi(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) \propto L(\boldsymbol{\beta}, \boldsymbol{\delta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})\pi(\boldsymbol{\beta}) \tag{5.2}$$

$$\text{but } L(\boldsymbol{\beta}, \boldsymbol{\delta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})\pi(\boldsymbol{\beta}) = \prod_{i=1}^{n} f(Y_i|\boldsymbol{\beta}) \propto \exp\Big[-\frac{1}{2}\sum_{i=1}^{n}(Y_i - X\boldsymbol{\beta})^2\Big]$$

and our prior

$$\pi(\boldsymbol{\beta}) \propto \exp\Big[-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)'\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\Big]$$

$$\pi(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) \propto \exp\Big[-\frac{1}{2}\sum_{i=1}^{n}(Y_i - \boldsymbol{X}'\boldsymbol{\beta})^2\Big] \times \exp\Big[-\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)'\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta} - \boldsymbol{\beta_0})\Big]$$

$$\pi(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) \propto \exp\Big[-\frac{1}{2}\sum_{i=1}^{n}(y_i - \boldsymbol{X}'\boldsymbol{\beta})^2 + (\boldsymbol{\beta} - \boldsymbol{\beta_0})'\boldsymbol{\Sigma_0^{-1}}(\boldsymbol{\beta} - \boldsymbol{\beta_0})\Big] \tag{A}$$

using the fact that

$$\sum_{i=1}^{n}(Y_i - \boldsymbol{X}'\boldsymbol{\beta})^2 = (Y - \boldsymbol{X}\boldsymbol{\beta})'(Y - \boldsymbol{X}\boldsymbol{\beta}) = (Y' - \boldsymbol{\beta}'\boldsymbol{X}')(Y - \boldsymbol{X}\boldsymbol{\beta})$$

$$\sum_{i=1}^{n}(Y_i - \boldsymbol{X}'\boldsymbol{\beta})^2 = \boldsymbol{Y}'\boldsymbol{Y} - 2\boldsymbol{Y}'\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta}$$

we pick terms depending on $\boldsymbol{\beta}$

$$\sum_{i=1}^{n}(Y_i - \boldsymbol{X}'\boldsymbol{\beta})^2 = -2\boldsymbol{Y}'\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\boldsymbol{X}'\boldsymbol{X}\boldsymbol{\beta} + \text{constant} \tag{B}$$

$$(\boldsymbol{\beta} - \boldsymbol{\beta_0})'\boldsymbol{\Sigma_0^{-1}}(\boldsymbol{\beta} - \boldsymbol{\beta_0}) = \boldsymbol{\beta}'\boldsymbol{\Sigma_0^{-1}}\boldsymbol{\beta} - 2\boldsymbol{\beta}_0'\boldsymbol{\Sigma_0^{-1}}\boldsymbol{\beta} + \text{constant} \tag{C}$$

substituting (B) and (C) into (A)

$$\pi(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) \propto \exp\Big[-\frac{1}{2}\Big(-2\boldsymbol{Y'X\beta} + \boldsymbol{\beta' X'X\beta} + \boldsymbol{\beta'\Sigma_0^{-1}\beta} - 2\boldsymbol{\beta_0'\Sigma_0^{-1}\beta}\Big)\Big]$$

$$\text{but} \quad \boldsymbol{\beta' X'X\beta} + \boldsymbol{\beta'\Sigma_0^{-1}\beta} = \boldsymbol{\beta'(X'X + \Sigma_0^{-1})\beta}$$

$$\text{and} \quad -2\boldsymbol{Y'X\beta} + -2\boldsymbol{\beta_0'\Sigma_0^{-1}\beta} = -2\boldsymbol{(Y'X + \beta_0'\Sigma_0^{-1})\beta}$$

$$\pi(\boldsymbol{\beta} \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) \propto \exp\Big[-\frac{1}{2}\Big(\boldsymbol{\beta'(X'X + \Sigma_0^{-1})\beta} - 2\boldsymbol{(Y'X + \beta_0'\Sigma_0^{-1})\beta}\Big)\Big]$$

