2016

TOPICS IN LOGISTIC REGRESSION ANALYSIS

Zhiheng Xie
University of Kentucky, zhiheng.xie@uky.edu
Digital Object Identifier: http://dx.doi.org/10.13023/ETD.2016.309

STUDENT AGREEMENT:

I represent that my thesis or dissertation and abstract are my original work. Proper attribution has been given to all outside sources. I understand that I am solely responsible for obtaining any needed copyright permissions. I have obtained needed written permission statement(s) from the owner(s) of each third-party copyrighted matter to be included in my work, allowing electronic distribution (if such use is not permitted by the fair use doctrine) which will be submitted to UKnowledge as Additional File.

I hereby grant to The University of Kentucky and its agents the irrevocable, non-exclusive, and royalty-free license to archive and make accessible my work in whole or in part in all forms of media, now or hereafter known. I agree that the document mentioned above may be made available immediately for worldwide access unless an embargo applies.

I retain all other ownership rights to the copyright of my work. I also retain the right to use in future works (such as articles or books) all or part of my work. I understand that I am free to register the copyright to my work.

REVIEW, APPROVAL AND ACCEPTANCE

The document mentioned above has been reviewed and accepted by the student's advisor, on behalf of the advisory committee, and by the Director of Graduate Studies (DGS), on behalf of the program; we verify that this is the final, approved version of the student's thesis including all changes required by the advisory committee. The undersigned agree to abide by the statements above.

Zhiheng Xie, Student

Dr. Richard Kryscio, Major Professor

Dr. Constance Wood, Director of Graduate Studies

TOPICS IN LOGISTIC REGRESSION ANALYSIS

DISSERTATION

A dissertation submitted in partial fulfillment of
the requirements for the degree of Doctor of
Philosophy in the College of Arts and Sciences
at the University of Kentucky

By
Zhiheng Xie
Lexington, Kentucky
Director: Dr.Richard Kryscio, Professor of Statistics
Lexington, Kentucky

2016

ABSTRACT OF DISSERTATION

TOPICS IN LOGISTIC REGRESSION ANALYSIS

Discrete-time Markov chains have been used to analyze the transition of subjects from intact cognition to dementia with mild cognitive impairment and global impairment as intervening transient states, and death as competing risk. A multinomial logistic regression model is used to estimate the probability distribution in each row of the one step transition matrix that correspond to the transient states. We investigate some goodness of fit tests for a multinomial distribution with covariates to assess the fit of this model to the data. We propose a modified chi-square test statistic and a score test statistic for the multinomial assumption in each row of the transition probability matrix.

Multinomial logistic regression with categorical covariates can be analyzed by contingency tables. Exact p-value of goodness of fit test can be calculated based on MCMC samples. We show a hybrid scheme of the sequential importance sampling (SIS) procedure and the MCMC procedure for two-way contingency tables. We apply the SIS-MCMC procedure to the Nun Study data, a cohort of 461 participants on aging disease. The presence of the APOE-4 allele, levels of education are included as covariates in the application. Different grouping methods on age are also discussed. Separating data into four groups based on quantiles of age is recommended in the Nun Study.

The traditional logistic regression model restricts the analysis on observations with complete covariate data, and ignores the incomplete observations due to missing or censored covariates. However, much information is lost in this approach. We introduce a maximum likelihood estimation based on the joint distribution of binary response variable, complete covariate and a right censored covariate. Simulation results show that the estimates with the new method are more accurate than those with the traditional complete case method when the sample size is relatively small or medium, across different censoring pattern. The proposed method is also applied to a model to analyze the relationship between the presence of arteriolosclerosis and the stay time in mild cognitive impairment of patients from SMART Study.

KEYWORDS: Multinomial logistic regression, Goodness-of-fit test, Sequential Importance Sampling, Nun Study

Author's signature: Zhiheng Xie

Date: July 15, 2016

TOPICS IN LOGISTIC REGRESSION ANALYSIS

By
Zhiheng Xie

Director of Dissertation: Dr.Richard Kryscio

Director of Graduate Studies: Dr. Constance Wood

Date: July 15, 2016

ACKNOWLEDGMENTS

I will express my deepest gratitude to my advisor, Dr. Richard Kryscio. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own, and at the same time the guidance to recover when my steps faltered. His patience and support helped me overcome many difficulties and finish this dissertation.

Next, I wish to thank all my complete Dissertation Committee: Dr. Ruriko Yoshida, Dr. William Griffith, Dr. David Fardo, Dr. Mai Zhou and Dr. Yanbing Zheng. Each individual provided insights that guided and challenged my thinking, improving the my research. I am deeply grateful to Dr. Yoshida. Thank you for your brilliant ideas and kindly guidance that helped me a lot to finish this dissertation. I am also grateful to Dr. Fardo for his helpful guidance and encouragement during my research.

Finally, I would like to thank my mom, none of this would have been possible without the love and patience of my family. I would especially like to thank my fiancée, Qingcong Yuan, for her on-going support and love throughout my study process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

1.1 Multinomial Logistic Regression Model

Assume Y_i is an outcome variable for the ith observation, which can take $c+1$ possible values denoted by $(0, 1, 2, ..., c)$, with corresponding probability $\boldsymbol{\pi_i}$ when

$$\boldsymbol{\pi_i} = (\pi_{i0}, \pi_{i1}, \cdots, \pi_{ic}). \tag{1.1}$$

Let $\boldsymbol{x_i}$ be the independent predictor variable or covariate vector for ith observation, $\boldsymbol{x_i} = (x_1, x_2, ..., x_p)'$. Under multinomial logistic regression structure with $Y_i = 0$ as the reference category, the model is:

$$log\frac{\pi_{ij}}{\pi_{i0}} = \boldsymbol{x_i'\beta_j} \tag{1.2}$$

for any $j \neq 0$ and the coefficient vector $\boldsymbol{\beta_j} = (\beta_{j1}, \beta_{j2}, \cdots, \beta_{jp})'$ where $j = 1, 2, ..., c$. To calculate the probabilities, we have:

$$\pi_{ij} = \frac{\exp(\boldsymbol{x_i'\beta_j})}{1 + \sum_{k\neq 0}\exp(\boldsymbol{x_i'\beta_k})} \tag{1.3}$$

for the non-reference categories $j \neq 0$ while for the reference category probability is

$$\pi_{i0} = \frac{1}{1 + \sum_{k\neq 0}\exp(\boldsymbol{x_i'\beta_k})}. \tag{1.4}$$

We can also define y_{ij} to be the indicator of j if the outcome of the ith observation is j or not, which means $y_{ij} = 1$ if $Y_i = j$ and $y_{ij} = 0$ otherwise.

1.2 Grouped Pearson's Chi-square Test statistics

In general, a goodness-of-fit test compares the observed binary variable y_{ij} with the estimated probability $\hat{\pi}_{ij}$. A convenient way is to show the observations in a contingency table with n rows and c columns, where n is the sample size of the dataset. The observed frequency in cell (i, j) is denoted by $\hat{y}_{ij}$ and the estimated probability is denoted by $\hat{\pi}_{ij}$. Then, based on that table, the Pearson chi-square test statistic can be calculated as:

$$X^2 = \sum_{i=1}^{n} \sum_{j=0}^{c} \frac{(\hat{y}_{ij} - \hat{\pi}_{ij})^2}{\hat{\pi}_{ij}} \tag{1.5}$$

When $c = 1$, this expression reduces to the test statistic for a binomial logistic regression model. In the binomial logistic scenario, the p-value can be calculated using the χ^2 distribution with $n - 1$ degrees of freedom(Fagerland et al (2008)). The Pearson's chi-square test is used generally for the goodness-of-fit test for the binomial logistic regression model originally, but we can extend it to the multinomial regression model.

An important assumption for the Pearson's chi-square test is that the expected cell counts should not be too small. That assumption is legitimate when there are a few discrete covariates. However, when we have more covariates or a continuous covariate is included in the model, this assumption is clearly violated.

Hosmer and Lemeshow (1980) proposed an extension of the Pearson's chi-square test for logistic regression model with continuous covariates to avoid the violation of that assumption of traditional Pearson's chi-square test based on contingency tables. Their method groups the observations based on the estimated probabilities. For the $n \times c$ contingency table, instead of considering each observation as a row, they set the rows into a fixed number, g, so that the expected cell counts increase as n increases.

One possible problem is the grouping strategy, which is clear for the binomial logistic model, but less clear for the multinomial setting due to numerous grouping

strategies. Fagerland et al (2008) suggested using the 'deciles' of risk formed from the reference group, $\hat{\pi}_{i0}$. For g groups, group 1 will contain n/g observation with the lowest estimated probability in the reference group. The quantity n/g may not be a integer value, and all groups may not have the same number of observations. When we have continuous covariates in model, the tied estimated probability will be rare and small imbalance in the group size will not affect the value of statistic greatly.