Comparing the above to the kernel form of the Multivariate Gaussian distribution i.e

$$\boldsymbol{(\beta - \tilde{\beta})'\tilde{\Sigma}^{-1}(\beta - \tilde{\beta})}$$

to get our posterior mean for $\boldsymbol{\beta}$ i.e. $\tilde{\boldsymbol{\beta}}$

$$\boldsymbol{(\beta - \tilde{\beta})'\tilde{\Sigma}^{-1}(\beta - \tilde{\beta})} = \boldsymbol{\beta'\tilde{\Sigma}^{-1}\beta} - 2\boldsymbol{\tilde{\beta}'\tilde{\Sigma}^{-1}\beta} + \text{constant}$$

Comparing the above Kernel to the that of the posterior of $\boldsymbol{\beta}$, we see that:

$$\boldsymbol{\beta'(X'X + \Sigma_0^{-1})\beta} = \boldsymbol{\beta'\tilde{\Sigma}^{-1}\beta}$$

$$\Rightarrow \boldsymbol{\tilde{\Sigma}} = \boldsymbol{(X'X + \Sigma_0^{-1})^{-1}}$$

Thus the posterior variance is:

$$\boldsymbol{\tilde{\Sigma}} = \boldsymbol{(X'X + \Sigma_0^{-1})^{-1}} \tag{5.3}$$

And also by little algebra we obtain our posterior mean as:

$$-2\boldsymbol{(Y'X + \beta_0'\Sigma_0^{-1})\beta} = -2\boldsymbol{\tilde{\beta}'\tilde{\Sigma}^{-1}\beta}$$

$$\tilde{\boldsymbol{\beta}} = [\boldsymbol{(Y'X + \beta_0'\Sigma_0^{-1})\tilde{\Sigma}}]'\tilde{\boldsymbol{\beta}} = [\boldsymbol{\Sigma'(Y'X + \beta_0'\Sigma_0^{-1})'}]$$

$$\tilde{\boldsymbol{\beta}} = [\boldsymbol{(X'X + \Sigma_0^{-1})^{-1}(X'Y + \Sigma_0^{-1}\beta_0)}]$$

Since the posterior variance i.e.

$$\boldsymbol{\tilde{\Sigma}} = \boldsymbol{(X'X + \Sigma_0^{-1})^{-1}}$$

Then by substitution, the posterior mean is thus:

$$\tilde{\boldsymbol{\beta}} = [\tilde{\boldsymbol{\Sigma}}(\boldsymbol{X'Y} + \boldsymbol{\Sigma_0^{-1}}\boldsymbol{\beta}_0)] \tag{5.4}$$

Proof of (3.13)

Let w be the random variable and we want to show that:

$$w = \frac{F(\delta_j) - F(\delta_{j-1})}{F(\delta_{j+1}) - F(\delta_{j-1})} \sim \text{Beta}(\alpha_j, \alpha_{j+1}), \text{truncated at } w_1 < w < w_2 \tag{5.5}$$

$$\text{let } F(\delta_{j-1}) = a, \text{and } F(\delta_{j+1}) = b,$$

$$\text{then } w = \frac{F(\delta_j) - a}{b - a} \sim \text{Beta}(\alpha_j, \alpha_{j+1})$$

$$F(\delta_j) = (b-a)w + a, \implies \delta_j = F^{-1}[(b-a)w + a] \tag{5.6}$$

we want a distribution in terms of w , by transformation

$$f_W(w) = \pi_\delta(\delta_j) \times \frac{\partial \delta_j}{\partial w} = \pi_\delta(F^{-1}((b-a)w + a)) \times \frac{\partial \delta_j}{\partial w} \tag{5.7}$$

but from (3.12),

$$\pi_\delta(F^{-1}((b-a)w + a)) = [F(\delta_j) - a]^{\alpha_j - 1}[b - F(\delta_j)]^{\alpha_{j+1} - 1} f(\delta_j)$$

substituting $F(\delta_j) = (b-a)w + a$ into the above,

$$\pi_\delta(F^{-1}((b-a)w + a)) = [(b-a)w + a - a]^{\alpha_j - 1} \times [b - (b-a)w - a]^{\alpha_{j+1} - 1} f(\delta_j)$$

$$\pi_\delta(F^{-1}((b-a)w + a)) = (b-a)^{\alpha_{j-1}} w^{\alpha_{j-1}} (b-a)^{\alpha_{j+1} - 1} (1-w)^{\alpha_{j+1} - 1} f(\delta_j).$$

ignoring terms that does not depend on w;

$$\pi_\delta(\delta_j) = \pi_\delta(F^{-1}((b-a)w + a)) \propto w^{\alpha_{j-1}} (1-w)^{\alpha_{j+1} - 1} f(\delta_j) \tag{5.8}$$

we now proceed to find $\frac{\partial \delta_j}{\partial w}$. But since $\delta_j = F^{-1}[(b-a)w + a]$, to avoid the complexity in computing $\frac{\partial \delta_j}{\partial w}$, we rather find:

$$\frac{\partial \delta_j}{\partial w} = \frac{1}{\frac{\partial w}{\partial j}}, \qquad w = \frac{F(\delta_j) - a}{b-a}$$

$$\text{but } \frac{\partial w}{\partial j} = \frac{f(\delta_j)}{b-a}, \text{ which } \implies \frac{\partial \delta_j}{\partial w} = \frac{b-a}{f(\delta_j)}$$

now substituting (5.8) i.e $\pi_\delta(\delta_j)$ and $\frac{\partial \delta_j}{\partial w}$ into (5.7), we have:

$$f_W(w) \propto w^{\alpha_{j-1}}(1-w)^{\alpha_{j+1}-1} f(\delta_j) \times \frac{b-a}{f(\delta_j)}$$

$$\therefore \; f_W(w) \propto w^{\alpha_{j-1}}(1-w)^{\alpha_{j+1}-1} \tag{5.9}$$

which looks exactly as the kernel of the beta distribution, hence:

$$w \sim \text{Beta}(\alpha_j, \alpha_{j+1}) \tag{5.10}$$

with w truncated in the interval

$$\frac{F(\delta_{j,1}) - F(\delta_{j-1})}{F(\delta_{j+1}) - F(\delta_{j-1})} \leq w \leq \frac{F(\delta_{j,2}) - F(\delta_{j-1})}{F(\delta_{j+1}) - F(\delta_{j-1})} \tag{5.11}$$