Let O_{kj} and E_{kj} denote the observed frequencies and estimated probabilities in kth group and jth outcome, where k in $1, 2, \cdots, g$ and j in $0, 1, \cdots, c$.

$$O_{kj} = \sum_{\text{l in group k}} \hat{y}_{lj} \tag{1.6}$$

$$E_{kj} = \sum_{\text{l in group k}} \hat{\pi}_{lj}. \tag{1.7}$$

The Grouped Pearson's chi-square statistic is:

$$C = \sum_{k=1}^{g} \sum_{j=0}^{c} \frac{(O_{kj} - E_{kj})^2}{E_{kj}}. \tag{1.8}$$

Fagerland et al (2008) suggested that the statistic C under the null hypothesis has an approximate χ^2 distribution with degree of freedom $(g-2) \times (c-1)$.

1.3 Likelihood Ratio Test Statistic for Goodness-of-fit Test

To test the goodness-of-fit for multinomial logistic regression model, we can also use a likelihood ratio test. The likelihood ratio test statistic is two times of the difference between the likelihood of logistic regression model and saturated model.

The log-likelihood of multinomial logistic regression model is

$$
\begin{aligned}
l_1 &= \log \prod_{i=1}^{n} [\prod_{j=0}^{c} \pi_{ij}^{y_{ij}}] && (1.9)\\
&= \sum_{i=1}^{n} \{ \sum_{j=0}^{c-1} y_{ij} \log \pi_{ij} + (1 - \sum_{j=0}^{c-1} y_{ij}) \log[1 - \sum_{j=0}^{c-1} \pi_{ij}] \} && (1.10)\\
&= \sum_{i=1}^{n} \{ \sum_{j=0}^{c-1} y_{ij} (\beta_{j0} + \boldsymbol{x_i'\beta_j}) - \log[1 + \sum_{j=0}^{c-1} \exp(\beta_{j0} + \boldsymbol{x_i'\beta_j})] \} && (1.11)\\
&= \sum_{j=0}^{c-1} [\beta_{j0} (\sum_{i=1}^{n} y_{ij}) + \sum_{k=1}^{P} \beta_{jk} (\sum_{i=1}^{n} x_{ik} y_{ij})] && (1.12)\\
&- \sum_{i=1}^{n} \log[1 + \sum_{j=0}^{c-1} \exp(\beta_{j0} + \boldsymbol{x_i'\beta_j})]
\end{aligned}
$$

For the saturated model, the estimated probability is

$$\hat{\pi}_{ij} = \frac{\sum_{i=1}^{n} y_{ij}}{n} \tag{1.13}$$

and the corresponding log-likelihood is

$$
\begin{aligned}
l_2 &= \log \prod_{i=1}^{n} [\prod_{j=0}^{c} \hat{\pi}_{ij}^{y_{ij}}] && (1.14)\\
&= \sum_{i=1}^{n} \{ \sum_{j=0}^{c-1} y_{ij} \log \hat{\pi}_{ij} + (1 - \sum_{j=0}^{c-1} y_{ij}) \log[1 - \sum_{j=0}^{c-1} \hat{\pi}_{ij}] \} && (1.15)\\
&= \sum_{i=1}^{n} \{ \sum_{j=0}^{c-1} y_{ij} \log \frac{\sum_{i=1}^{n} y_{ij}}{n} && (1.16)\\
&+ (1 - \sum_{j=0}^{c-1} y_{ij}) \log[1 - \sum_{j=0}^{c-1} \frac{\sum_{i=1}^{n} y_{ij}}{n}] \}
\end{aligned}
$$

Then we have the likelihood ratio test statistic is

$$-2(l_1 - l_2) \sim \chi^2_{nc-p}$$

where n is the number of combinations of different values of covariates, $c+1$ is the

number of possible response values and p is the number of covariates.

1.4 Sequential Importance Sampling (SIS)

For a K-choice multinomial logistic regression model with 2 discrete covariates, the dataset can be described as a $K \times I \times J$ contingency tables (X), as shown in Table 1.1. In this table, each element X_{ijk} is the count of observations where the response variable equals k, the first covariate equals i, and the second covariate equals j. If we add all cell counts in X, then it becomes the sample size of a given data.

For a contingency tables X, from Hara et al. (2010), the sufficient statistics for parameters in a multinomial logistic regression model are:

$$X_{k++}, \quad \sum_{i=1}^{I} iX_{ki+}, \quad \sum_{j=1}^{J} jX_{k+j}, \quad X_{+ij}, \tag{1.17}$$

where $i = 1, 2, ..., I, j = 1, 2, ..., J, k = 1, 2, ..., K-1$.

In this section, we will introduce an algorithm to sample $K \times I \times J$ contingency tables with the same sufficient statistics for the multinomial logistic regression model by the sequential importance sampling (SIS).

Generally, an importance sampling method is a statistical technique to get a sample from a targeted distribution by using a sample from the proposal distribution. A proposal distribution can be any distribution that is easy to implement. Assume that F_T is the set of all $K \times I \times J$ contingency tables (X) satisfying marginal conditions (for example, the sufficient statistics shown in (1.17)). Let $p(\boldsymbol{n})$, for any $\boldsymbol{n}$ in F_T, be the uniform distribution over F_T, then $p(\boldsymbol{n}) = 1/|F_T|$. Let $q()$ be a proposal

Table 1.1: $K \times I \times J$ contingency tables for the K-choice multinomial logistic regression model with 2 discrete covariates.

Choice	Two-way table		
1	X_{111}	$\dots$	X_{1J1}
	$\vdots$	$\ddots$	$\vdots$
	X_{I11}	$\dots$	X_{IJ1}
2	X_{112}	$\dots$	X_{1J2}
	$\vdots$	$\ddots$	$\vdots$
	X_{I12}	$\dots$	X_{IJ2}
$\dots$	$\dots$	$\dots$	$\dots$
K	X_{11K}	$\dots$	X_{1JK}
	$\vdots$	$\ddots$	$\vdots$
	X_{I1K}	$\dots$	X_{IJK}

distribution such that $q(\boldsymbol{n}) > 0$ for all $\boldsymbol{n}$ in F_T. Then we have:

$$E_q[\frac{1}{q(\boldsymbol{n})}] = \sum_{\boldsymbol{n} \in F_T} \frac{1}{q(\boldsymbol{n})} q(\boldsymbol{n}) = |F_T|, \tag{1.18}$$

and we can estimate the count of F_T as:

$$|\hat{F_T}| = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{q(\boldsymbol{n_i})} \tag{1.19}$$

from N iid tables sampled from $q(\boldsymbol{n})$. This is an example of how we use ab importance sampling method to estimate a property of a target space F_T.

An important problem in an importance sampling is constructing a good proposal distribution $q()$, as the target space F_T can be complicated. Chen et al. (2005a) noticed that if we vectorize the table $\boldsymbol{n} = (\boldsymbol{n_1}, \boldsymbol{n_2}, ..., \boldsymbol{n_J})$, then by the multiplication rule, we have:

$$q(\boldsymbol{n}) = q(\boldsymbol{n_1}) q(\boldsymbol{n_2}|\boldsymbol{n_1}) \cdots q(\boldsymbol{n_J}|\boldsymbol{n_{J-1}}, ..., \boldsymbol{n_1}). \tag{1.20}$$

This factorization suggests that we can generate the table sequentially, a column by a column. This recursive property gives rise to the name *Sequential Importance Sampling (SIS)*. Chen et al. (2005b) noticed that one can sample a cell count from the

interval at each step to produce a table satisfied with the marginal constraints. Here, we are using Integer Programming (IP) to obtain the tight bounds sequentially. By this method, we can generate a $K \times I \times J$ contingency table with the same marginal constraint.

Sequential importance sampling (SIS) is an importance sampler with a proposal distribution constructed iteratively via conditional univariate distributions. It proceeds by simply sampling cell entries of a contingency table sequentially such that the final distribution approximates the target distribution. It was first applied to two-way contingency tables in Chen et al. (2005a). The SIS procedure overcomes some disadvantages in the traditional Monte Carlo Markov Chain (MCMC) procedure. Compared to the MCMC method, the SIS procedure does not need expensive pre-computation. Also, the SIS procedure is guaranteed to sample a table from the distribution independently, whereas the MCMC approach needs a long time to run a chain to satisfy the independent condition. Typically, an interval based on the support of the marginal distribution is calculated through Integer Programming (IP), Linear Programming (LP), or Shuttle Algorithm (Dobra and Fienberg (2010)). Under the independence model, the SIS procedure will always produce tables with the marginal constraints (Chen et al. (2005c)), i.e. the SIS procedure does not reject a table. However, in general, the SIS procedure might reject a sampled table. Chen et al. (2006a) showed an algorithm to check the necessary condition of a given model, so that the SIS procedure does not reject a table under the model.

To apply the SIS procedure to a $K \times I \times J$ contingency table in the multinomial logistic regression model, the first step is to construct a design matrix, named as $\Lambda(A \otimes B)$, which comes from the linear constrains of the sufficient statistics for the set of contingency tables as described in (1.17).

Assume two matrices $A_0 = (\boldsymbol{a_1}, ..., \boldsymbol{a_I})$ and $B_0 = (\boldsymbol{b_1}, ..., \boldsymbol{b_J})$, where $\boldsymbol{a_i}$ and $\boldsymbol{b_j}$ are column vectors. The configuration $A_0 \otimes B_0$, i.e. the Segre product of A_0 and B_0 is

defined as:

$$A_0 \otimes B_0 = (\boldsymbol{a_i} \oplus \boldsymbol{b_j}, \quad i = 1, \ldots, I, j = 1, \ldots, J), \tag{1.21}$$

$$\boldsymbol{a_i} \oplus \boldsymbol{b_j} = \begin{bmatrix} \boldsymbol{a_i} \\ \boldsymbol{b_j} \end{bmatrix}. \tag{1.22}$$

The Lawrence Lifting of a matrix Z with I columns is defined as:

$$\Lambda(Z) = \begin{bmatrix} Z & 0 \\ E_I & E_I \end{bmatrix}, \tag{1.23}$$

where E_I is a $I \times I$ identity matrix.

Now we consider two matrices A and B as:

$$A = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & 2 & 3 & \ldots & I \end{bmatrix} \tag{1.24}$$

$$B = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & 2 & 3 & \ldots & J \end{bmatrix}. \tag{1.25}$$

The design matrix for the a bivariate regression model can be constructed as $\Lambda(A \otimes B)$:

$$\Lambda(A \otimes B) = \begin{bmatrix} A \otimes B & 0 & \ldots & 0 & 0 \\ 0 & A \otimes B & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & A \otimes B & 0 \\ E_{IJ} & E_{IJ} & \ldots & E_{IJ} & E_{IJ} \end{bmatrix}, \tag{1.26}$$

which is a $[4(K-1) + IJ] \times IJK$ matrix. Note that there are K $A \otimes B$s in the diagonal of the matrix.

If X is a $K \times I \times J$ contingency table for a bivariate logistic regression model, and

$\boldsymbol{x}$ is a IJK-dimensional vector by stacking each column of in X, then we have

$$\Lambda(A \otimes B)\boldsymbol{x} = \boldsymbol{b}, \tag{1.27}$$

where

$$\boldsymbol{b} = (X_{1++}, \sum_{i=1}^{I} iX_{1i+}, X_{1++}, \sum_{j=1}^{J} jX_{1+j}, ..., X_{K++}, \sum_{i=1}^{I} iX_{Ki+}, X_{K++}, \sum_{j=1}^{J} jX_{K+j}, X_{+11}, X_{+12}, ..., X_{+1J}, X_{+21}, ..., X_{+IJ}), \tag{1.28}$$

which is a $(4(K-1)+IJ)$ dimensional vector. Also $\boldsymbol{b}$ is the sufficient statistics of the K-choice bivariate multinomial logistic regression model as described in Equation (1.17).

If a table X_c and its corresponding vector $\boldsymbol{x_c}$ has the property

$$\Lambda(A \otimes B)\boldsymbol{x_c} = \Lambda(A \otimes B)\boldsymbol{x} = \boldsymbol{b}, \tag{1.29}$$

then, X_c is a contingency table that has the same sufficient statistics with X. The following SIS algorithm can generate tables that are independent of the original data X while maintaining the sufficient statistics.

Below is the algorithm for applying the SIS to a K-choice multinomial logistic regression model with two discrete covariates.

1. Assume a multinomial logistic regression model with K choices in response variable, two discrete covariates are included, one with I levels and the other with J levels. In this circumstance, each dataset can constitute a $K \times I \times J$ tables X. X can also be transformed to a IJK-dimensional vector ($\boldsymbol{x}$) by stacking each column of itself.

2. Under above assumptions, construct a $d_1 \times d_2$ design matrix $\Lambda(A \otimes B)$ as de-

scribed in Equation (1.26), and calculate the d_1-dimensional marginal vector $\boldsymbol{b}$, which is also the sufficient statistics in the model. We have the relationship

$$\Lambda(A \otimes B)\boldsymbol{x} = \boldsymbol{b}, \tag{1.30}$$

$$d_1 = 4(K-1) + IJ \qquad d_2 = IJK.$$

3. With observed vector $\boldsymbol{x}$ and the index i from 1 to IJ. Set a vector $\boldsymbol{c} = \boldsymbol{0}$ and the ith element $c_i = 1$. Run linear programming function (package lpsolve in R Berkelaar et al. (2004)) with $\Lambda(A \otimes B), \boldsymbol{b}, \boldsymbol{c}$ for

$$min\ \boldsymbol{c} \cdot \boldsymbol{x} \quad \text{such that} \quad \Lambda(A \otimes B)\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq 0.$$

Return $L = \boldsymbol{c} \cdot x^*$ where x^* is the output of lpsolve. L is the lower bound for the ith element $\boldsymbol{x}_i$.

4. Rerun the linear programming function for

$$max\ \boldsymbol{c} \cdot \boldsymbol{x} \quad \text{such that} \quad \Lambda(A \otimes B)\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \geq 0.$$

Return $U = \boldsymbol{c} \cdot x^*$ where x^* is the output of lpsolve. U is the upper bound for the ith element $\boldsymbol{x}_i$.

5. Sample an integer x_c uniformly from L and U as the ith element in $\boldsymbol{x}$.

6. Let A_1 be the first column of $\Lambda(A \otimes B)$ and let A' be the $d_1 \times (d_2 - 1)$ matrix after removing the first column from $\Lambda(A \otimes B)$. Set

$$\boldsymbol{b'} = \boldsymbol{b} - x_c \cdot A_1. \tag{1.31}$$

7. Using the updated A' and $\boldsymbol{b'}$ to repeat steps 3 to 6, until the completion of a new sampled point $\boldsymbol{x}$.

It should be noted that with the above algorithm, it is possible that the lower bound or upper bound might not give an interval at some step. In these cases, we can just run the algorithm from the very beginning again.

1.5 MCMC

For two-way and multiway contingency tables, a MCMC sampling method has a wide range of applications, such as computing the exact p-values of goodness-of-fit tests and estimating the number of contingency tables with certain margins (Besag and Clifford (1989)). In order to apply the MCMC approach to contingency tables, all tables must be connected via a Markov basis. A Markov basis is a set of moves to connect all contingency tables via Markov chain (Diaconis and Sturmfels (1998)). When a Markov basis is known and fairly small, the MCMC method is not memory intensive and easy to implement. However, for three-way contingency tables with fixed margins, the number of elements in a Markov basis can be arbitrary (De Loera and Onn (2005)). Also, to sample a table independently from the distribution and satisfy the independence assumption, a Markov Chain may take a long time to converge.

Here, we use a binomial logistic regression model as an example of the MCMC algorithm. In this part, we use Metropolis Hastings algorithm to obtain a sample of two-way contingency tables based on bivariate logistic regression model. For Metropolis Hastings algorithm, we add a move to the previous table to get the proposal table. A $K \times I \times J$ contingency table can be described as K layers of $I \times J$ tables.

For a binomial logistic regression, we want to know a Markov basis of the contingency tables. In general, it is very complicated. However, when the marginal of contingency tables are fixed and positive, Chen et al. (2006b) showed that a simple subset of Markov basis can contain the connectivity of all sets of two-way contingency tables with a fixed positive marginal.

Let $\boldsymbol{e_{ijk}}$ be an integer array with 1 at the cell (1jk), -1 at cell (2jk) and 0 elsewhere.

Define $B_{\Lambda(A\otimes B)}$ as a set of moves $z = (z_{ijk})$ with the conditions,

$$z = \boldsymbol{e_{j_1k_1}} - \boldsymbol{e_{j_2k_2}} - \boldsymbol{e_{j_3k_3}} + \boldsymbol{e_{j_4k_4}} \tag{1.32}$$

$$(j_1, k_1) - (j_2, k_2) = (j_3, k_3) - (j_4, k_4). \tag{1.33}$$

Hara et al. (2010) proved that $B_{\Lambda(A\otimes B)}$ connect every two-way contingency table satisfying $X_{+jk} > 0$.