Appendix B

```
## Read in the simulated data ##
library(plyr); #library(dplyr)
library(mcmc);library(psycho);#library(tidyverse)
library(mvtnorm)
## Simulating data ###
p=4;J=3;nsub=c(30,30,30) # Sub sample size for each category
rhovec=c(0.1,0.5,0.9)
mu=c(3,2,4,1,3,-2,4,-1,-3,-2,-4,-1) # First Simulated data
#mu=c(5,1,4,6,3,-2,-4,-1,-5,7,4,-10)  # Second Simulated data
mu=matrix(mu, nrow = J, byrow = T)
Idmat=diag(1,p);onevec=rep(1,p)
sigma =  2
Xmat=vector(mode="numeric",length=0)
set.seed(111)
for (i in 1:J){
  cormat = (1-rhovec[i])*Idmat + rhovec[i]*onevec%*%t(onevec)
  covmat = sigma^(2)*cormat
  x=rmvnorm(nsub[i],mu[i,],covmat)
  u=cbind(x,rep(i,nsub[i]))
  Xmat=rbind(Xmat,u)
}
N=nrow(Xmat)
onescol=as.matrix(rep(1,N)) ## Add a column of ones to the data
Xmat=cbind(onescol,Xmat)
write.table(Xmat,"simdata.csv",sep=",",row.names=FALSE)# Save sim data
```

```
Dmat=read.table("C:\\Users\\Benard\\Desktop\\My␣Proj\\simdata.csv",
header= T, sep = ",") ## Read in the sim data

#########################################################
### Skull Data set implementation
## Read and prepare the Skull data ##
# data("skulls", package = "HSAUR2");skull=skulls
# skull$epoch <- gsub('c4000BC', '1', skull$epoch)
# skull$epoch <- gsub('c3300BC', '2', skull$epoch)
# skull$epoch <- gsub('c1850BC', '3', skull$epoch)
# skull$epoch <- gsub('c200BC', '4', skull$epoch)
# skull$epoch <- gsub('cAD150', '5', skull$epoch)
# N=nrow(skull)
# onescol=as.matrix(rep(1,N)) ## Add a column of ones to the data
# skull=cbind(onescol,skull)
# skull <- skull[,c("onescol", "mb", "bh", "bl", "nh", "epoch")]
# attach(skull);library(plyr)
## Rename the columns
#Dmat=rename(skull,c("onescol"="V1","mb"="V2","bh"="V3",
"bl"="V4","nh"="V5","epoch"="V6"))
#Dmat$V6=as.numeric(Dmat$V6)
#J=length(unique(Dmat$V6))
#########################################################

#Use the codes below as begining loop for CV for each data type

### For Simulated Data
#Nrow=5 # Number of rows to pick from each category of sim data
#for (h in 1:6){
# nr=Nrow*(h-1)+1
```

```
#rownum=c(nr:(nr+4),(nr+30):(nr+34),(nr+60):(nr+64))
##=====================================================#
# For Skull Data
#Nrow=5 # Number of rows to pick from each category
#for (h in 1:6){
#nr=Nrow*(h-1)+1
#rownum=c(nr:(nr+4),(nr+30):(nr+34),(nr+60):(nr+64),(nr+90):(nr+94),
#(nr+120):(nr+124))
##=====================================================##

#####################################################################

# Posterior sampling and Prediction
# Start the clock!
ptm <- proc.time()
# Empty vectors to store the Error rates from Cross Validation
point.est.errorvec=vector(mode="numeric",length=0)
prob.pred.errorvec=vector(mode="numeric",length=0)
weight.pred.errorvec=vector(mode="numeric",length=0)
Smat.point = matrix(0,J,J)
Smat.prob =  matrix(0,J,J)
Smat.Weight.Average =  matrix(0,J,J)

# For Simulated Data
Nrow=5 # Number of rows to pick from each category of sim data
for (h in 1:6){
  nr=Nrow*(h-1)+1
  rownum=c(nr:(nr+4),(nr+30):(nr+34),(nr+60):(nr+64))

  test=Dmat[rownum,]
```

```
train=Dmat[-rownum,]
# Gibbs Sampling/Sampling from the Posterior #
library(tmvtnorm);library(sandwich);library(gmm);library(Matrix)
library(evd);library(truncdist);library(stats4)