Below is the MCMC (Metropolis-Hastings) algorithm on a set of contingency tables:

1. Set sample S as empty set, starting point as $\boldsymbol{x_0}$.
2. Compute a Markov sub-basis $B_{\Lambda(A\otimes B)}$ for two-way contingency tables as described in (1.32).
3. Pick a move $\boldsymbol{z}$ from $B_{\Lambda(A\otimes B)}$ uniformly.
4. Calculate a candidate table $\boldsymbol{x_c} = \boldsymbol{x_{i-1}} + \boldsymbol{z}$.
5. If $\boldsymbol{x_c} \geq 0$, compute the acceptance ratio

$$r = \frac{Pr(\boldsymbol{x_c}|\boldsymbol{m})}{Pr(\boldsymbol{x_{i-1}}|\boldsymbol{m})}. \tag{1.34}$$

   where $\boldsymbol{m}$ is the set of marginals. For a two-way contingency table,

$$r = \frac{\Pi_{\text{all cell counts k in } \boldsymbol{x_{i-1}}} k!}{\Pi_{\text{all cell counts j in } \boldsymbol{x_c}} j!}. \tag{1.35}$$

   With probability $\min(r, 1)$ and $\boldsymbol{x_c} \geq 0$, accept $\boldsymbol{x_i} = \boldsymbol{x_c}$. If the candidate is rejected, $\boldsymbol{x_i} = \boldsymbol{x_{i-1}}$.
6. If $\boldsymbol{x_c} < 0$, $\boldsymbol{x_i} = \boldsymbol{x_{i-1}}$.
7. Repeat steps 2 to 6 n times.
8. return sample S.

1.6 Nun Study Data

The Nun Study began in 1991. All participants from the School Sisters of Notre Dame born before 1917 and living in communities in the mid-western, eastern, and southern United States were recruited to the cohort during 1991-1993. 672 participants agreed to join the cohort out of 1031 eligible Catholic sisters aged 75 years or elder (A Mortimer (2012)). Each participant agreed to share their collection of medical and archival records, undergo annual physical and cognitive examinations, and brain donation after death. There is no significant difference between participants and nonparticipants in age, race, or mortality rate. Follow-up assessments took place with unequally spaced periods varying from 0.421 to 3.911 years in a span of 15 years.

The cognitive status of each participant at each assessment was summarized as: 1 = intact cognition, 2 = mild cognitive impairments (M.C.I.), 3 = global impairment (G.I.), 4 = dementia, and 5 = death. In those five states, dementia and death are treated as absorbing states, and the other three are treated as transient states. The starting status for one participant could be any one of the 3 transient states. Backward transitions between transient states are allowed. The risk factors of interest in the Nun Study include presence of APOE-4 allele (APOE4, binary variable), education level (no college, college degree and post graduate degree) and age (continuous variable). Figure 1.1 shows the multi-state structure and possible transitions in the Nun Study data.

Several models have been applied in Nun Study, including Markov chain model (Tyas et al (2007)) and Semi-Markov model (Wei and Kryscio(2014)). In both above

models, we can construct a one-step transition probability matrix as:

$$\begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} & P_{15} \\ P_{21} & P_{22} & P_{23} & P_{24} & P_{25} \\ P_{31} & P_{32} & P_{33} & P_{34} & P_{35} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{1.36}$$

Both the Markov model and Semi-Markov model treat probabilities within each row of the transition matrix as outcomes of a multinomial regression model. For example, Salazar et al.(2007) formed a multinomial logistic parameterization that linked these transition probabilities to the vector of covariates as follows:

$$log(\frac{P_{sv}}{P_{s1}}) = \alpha_v + \boldsymbol{X'\beta_v} \tag{1.37}$$

where $s = 1, 2, 3$ and $v = 1, 2, 3, 4, 5$

1.7 SMART Data

The Statistical Modeling of Aging and Risk of Transition study (SMART) is an aggregation of 11 different high-quality longitudinal cohorts of elder adults with high autopsy rates. It enrolled 11,541 participants, of which 3,001 died and came to autopsy (Abner et al. (2015)). SMART is an important resource for the field of mixed dementia epidemiology and neuropathology. In SMART, participants were primarily cognitively intact at baseline and were subsequently assessed for transition to mild cognitive impairment (MCI) and dementia over years of follow-up. We are interested in those participants who died while in the MCI state, since they had neither normal cognition at time of death nor were they demented.

Cerebrovascular disease affecting the small arteries and arterioles of the brain is often seen in brains of persons with dementia (Esiri et al. (1997), Pantoni et al.

(1996)). Arteriolosclerosis is a form of vascular disease that associated with vessel wall thickening and luminal narrowing that may cause downstream ischemic injury (Kumar et al. (2012)). Risk factors and cognitive sequelae of brain arteriolosclerosis pathology are not fully understood. Ighodaro et al. (2016) provide results to show brain arteriolosclerosis is associated with altered cognitive status. Of specific interest in this dissertation (chapter 4) is the relationship between time spent in the MCI state and the presence of arteriolosclerosis in the brain upon autopsy. Since some of the participants were in MCI at baseline, this time variable is subject to right censoring.

1.8 Outline of Dissertation

The remainder of this dissertation is organized as follows:

In Chapter 2, we introduce the grouped chi-square test for a goodness-of-fit test of the multinomial logistic regression model. We show the traditional chi-square test inflates the type I error due to a clustering effect within each subject. We modify the traditional chi-square statistic and show our new test statistic will preserve type I error better with similar power when the alternative model has a squared term of the covariate. We apply our new test to a multinomial logistic regression model to estimate the transition probability matrix in Nun Study data.

In Chapter 3, we introduce a hybrid method of sequential importance sampling and MCMC based on subset of Markov bases to sample two-way contingency tables for multinomial logistic regression model with two categorical covariates. This new method combines the advantages of both methods. We apply the new sampling method to Nun Study data with discussion of different grouping methods on age.

In Chapter 4, we propose a maximum likelihood estimate based on joint probability to deal with logistic regression model with censored covariates. Simulation results show that the new method estimates the coefficients better than traditional methods

based on complete cases for small or medium sized datasets. We also apply the new method to a study of the relationship between the presence of arteriolosclerosis and time in mild cognitive impairment which is sometimes right censored.

In Chapter 5, we introduce some potential future work based on each chapter's model.

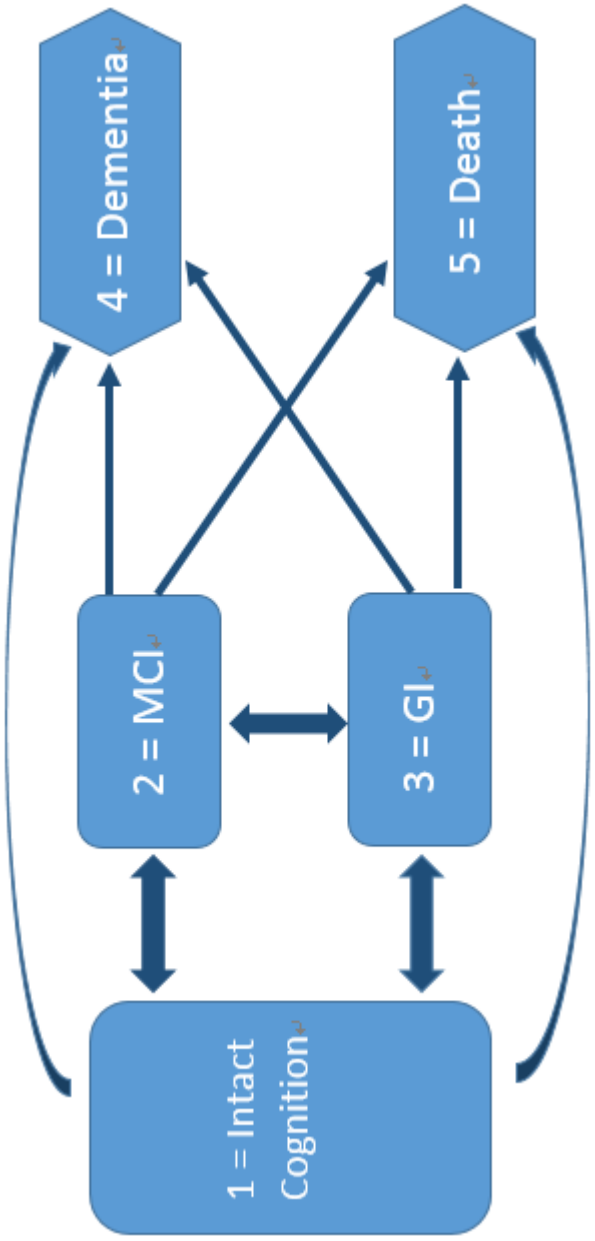

Figure 1.1: Multi-state structure in Nun Study

Chapter 2 Goodness-of-fit Test for Multinomial Logistic Regression Model with Nun Study Data

2.1 Introduction

The Markov model and Semi-Markov model are widely used in multi-state data in clinical trials and observational studies. For example, Salazar et al. (2007) and Wei et al. (2014) applied Markov Chain model while Wei and Kryscio (2014) applied Semi-Markov model ito the Nun Study data. Both models assume a multinomial logistic regression model to calculate the one-step transition probabilities. It is important to have a goodness-of-fit test to verify the validity of these models. An important feature for the Nun Study data is that the clustering effect is strong within subjects. To our knowledge, there is not a test that can deal with the clustering effect in multinomial regression model. Our motivation for the goodness-of-fit test here is to test the appropriateness of calculating the one-step transition probabilities from each state by the multinomial logistic regression model in the Nun Study data.