library(truncnorm);library(MASS)
N=nrow(Dmat);P=ncol(Dmat)-1; #number of covariates
including ones in the first column.
Z=train[,P+1] #  Response variable of training data
X = train[,-(P+1)]; X=as.matrix(X)
truez=test[,(P+1)]
xnew=test[,-(P+1)];# removing Z cloumn from the test data
xnew=as.matrix(xnew) # New data to be used for testing.
N1=nrow(X) # total sample size of training data
J=length(unique(Z)) #number of categories
# Initial value settings for Y i.e. Y|X ~ N(X'B,1) #
set.seed(12111)
betasd=1
Beta=rnorm(P, mean = 0, sd = betasd) # Initial values for Beta (B)
#Initial values of delta#
set.seed(1211)
deltavec=runif(J-1,-20,20);deltavec=sort(deltavec)
deltavec=c(-Inf,deltavec,Inf)
alphavec=rep(1,J) # shape and scale parameters of trunc Beta.
stdnorm= 10 # SD value for finding CDF of the delta posterior.
### Set empty vectors to store results##
ymat=vector(mode="numeric",length=0)
betamat=vector(mode="numeric",length=0)
deltamat=vector(mode="numeric",length=0)
####Prior information for beta ###
c=10
```

```
sigma_not = c*diag(P); beta_not = rep(0, P)
beta_not = as.matrix(beta_not)
Inv_sigma_not = solve(sigma_not)

###Gibbs Sampling#######
#=======================================================#
set.seed(1111);niter=10000; burning=2000
for (i in 1:niter){
  L1=deltavec[Z];L2=deltavec[Z+1] #lower/upper truncation intervals.
  ymean=X%*%Beta  ## Mean of Y|X
  y=rtruncnorm(N1, a=L1, b= L2, mean = ymean, sd = 1)
  y=as.matrix(y)
  ymat=cbind(ymat,y)
  # Generating samples from posterior dist. of Beta|data
  post_var = solve(t(X)%*%X + Inv_sigma_not)
  post_mean = post_var%*%(t(X)%*%y +Inv_sigma_not%*%beta_not)
  Beta=rmvnorm(1,post_mean, post_var)
  Beta=t(Beta)
  betamat = cbind(betamat,Beta)
  ###### Sampling for the delta's#######
  for (j in 1: (J-1)){
    ycat1=y[Z==j];c1=max(ycat1);ycat2=y[Z==j+1];c2=min(ycat2)
    a=pnorm(deltavec[j],mean=0,stdnorm)
    b=pnorm(deltavec[j+2],mean=0,stdnorm)
    w1=(pnorm(c1,0,stdnorm)-a)/(b-a)
    w2=(pnorm(c2,0,stdnorm)-a)/(b-a)
    w=rtrunc(1,spec="beta",shape1=alphavec[j],
    shape2=alphavec[j+1],a=w1,b=w2)
    deltavec[j+1]=qnorm((b-a)*w+a, mean=0,stdnorm)
  }
```

```
  deltamat=cbind(deltamat,deltavec)
}
#Diagnostic Plots for determinning Burn-in periods of Beta#
#=========================================================#
par(mfrow=c(3,2))
for (k in 1:P){
  plot(betamat[k,],type="l", xlab="Iterations",
  col="blue",main = paste("Beta", k-1))
}
#Diagnostic Plots for determinning Burn-in periods of delta#
#==========================================================#
par(mfrow=c(2,2))
for (j in 1:(J-1)){
  plot(deltamat[j+1,],type="l",
  xlab="Iterations",col="blue",main = paste("Delta", j))
}
######Estimates for simulated Y ######
##Estimation of Parameters##
##=======================================##
### Y estimates after discarding first burn-in periods
yhat.mean = apply(ymat[,-c(1:burning)], 1, mean)
yhat.sd = apply(ymat[,-c(1:burning)], 1, sd)
### Beta estimates after discarding first burn-in periods ##
betahat.mean=apply(betamat[,-c(1:burning)], 1, mean)
betahat.sd=apply(betamat[,-c(1:burning)], 1, sd)

### Delta estimates after discarding first burn-in periods##
deltamat=deltamat[c(2:J),]
deltahat.mean=apply(deltamat[,-c(1:burning)], 1, mean)