Most of the goodness-of-fit tests for the logistic regression are designed for a binary outcome. Some of these are widely used. Hosmer and Lemeshow (1980,1989) proposed an extension to Pearson's chi-square test using a grouping method based on estimated probabilities. Another test based on smoothed residuals was proposed by Cessie and van Houwelingen (1991); this test has a clear alternative that residuals of samples close in covariates space tend to in the same direction. It also can be structured as a score test.

The goodness-of-fit test for a multinomial logistic regression model is less developed. As the multinomial logistic model can be considered a generalization of the binomial logistic regression with multiple possible outcomes, many authors extended their test statistic from the binary case. Hosmer and Lemeshow (2000) suggest first

using a series of individual binomial logistic model tests, such as Hosmer-Lemeshow statistic, and then integrating the results. This method is easy to calculate but need a further consideration. Bull (1994) and Fagerland et al. (2008) extend the Hosmer-Lemeshow statistic to the multinomial case, and provide a type I error analysis using simulations. Pigeon (1999) made an improvement on those tests by modification to deal with an underdispersion problem when estimated probability in each cell are largely different. Goeman and Cessie (2006) proposed a smoothed residuals test statistic.

All the above tests have a simple random sampling design assumption and cannot handle data from complex survey designs. However, complex survey designs are common in many areas, especially in clinical trials with longitudinal data, which has non-zero correlation between observations. When a clustering effect is present, the chi-square statistic uses groups based on estimated probabilities, which may inflate the type I error (Rao and Scott (1992)), and the modified statistic can control it better. Rao and Scott (1979,1981) proposed an adjustment based on design-effect matrix to deal with the complex design survey. We followed this spirit to propose a new statistic that can deal with the underdispersion and cluster effect together, and apply it to the Nun Study data.

2.2 Method

Modification for Underdispersion of Chi-square Test

The multinomial logistic model and corresponding grouped chi-square goodness-of-fit test are introduced in chapter 1. We can also construct a contingency table for the grouped Pearson's chi-square statistic, which include g rows for each group and $c+1$ columns for each value of outcome. The Pearson's chi-square test statistic X^2 assumed the estimated probability in each cell are the same for all observations. In other words, the estimated probability for each observation in the same group and

same outcome should not be largely different. It can be violated in multinomial regression models. Pigeon and Heyse (1999) proposed a modification for the Pearson's chi-square test statistic for both binomial and multinomial settings.

Since

$$\sum_{\text{l in group k}} \hat{\pi}_{lj}(1-\hat{\pi}_{lj}) \leq n_{kj}\bar{\pi}_{lj}(1-\bar{\pi}_{lj}) \tag{2.1}$$

which means the estimated variance of O_{kj} is less than the variance of O_{kj} assuming the observations in each cell have same expected probabilities, and here

$$\bar{\pi}_{lj} = \sum_{\text{l in group k}} \hat{\pi}_{lj}/n_k \tag{2.2}$$

Then in classical Pearson's chi-square statistic, the O_{kj} are underdispersed relative to a multinomial situation where each subject has the same value of $\bar{\pi}_{lj}$ in cell (k, j). To correct the underdispersion, Pigeon and Heyse (1999) proposed a modification by adjusting the J statistic using

$$J = \sum_{k=1}^{g}\sum_{j=0}^{c} \frac{(O_{kj}-E_{kj})^2}{\phi_{kj}E_{kj}} \tag{2.3}$$

where

$$\phi_{kj} = \frac{\sum_{\text{l in group k}} \hat{\pi}_{lj}(1-\hat{\pi}_{lj})}{n_{kj}\bar{\pi}_{lj}(1-\bar{\pi}_{lj})} \tag{2.4}$$

The modification parameter ϕ_{kj} is the ratio of real variance of O_{kj} to variance with the same expected probabilities within the same cell. Also we can show that

$$\phi_{kj} = 1-(n-1)S^2_{kj}/n_{kj}\bar{\pi}_{lj}(1-\bar{\pi}_{lj}) \tag{2.5}$$

where

$$S^2_{kj} = \sum_{\text{l in group k}} (\hat{\pi}_{lj}-\bar{\pi}_{lj})^2/(n_{kj}-1) \tag{2.6}$$

When all $\hat{\pi}_{lj} = \bar{\pi}_{lj}$, $S^2_{kj} = 0$ and $J = C$. In a more general case, $\phi_{kj} < 1$ and $J > C$.

We also noticed that Hosmer-Lemeshow statistic C can alleviate this violation by grouping based on the decile of the estimated probabilities, which makes the $\hat{\pi}_{ij}$ closer to each other in the same cell. The modification of J is suggested based on the simulation results below.

Modification for Clustering Effect

An important assumption in both Hosmer-Lemeshow test statistic C and Pigeon's modification J is that all observations are independent of each other. This assumptions can be violated in some data when the observations are not from a simple random sampling survey. For example, in some medical data, several observations may come from the same patient since it is a longitudinal study.

Rao and Scott (1979,1981) showed the effects that complex sampling procedures, such as clusters, have on the use of standard Pearson chi-square test. Their study showed the type I error would be inflated if the standard method is used regardless of clustering effects.

Rao and Scott (1979,1981) also proved that under a complex design, standard Pearson's chi-square statistic X^2 with I cells is distributed asymptotically as a weighted sum

$$\delta_1 W_1 + \delta_2 W_2 + \cdots + \delta_{I-1} W_{I-1} \tag{2.7}$$

where $W_i \sim \chi^2_1$ and δ_i are the eigenvalues of the design effect matrix $P_0^{-1}V$. P_0 is the multinomial covariance matrix, and V is the covariance matrix of the actual design.

Under the circumstance of Pearson's chi-square statistic for K cells, they proposed a first-order correction for the standard X^2 statistic:

$$\frac{X^2}{\delta_\cdot} \sim \chi^2_{df} \tag{2.8}$$

where

$$\hat{\delta}_{\cdot} = \frac{n}{K-1} \sum_{i=1}^{K} \frac{\hat{v}_i}{p_{0i}} \tag{2.9}$$

$\hat{v}_i$ is the estimated variance of p_i. p_i is the probability of success in cell i and p_{0i} is that probability under null hypothesis.

Although we have some information about the design effect matrix, in most cases, we only have the variance of the actual design, which is the variance of each cell in Pearson's type test statistic. The good thing about Rao-Scott first-order corrections is that this correction does not require full information about the design effect matrix. But instead, we only need the diagonal of that matrix.

We can calculate $\hat{v}_i$ using the results from Rao and Scott (1992) in the multinomial regression model with clustering effect:

$$\hat{v}_i = \frac{m_i}{(m_i-1)n_i^2} \sum_{j=1}^{m_i} (x_{ij} - n_{ij}\hat{p}_i)^2 \tag{2.10}$$

where n_{ij} is the number of observations from the jth cluster in the ith cell; x_{ij} is the number of successes in the ith cell and m_i is the number of clusters in the ith cell.

Combine the modification of clustering effect to the Hosmer-Lemeshow statistic and the Pigeon statistic, we have the new test statistic for goodness-of-fit test of the multinomial logistic regression model as:

$$Cc = C/\hat{\delta}_{\cdot} = \sum_{k=1}^{g} \sum_{j=0}^{c} \frac{(O_{kj} - E_{kj})^2}{E_{kj}\hat{\delta}_{\cdot}} \tag{2.11}$$

$$Jc = J/\hat{\delta}_{\cdot} = \sum_{k=1}^{g} \sum_{j=0}^{c} \frac{(O_{kj} - E_{kj})^2}{\phi_{kj}E_{kj}\hat{\delta}_{\cdot}} \tag{2.12}$$

2.3 Simulations for Type I error

Construction of Simulation Data

The goal in this section is to determine if these four test statistics can retain the type I error in different situations.

In the simulation study, we will assume there are m clusters in the dataset, and observations within each cluster share some same covariates to make them correlated with each other. This situation is also common in longitudinal data when each patients have specific baseline information such as age, gender, etc.

The predictor variables and response variable we are using like below:

- Y: the response variable, which can be 0, 1 or 2. State 0 is treated as reference status.