deltahat.sd=apply(deltamat[,-c(1:burning)], 1, sd)
```

```
#### ACF plots to determine model mixing and convergence
#=======================================================#
# ACF plots of Beta
#betamat.burn=betamat[,-c(1:burning)]
par(mfrow=c(3,2))
for (k in 1:P){
  acf(betamat[k,], xlab="Lag",col="blue",
  main = paste("acf of Beta", k-1))
}
# ACF plots of Delta
#deltamat.burn=deltamat[,-c(1:burning)]
par(mfrow=c(2,2))
for (j in 1:(J-1)){
  acf(deltamat[j,], xlab="Lag",col="blue",
  main = paste("ACF of Delta", j))
}

################### Predictions ########################

# (1) Prediction using the latent variable Y=XB #####
# i. Prediction Method 1, using Beta as estimates from gibbs.
yhatvec=xnew%*%betahat.mean
yhatvec=as.vector(yhatvec) # Point Estimates of Y based on Xnew
### This code will identify which category y falls in.
zhatpred=vector(mode="numeric",length=0)
deltahatvec=c(-Inf,deltahat.mean,Inf)
for (k in 1:length(yhatvec)){
  yhat=yhatvec[k];
  zhat=which(yhat<=deltahatvec)[1]-1;
  zhatpred=cbind(zhatpred,zhat)
```

```
}
length(zhatpred)
# Confusion matrix for finding missclassification error rate
confmat=table(zhatpred,truez);
Smat.point=Smat.point+confmat
point.est.error =1-sum(diag(confmat))/sum(confmat)
point.est.errorvec=cbind(point.est.errorvec,point.est.error)

#### Prediction Via Probability.###
ynew=xnew%*%betahat.mean ;ynew=as.vector(ynew) ## Y|X ~ N(X'B,1)
## The code below gives the prob of each y_{i} falling in each of
#the category and obtain the category with the maximum probability.
deltahatvec=c(-Inf,deltahat.mean,Inf)
yprobvec=vector(mode="numeric",length=0)
for (k in 1:length(ynew)) {
  yprob=pnorm(deltahatvec,mean=ynew[k],sd=1)
  yprob=diff(yprob)
  yprobvec=rbind(yprobvec,yprob)
}
library(ramify)
zpred=argmax(yprobvec,rows=TRUE)
confmatrix=table(zpred,truez);
Smat.prob = Smat.prob+confmatrix
prob.error =1-sum(diag(confmatrix))/sum(confmatrix)
prob.pred.errorvec = cbind(prob.pred.errorvec,prob.error)
### Prediction by finding the weighted average of the predicted Z
catvec=seq(1:J) ## Vector of categorical values
zbarpred=round(yprobvec%*%catvec)
confmat1= table(zbarpred, truez); #confmat1 ## Confusion matrix
Smat.Weight.Average = Smat.Weight.Average + confmat1
```

```
  pred.weight.err= 1-sum(diag(confmat1))/sum(confmat1)
  weight.pred.errorvec = cbind(weight.pred.errorvec,pred.weight.err)
}
# Stop the clock
proc.time() - ptm ## Calculate time for Gibbs sampling
point.est.errorvec;prob.pred.errorvec;weight.pred.errorvec
## Overall average error rates from the predictions
Error.Point=mean(point.est.errorvec)
Error.Prob=mean(prob.pred.errorvec)
Error.weighted=mean(weight.pred.errorvec)
Predicted_Error_rates=cbind(Error.Point,Error.Prob,Error.weighted)
Predicted_Error_rates
## Misclassification Tables
Smat.point;  Smat.prob; Smat.Weight.Average
#=========================================================#

### R Codes For Implementing the POLR Model #####
## Simulating Data ###
p=4;J=3;nsub=c(30,30,30) # Sub sample size for each category
rhovec=c(0.1,0.5,0.9,0.2)
mu=c(3,2,4,1,3,-2,4,-1,-3,-2,-4,-1) # First sim data
#mu=c(5,1,4,6,3,-2,-4,-1,-5,7,4,-10) # Second sim data
mu=matrix(mu, nrow = J, byrow = T)
Idmat=diag(1,p);onevec=rep(1,p)
sigma =  2
Xmat=vector(mode="numeric",length=0)
set.seed(111)
for (i in 1:J){
  cormat = (1-rhovec[i])*Idmat + rhovec[i]*onevec%*%t(onevec)
  covmat = sigma^(2)*cormat
```

```
  x=rmvnorm(nsub[i],mu[i,],covmat)
  u=cbind(x,rep(i,nsub[i]))
  Xmat=rbind(Xmat,u)
}
#N=nrow(Xmat) ## Ones column not needed for POLR
write.table(Xmat,"simdata_POLR.csv",sep=",",row.names=FALSE)
Dmat=read.table("C:\\Users\\Benard\\Desktop\\My␣Proj
\\simdata_POLR.csv", header= T, sep = ",") ## Read in the sim data
############################################################
### Skull Data set implementation
## Read in the Skull data ##
data("skulls", package = "HSAUR2");skull=skulls
skull$epoch <- gsub('c4000BC', '1', skull$epoch)
skull$epoch <- gsub('c3300BC', '2', skull$epoch)
skull$epoch <- gsub('c1850BC', '3', skull$epoch)
skull$epoch <- gsub('c200BC', '4', skull$epoch)
skull$epoch <- gsub('cAD150', '5', skull$epoch)
skull <- skull[,c("mb", "bh", "bl", "nh", "epoch")]
attach(skull);library(plyr)
## Rename the columns
Dmat=rename(skull,c("mb"="V1","bh"="V2","bl"="V3",
            "nh"="V4","epoch"="V5"))
## POLR R code For Simulated
Dmat$V5<-as.ordered(Dmat$V5)
logistic.errvec=vector(mode="numeric",length=0)
Smat1 = matrix(0,J,J)
set.seed(111)
START=runif(6,1,2)
Nrow=5 # Number of rows to pick from each category at a time
for (h in 1:6){
```

```
  nr=Nrow*(h-1)+1
  rownum=c(nr:(nr+4),(nr+30):(nr+34),(nr+60):(nr+64))
  test=Dmat[rownum,]
  train=Dmat[-rownum,]
  model <- polr(V5~ V1+V2+V3+V4,start=START,train, Hess = T)
  #summary(model)
  # Prediction with test data
  pred <- predict(model, test)
  ## Confusion Matrix for Test Data
  Conf_Mat <- table(Predicted=pred, Actual= test$V5); #Conf_Mat
  Smat1 = Smat1 + Conf_Mat
  ## Misclassification Error for the Test Data
  APER <- 1-sum(diag(Conf_Mat))/sum(Conf_Mat)
  logistic.errvec=cbind(logistic.errvec,APER)

}
Smat1
mean(logistic.errvec)

# POLR R code For Skull
#################################################################
Dmat$V5<-as.ordered(Dmat$V5)
logistic.errvec=vector(mode="numeric",length=0)
Smat2 = matrix(0,5,5)
Nrow=5 # Number of rows to pick from each category
for (h in 1:6){nr=Nrow*(h-1)+1
rownum=c(nr:(nr+4),(nr+30):(nr+34),
    (nr+60):(nr+64),(nr+90):(nr+94),(nr+120):(nr+124))
test=Dmat[rownum,]
train=Dmat[-rownum,]
```

```
model <- polr(V5~ V1+V2+V3+V4,train, Hess = T)
#summary(model)
# Prediction with test data
pred <- predict(model, test)
## Confusion Matrix for Test Data
Conf_Mat1 <- table(Predicted=pred, Actual= test$V5)
Smat2 = Smat2 + Conf_Mat1
## Misclassification Error for the Test Data
APER <- 1-sum(diag(Conf_Mat1))/sum(Conf_Mat1)
logistic.errvec=cbind(logistic.errvec,APER)
}
Smat2
mean(logistic.errvec)
##########################################################
### Iris data set implementation
data("iris") ### Pull in the iris dataset
iris$Species <- gsub('setosa', '1', iris$Species)
iris$Species <- gsub('versicolor', '2', iris$Species)
iris$Species <- gsub('virginica', '3', iris$Species)
iris$Species = as.numeric(iris$Species)
## Rename the columns
Dmat=rename(iris,c("Sepal.Length"="V1","Sepal.Width"="V2",
"Petal.Length"="V3", "Petal.Width"="V4","Species" ="V5"))
Dmat$V5<-as.ordered(Dmat$V5)
logistic.errvec=vector(mode="numeric",length=0)
Smat3 = matrix(0,J,J)
set.seed(111)
START=runif(6,1,2)