- X_1: a random continuous variable, independent between clusters but may not be independent among observations in the same cluster.

- X_2: a discrete variable, unique for each cluster.

- X_3 always equal to 1, and is the intercept covariate.

The corresponding coefficients

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_{01} & \beta_{02} & \beta_{03} \\ \beta_{11} & \beta_{12} & \beta_{13} \\ \beta_{21} & \beta_{22} & \beta_{23} \end{pmatrix}$$

As state 0 is the reference state, we have $\beta_{01} = \beta_{02} = \beta_{03} = 0$.

Here are the steps to generate the data from null distribution with given set of coefficients $\boldsymbol{\beta}$:

1. We assume there are C clusters in the data. C is 50 here.

2. Each cluster contains k observations, where k is randomly chosen from 1-5.

3. Within each cluster:

$$X_1 \sim \text{Multivariate Normal}(\mu, \Sigma)$$

$$\Sigma = \begin{pmatrix} v & \rho & \cdots & \rho \\ \rho & v & \cdots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \cdots & v \end{pmatrix}$$

   X_2 is a given value unique for each cluster.

   $X_3 = 1$ represents for the intercept.

4. Calculate the multinomial logistic probabilities (π_0, π_1, π_2) for each observation with given coefficients $\boldsymbol{\beta}$.

5. Generate an independent U(0,1) variable. Then generate simulated Y using the rule: (i) $Y = 0$ if $u < \pi_0$, (ii) $Y = 1$ if $u < \pi_0 + \pi_1$, (iii) $Y = 3$ otherwise.

6. Fit a multinomial logistic regression model based on simulated data and obtain the estimated probabilities $(\hat{\pi}_0, \hat{\pi}_1, \hat{\pi}_2)$.

7. Calculate different statistics based on simulated beta and estimated probabilities.

8. Repeat steps all above for B times (here $B = 10000$) . Calculate the rejection proportion of different nominal α levels.

Simulation Results

Table 2.1: Multinomial regression coefficients and tuning parameter for different model settings

Model	μ	v	ρ	β_{11}	β_{12}	β_{13}	β_{21}	β_{22}	β_{23}
1	1	1	0.5	0.1	0.5	0.1	0.2	1.2	0.2
2	2	1	0.5	0.1	0.5	0.1	0.2	1.2	0.2
3	2	4	0.5	0.1	0.5	0.1	0.2	1.2	0.2
4	1	2	0.5	0.1	0.5	0.1	0.2	1.2	0.2
5	3	8	0	0.1	1.5	0.1	0.2	1.2	0.15
6	2	6	0	1	1.5	0.1	2	1.2	0.15

Referring to Table 2.2, all tests performed poorly at alpha level of 0.01; When alpha level is 0.05 or 0.1, the Hosmer-Lemeshow statistic (C) and Pigeon's statistic (J) both inflated the type I error, while the two statistics modified for the cluster effect (Cc and Jc) both control the type I error well. This result shows that the modification for the cluster effect is important. We can also see in these two models, Cc has a better control than Jc, which shows the modification for underdispersion is not necessary here.

For Model 3 and 4, we make the variance of X_1 larger (2 times mean), hence, the linear predictor between observations within the same cluster varies more. The type I error in Model 3 shows that the performance of Jc (0.0501) is better than Cc (0.0411) when nominal $\alpha = 0.05$ and Jc (0.09520) comparing to Cc (0.0872) when nominal $\alpha = 0.10$. These simulation results shows some advantage in modification for underdispersion.

In Model 5, we set the correlation of X_1 to be independent within each cluster, also we set X_2 to be independent for each observation. There is no cluster effect but large difference in estimated probabilities. As a result, the ϕ is 0.73 now comparing

Table 2.2: Type I error for different model settings

Model	Result (0.01)				Result (0.05)				Result (0.1)			
	C	J	Cc	Jc	C	J	Cc	Jc	C	J	Cc	Jc
1	0.035	0.042	0.054	0.020	0.106	0.122	0.054	0.066	0.179	0.199	0.108	0.118
2	0.043	0.044	0.058	0.024	0.102	0.115	0.058	0.067	0.170	0.188	0.097	0.108
3	0.022	0.032	0.041	0.011	0.080	0.091	0.041	0.050	0.140	0.161	0.087	0.095
4	0.037	0.042	0.054	0.016	0.103	0.112	0.054	0.061	0.167	0.187	0.098	0.117
5	0.006	0.020	0.018	0.021	0.016	0.051	0.018	0.058	0.038	0.090	0.037	0.097
6	0.007	0.018	0.012	0.024	0.010	0.044	0.012	0.054	0.013	0.079	0.013	0.089

Table 2.3: Modification Parameter

Model	ϕ	Δ
1	0.93	1.22
2	0.96	1.21
3	0.94	1.21
4	0.96	1.22
5	0.73	1.01
6	0.49	1.00

to close to 1 in previous models, and Δ is almost 1 now. The type I error for J and Jc are close to the nominal α level. In model 6, we set β_{11} and β_{21} to be large to enlarge the difference of linear predictor. The ϕ decreased to 0.49, and we can also see effect of the modification for underdispersion.

2.4 Simulation for Power

Construction of Simulation Data

- Y: the response variable, which can be 0, 1 or 2. State 0 is treated as reference status.

- X_1: a random continuous variable, independent between clusters and may not be independent between observations in the same cluster.

- X_2: a discrete variable, unique for each cluster.

- $X_3 = 1$ all the time, intercept covariate.

- X_1^2: square of X_1

The corresponding coefficients

$$\boldsymbol{\beta} = \begin{pmatrix} \beta_{01} & \beta_{02} & \beta_{03} & \beta_{0s} \\ \beta_{11} & \beta_{12} & \beta_{13} & \beta_{1s} \\ \beta_{21} & \beta_{22} & \beta_{23} & \beta_{2s} \end{pmatrix}$$

As state 0 is the reference state, we have $\beta_{01} = \beta_{02} = \beta_{03} = \beta_{0s} = 0$.

Now we have the model:

$$log\frac{\pi_{ij}}{\pi_{i0}} = \beta_{j1} \cdot X_{i1} + \beta_{j2} \cdot X_{i2} + \beta_{j3} \cdot X_{i3} + \beta_{js} \cdot X_{i1}^2 \tag{2.13}$$

where $j = 0, 1, 2$. Under the null hypothesis $\beta_{js} = 0$. And under alternative hypothesis $\beta_{js} \neq 0$.

Simulation Results

Table 2.6 are the results of power analysis based on 40 clusters and 1000 simulations.

Table 2.4: Multinomial regression coefficients and tuning parameter for different simulation settings

Model	μ	v	ρ	β_{11}	β_{12}	β_{13}	β_{1s}	β_{21}	β_{22}	β_{23}	β_{2s}
1	1	1	0.5	0.2	0.4	0.2	0	0.1	1	0.1	0
2	1	1	0.5	0.2	0.4	0.2	1	0.1	1	0.1	-1
3	1	1	0.5	0.2	0.4	0.2	2	0.1	1	0.1	-2
4	1	1	0.5	0.2	0.4	0.2	5	0.1	1	0.1	-5
5	1	1	0.5	0.2	0.4	0.2	8	0.1	1	0.1	-8

Table 2.5: Modification Parameter

Model	ϕ	Δ
1	0.99	1.22
2	0.86	1.36
3	0.86	1.32
4	0.94	1.21
5	0.97	1.06

Table 2.6: Percentage of null hypothesis rejections for different simulation settings

Model	Result (0.05)				Result (0.1)			
	C	J	Cc	Jc	C	J	Cc	Jc
1	0.098	0.101	0.041	0.041	0.206	0.213	0.086	0.089
2	0.115	0.155	0.075	0.100	0.171	0.237	0.125	0.161
3	0.144	0.187	0.123	0.136	0.210	0.249	0.165	0.201
4	0.224	0.245	0.206	0.235	0.283	0.299	0.274	0.302
5	0.285	0.428	0.285	0.428	0.428	0.528	0.428	0.528

Table 2.5 shows the modification parameters in each model. ϕ is the parameter to modify the underdispersion, suggested by Pigeon's paper. Δ is the parameter of Rao's first order correction, which can adjust for clustering effect.

Table 2.6 shows the percentage of times a test reject the null hypothesis for different simulation settings. In this table, column C lists the percentage using the traditional chi-square statistic (Hosmer-Lemeshow statistic); column J lists the percentage using the Hosmer-Lemeshow statistic modified for underdispersion by Pigeon, and columns Cc and Jc are using the new statistics considering the clustering effects, correspondingly based on C and J. We include the results for α size is 0.05 or 0.1 in the table.