Nrow=10 # Number of rows to pick from each category at a time
for (h in 1:5){
```

```
  nr=Nrow*(h-1)+1
  rownum=c(nr:(nr+9),(nr+50):(nr+59),(nr+100):(nr+109))
  test=Dmat[rownum,]
  train=Dmat[-rownum,]
  library(MASS)
  model <- polr(V5~ V1+V2+V3+V4,train,start=START,Hess = T)
  #summary(model)
  # Prediction with test data
  pred <- predict(model, test)
  ## Confusion Matrix for Test Data
  Conf_Mat3 <- table(Predicted=pred, Actual= test$V5)
  Smat3 = Smat3 + Conf_Mat3
  ## Misclassification Error for the Test Data
  APER <- 1-sum(diag(Conf_Mat3))/sum(Conf_Mat3)
  logistic.errvec=cbind(logistic.errvec,APER)
}
Smat3
mean(logistic.errvec)
##########################################################

### Data Ellipsoid Plots #####
library(biotools); library(heplots)
## Simulated Data
sim_data=read.table("C:\\Users\\Benard\\Desktop
\\MyProj\\simdata.csv",header=T,sep = ",")
sim_data$V1 = NULL ## Remove col of ones
colnames(sim_data)<-c("X1","X2","X3","X4","Z")
simdata.boxm <- boxM(sim_data[, 1:4], sim_data[, "Z"])
## Skull Data
data("skulls", package = "HSAUR2");skull=skulls
```

```
skull$epoch <- gsub('c4000BC', '1', skull$epoch)
skull$epoch <- gsub('c3300BC', '2', skull$epoch)
skull$epoch <- gsub('c1850BC', '3', skull$epoch)
skull$epoch <- gsub('c200BC', '4', skull$epoch)
skull$epoch <- gsub('cAD150', '5', skull$epoch)
attach(skull);library(plyr)
## Rename the columns
colnames(skull)<-c("period","mb","bh","bl","nh")
## Iris Data
data(iris)
colnames(iris)<-c("Sep.L","Sep.W","Pet.L","Pet.W","Species")
#### CovEllipse Plot
covEllipses(sim_data[,1:4],sim_data$Z,fill=c(rep(FALSE,3), TRUE),
center.cex=0,pooled = F, variables=1:4,fill.alpha=.1)
covEllipses(skull[, 2:5], skull$period,pooled=FALSE,
center.cex=0,variables=1:4,fill.alpha=.1,
heplot.colors =c("red", "blue", "black", "green", "magenta"))
covEllipses(iris[,1:4], iris$Species,fill=c(rep(FALSE,3),TRUE),
center.cex=0,pooled = F,variables=1:4,fill.alpha=.1)
```

Curriculum Vitae

Benard Owusu Dechi was born in February 22, 1991. He is the fourth son of Samuel Kwame Dechi and Hanna Owusua. He graduated from Swedru Senior High School in the central region - Ghana in 2011. He then entered Kwame Nkrumah University of Science and Technology in 2012, where he obtained his Bachelors in Statistics and graduated with a first class honors. After completion of his Bachelor's degree, he worked as a Teaching Assistant from 2016 - 2011 for the Department of Mathematics, where he was actively involved in teaching Algebra and Pure Mathematics.

Benard gained admission into the Graduate School of The University of Texas at El Paso, where he pursued his Masters in Statistics and Certificate in Big Data Analytics. While pursuing his Masters, he worked as a Graduate Teaching Assistant in the school undertook an Alternate Certification Program in Teaching. Benard worked under the supervision of Dr. Naijun Sha, where he completed a thesis on Bayesian Analysis of Ordinal Outcomes through Latent Variable Approach. He was awarded the Academic and Research Excellence Outstanding Graduate Student in Statistics in spring 2019.

He intends to work as a Teacher and later on pursue his PhD in computational Science in any renowned university.

Email address: benardowusudechi@gmail.com