The null hypothesis is that the square term should not be in the model, which means both β_{1s} and β_{2s} are 0. Model 1 represents this scenario, so that model 1 shows the type I error of four statistics with clustering effect presents. From the results, we can see the traditional chi-square statistic C (0.098)and J (0.101) have inflated type I errors, and the modified statistics Cc (0.041) and Jc (0.041) can preserve the type I error better, when clustering exists.

Model 2 to Model 5, as the coefficients of square term (β_{1s} and β_{2s}) are not 0, implying the percentage of rejection of the null hypothesis represents the power of each test. As traditional statistics C and J cannot preserve type I error with clustering effect exists, I use the 95 percent quantile in model 1 as the new critical value, instead

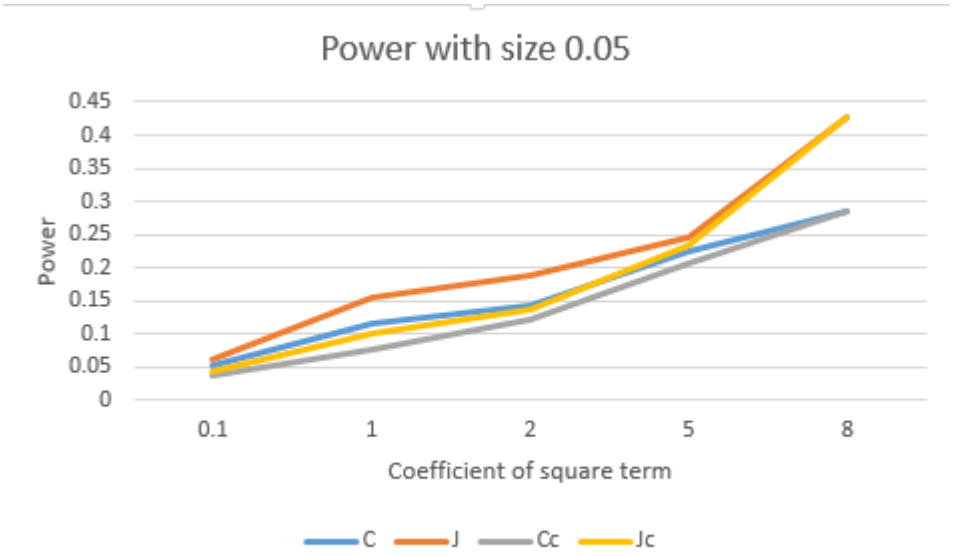

Figure 2.1: Power with Change of Coefficient of Squared Term

of using the quantile of chi-square distribution. In model 2, β_{1s} is 0.,1 and β_{2s} is -0.1, all four tests have low power around the α level. As the simulation model departs from the null hypothesis (β_{1s} from 0.1 to 8 and β_{2s} from -0.1 to -8), the power of all four tests are increasing as shown in the Figure 2.1.

From Figure 2.1, we realize in our setting, all those 4 tests have relatively low power. We can see as the coefficients are further from 0, the power of C and Cc (or J and Jc) are closer to each other. The Power of J is larger than the power of C for all settings.

2.5 Application

Here the goodness-of-fit test for multinomial regression is applied to the Nun Study data described in chapter 1. We note that the Nun Study data have clustering effects as each participant contributes four observations on average. For each row of the one-step transition matrix, a multinomial logistic regression model is fitted to calculate the transition probabilities. For each, we have 3 different estimated probabilities to

group the observations: probability to state 1, probability to state 2, probability to state 3. As state 4 (dementia) and state 5 (death) are both absorbing states, we do not consider the grouping strategy based on them.

Table 2.7 shows the p-values of the Nun Study data with different goodness-of-fit test statistics. As the p-values are different based on different grouping methods, it is still unclear about how to group the observations better, we list all results here. No p-values are significant. The p-value of transition from state 3 sorted by p2 are marginally significant for statistic C and J (0.117 and 0.112) correspondingly, but not significant when modified by the clustering effect.

These results basically showed the we cannot reject the multinomial regression model used to calculate each row of the one-step transition matrix. As we don't have a good strategy to group the data in multinomial case, we list all results together to make a decision.

2.6 Conclusion and Future work

In this chapter, we introduced several goodness-of-fit tests for the multinomial logistic regression model, and proposed a new statistic to deal with clustered data. We also examined the type I error control under null distributions for these statistic by simulations. The results showed that the standard or grouped Pearson's chi-square test have poor type I error control when clustering is present. The test statistic we proposed using a Rao-Scott first-order correction performed well. We also showed that when expected probabilities within cells varies much, a modification for the underdispersion is necessary, so we suggested to use the statistic modified by Pigeon (1999).

We applied those test statistics to the Nun Study to verify the multinomial regression model that has been used to calculate one-step transition matrix in some models. Our results showed that we failed to reject the multinomial regression model

Table 2.7: Nun Study Goodness of Fit Test Results

Test Results of Transitions from State 1									
	Statistic					P-value			
	C	J	Cc	Jc	Delta	C	J	Cc	Jc
Sort by p1	40.22	40.45	33.02	33.20	1.22	0.15	0.15	0.42	0.41
Sort by p2	35.13	35.31	28.84	28.98	1.22	0.32	0.31	0.63	0.62
Sort by p3	32.74	33.18	26.88	27.24	1.22	0.43	0.41	0.72	0.71
Test Results of Transitions from State 2									
	Statistic					P-value			
	C	J	Cc	Jc	Delta	C	J	Cc	Jc
Sort by p1	28.34	28.39	20.85	20.89	1.36	0.65	0.65	0.93	0.93
Sort by p2	27.62	28.02	20.32	20.62	1.36	0.69	0.67	0.95	0.94
Sort by p3	25.66	25.70	18.88	18.91	1.36	0.78	0.78	0.97	0.97
Test Results of Transitions from State 3									
	Statistic					P-value			
	C	J	Cc	Jc	Delta	C	J	Cc	Jc
Sort by p1	36.47	36.85	28.47	28.78	1.28	0.27	0.25	0.65	0.63
Sort by p2	41.71	41.95	32.57	32.76	1.28	0.12	0.11	0.44	0.43
Sort by p3	36.74	36.92	28.69	28.83	1.28	0.26	0.25	0.63	0.63

as a poor fit to the data.

There are still some problems to be answered here. One issue is we have different grouping strategy in multinomial regression model considering the estimated probabilities. Fagerland (2008) suggested to use the reference state to group the observations. Pigeon (1999) did not make a suggestion, and just listed all results based on different grouping strategies. This problem can be more serious when there is no obvious reference state in the data.

Another problem here for these goodness-of-fit tests is that there is no specific alternative making it difficult to estimate the power for these tests. The smoothed residuals test proposed by Goeman and Cessie (2006) has a specific alternative. Their method is based on the distance in covariates space instead of estimated probabilities so that it can avoid the choice of grouping strategy. We applied their method to the Nun Study, but the issue is that we cannot handle the cluster effect well, which seems to be serious in the Nun Study data.

Chapter 3 SIS-MCMC for Bivariate Multinomial Logistic Regression Model

3.1 Introduction

In this chapter, we apply the hybrid scheme of the SIS procedure and MCMC procedure proposed by D. Kahle and Garcia-Puente (2015) to the multinomial logistic regression model with two discrete covariates. This hybrid scheme of the SIS and MCMC procedures takes advantages of both methods to sample contingency tables from the conditional sample space. The hybrid scheme first runs the SIS procedure to sample tables independently from the conditional sample space with the uniform distribution, then it uses these sampled tables as initial tables to run multiple chains via the MCMC to sample tables from the conditional sample space with hypergeometric distribution.

With sample from SIS-MCMC procedure, we apply chi-square test and likelihood ratio test (as described in Chapter 1) to the multinomial logistic regression model for Nun study. We include the presence of apolipoprotein E-4 allele (Apoe4) and different levels of education as the covariates. We also consider age as a factor in the model with three different grouping method: ignore age, separate into 2 groups based on age and separate into four groups based on age. Based on the p-value from the SIS-MCMC samples, we recommend to use the model with four age groups to estimate the transition probability matrix.

This chapter is organized as follows: In Section 2, we use the multinomial logistic model to estimate the transition probability matrix, and we show its corresponding chi-square test statistic, and likelihood ratio statistic. In Section 3, we show algorithms to apply the SIS procedure and MCMC procedure to the multinomial logistic regression model. The goodness-of-fit test results of Nun Study based on the SIS-

MCMC procedure are summarized in Section 4.

3.2 SIS initialized MCMC

In order to sample contingency tables with marginal constraints by the MCMC method, one problem is that one may never be able to compute a Markov basis. For three-way contingency tables with fixed 2-margin constraints, the number of elements in a Markov basis can be too large to compute (De Loera and Onn (2005)). Even if we have a Markov Basis, with the standard MCMC, it can take a long time to converge to a stationary distribution in order to satisfy the independent assumption through a Markov Chain.

In order to solve the feasibility problem of computing a Markov basis and connectivity of a chain, D. Kahle and Garcia-Puente (2015) suggested the hybrid scheme of the SIS and MCMC procedures. The sampling scheme is outlined as follows:

1. Compute the sufficient statistics from the observed table X_0.

2. Uniformly sample the tables $X_1, \ldots, X_n$ from the conditional state space given the sufficient statistics of table X_0 by the SIS procedure.

3. Use sampled tables $X_1, \ldots, X_n$ in Step 2 as initials to run n many Markov chains to sample tables from the conditional state space according to the hypergeometric distribution given the sufficient statistics.

3.3 SIS-MCMC with Nun study

In this section, we apply SIS-MCMC algorithm to the Nun study data as described in chapter 1. To estimate the transition probability matrix (1.36), a multinomial logistic regression model can be constructed as:

$$log(\frac{P_{sv}}{P_{s1}}) = \alpha_{sv} + \beta_{1sv}X_1 + \beta_{2sv}X_2 \tag{3.1}$$

where $s = 1, 2, 3$ and $v = 1, 2, 3, 4, 5$.

In the above model, X_1 is the presence of apolipoprotein E-4 allele (APOE-4) with two levels (present and absent) and X_2 is the education with three levels (non-college degree, college level degree and post graduate degree). P_{sv} is the probability that a patient transfers from state s to state v, which is also the elements of the one-step transition probability matrix.

In studies of aging and Alzheimer disease, not only the presence of apolipoprotein e-4 allele and education level should be considered, but age is also an important factor that affects the transition probability among cognitive states. In this paper, we deal with age by three strategies: ignore age and treat participants in different ages the same; set a cutoff point 85 for age and analyze the transitions probability matrix with patients with age larger than 85 or not larger than 85 separately; or separate the data by quantiles of age, with the corresponding cutoff as 83.61, 87.12 and 90.54. Under these three different settings, we analyze the Nun Study data by two-way contingency table within each age group separately. Table 3.1 is the frequency table of transitions in Nun Study data without age seperation. Table 3.2 is the frequency table of transitions in Nun Study data separated at age equals 85. Table 3.3 is the frequency table of transitions in Nun Study data separated by age quantiles.

Table 3.1: Frequency Table of Transitions in Nun Data Ignore Age

Prior State	Current State				
	1	2	3	4	5
1	593	197	54	5	48
2	177	697	136	82	83
3	16	39	184	75	94

In the analysis of Nun Study data, as the prior state in each transition can be state 1, 2, or 3, we need to estimate the first 3 rows in the one-step transition probability matrix. Probabilities in each row can be estimated by the logistic regression model

Table 3.2: Frequency Table of Transitions in Nun Data by Age 85

Age	Prior State	Current State				
		1	2	3	4	5
$\leq$85	1	259	89	14	0	9
	2	91	256	31	23	16
	3	3	9	25	16	11
$>$ 85	1	334	108	40	5	39
	2	86	441	105	59	67
	3	13	30	159	59	83

Table 3.3: Frequency Table of Transitions in Nun Data by Age Quantiles

Age	Prior State	Current State				
		1	2	3	4	5
First Quantile	1	187	73	10	0	8
	2	73	186	18	13	9
	3	2	5	18	12	7
Second Quantile	1	182	44	15	3	10
	2	49	182	26	21	14
	3	5	9	32	8	19
Third Quantile	1	145	46	17	2	16
	2	32	169	43	16	20
	3	6	12	54	16	24
Fourth Quantile	1	79	34	12	0	14
	2	23	160	49	32	40
	3	3	13	80	39	44

in Equation (3.1). Thus, when we stratify the data into two sets by age equals to 85, we have six models in total to estimate the all the transition probabilities, and twelve models will be estimated if we separate data into four groups by quantile of patients' age. Here, we use the transitions from intact cognition (state 1) with participants older than 85 as an example. The other cases can be analyzed similarly.

As patients can transfer to five different states at each transition, the dataset of first model can be described as five two-way contingency tables. We can call it as a "point" here for simplicity. Each table contains two rows corresponding the presence of APOE-4 (X_1) and three columns corresponding to three levels of education (X_2). For example, the contingency table of the transitions from state 1 (intact cognition)

with age larger than 85 is as Table 3.4. Table 3.4 corresponds to a more detailed look at the first row in Table 3.2. The six values in Table 3.4 with current state equals 1 sum up to 334, which is the first number in Table 3.2 when age is larger than 85.

Table 3.4: Contingency Table From State 1 With Age> 85

Current state	Contingency table			
	Apoe4	non-college	college	post graduate
1	absence	5	113	185
	presence	0	4	27
2	absence	5	33	60
	presence	0	6	4
3	absence	0	14	22
	presence	0	1	3
4	absence	2	1	0
	presence	0	1	1
5	absence	2	9	24
	presence	0	1	3

Table 3.5: SIS sample based on Table 3.4

Current state	Contingency table			
	Apoe4	non-college	college	post graduate
1	absence	6	111	185
	presence	0	7	25
2	absence	8	30	59
	presence	0	6	5
3	absence	0	15	21
	presence	0	0	4
4	absence	0	3	1
	presence	0	0	1
5	absence	0	11	25
	presence	0	0	3

To apply the SIS-MCMC algorithm to each model in the Nun Study data, the first step is to generate 50 starting contingency tables by the SIS. Those 50 starting

points are independent with each other, while preserving the same sufficient statistics with the real data. Table 3.5 is one SIS sample based on Table 3.4.

Secondly, apply MCMC procedure to each starting point. Here, we set burn-in to 100, sample size 1000, and select one sample point each 20 steps.

Thirdly, for each sample point from MCMC, calculate the goodness-of-fit test statistic. Figure 3.1 shows the histogram of those test statistics. Based on Table 3.4, we can calculate the Chi-square test statistic 9.476. The p-value is the percentage of how many test statistics are larger than the statistic of real data (9.476). The result is 0.046 for the Nun Study data with prior state equals 1 and age larger than 85.

Above steps can be repeated for all other models. We also calculate the likelihood ratio test for the SIS-MCMC samples here. As we can see in Table 3.4, the Nun data is pretty sparse when we use contingency tables to describe it. This problem is even worse when we have more separations by age. As the chi-square test usually requires the expected cell number larger than 5, the results of chi-square test might be skewed. We include the results of likelihood ratio test in Table 3.7, and we include both the p-value based on exact distribution and asymptotic chi-square distribution.

Results

The results of chi-square test and likelihood ratio test are shown in Table 3.6 and Table 3.7. We may use 0.05 as the cutoff of p-value for each model. The models with a p-value less than 0.05 may imply some important covariates are missing in the model, or the multinomial logistic regression model assumption is not valid in this scenario.

For the model ignoring the age, all p-values of likelihood ratio test are not significant but relatively small. The p-value of chi-square test from state 1 (intact cognition)

Table 3.6: Test Statistic and P-value of Chi-Square test

SIS-MCMC Chi-square results			
	Prior State	Test Statistic	P-value
No Age	1	9.115	0.049
	2	14.127	0.096
	3	14.755	0.070
SIS-MCMC for NUN with 2 Age Groups			
Age	Prior State	Test Statistic	P-value
≤ 85	1	2.514	0.510
	2	17.550	0.038
	3	29.491	0.009
> 85	1	9.476	0.046
	2	8.408	0.202
	3	12.045	0.146
SIS-MCMC for NUN with 4 Age Groups			
Age group	Prior State	Statistic	P-value
First Quantile	1	1.180	0.744
	2	15.363	0.072
	3	13.366	0.031
Second Quantile	1	8.179	0.060
	2	8.000	0.369
	3	14.333	0.080
Third Quantile	1	8.951	0.116
	2	6.133	0.471
	3	9.477	0.192
Fourth Quantile	1	6.490	0.095
	2	11.373	0.379
	3	7.217	0.810

is marginally significant (0.049). For this transition probability from state 1, if we use the model with two age groups, the p-value of younger participants (younger than 85) is not significant (0.510), but the p-value of the elder participants is significant (0.046). If we check the p-values with 4 age groups, all p-values from chi-square test with prior state 1 are not significant. Also, all p-values with prior state 1 from likelihood ratio tests are not significant. Thus, we can conclude the significance of chi-square test from the model with prior state 1 and no age separation may mainly from the elder participants and is doubtable as the results of likelihood ratio tests are not significant.

Table 3.7: Test Statistic and P-value of Likelihood Ratio Test

SIS-MCMC LRT results				
	Prior State	Test Statistic	P-value	P-value(Asymptotic)
No Age	1	9.254	0.061	0.321
	2	15.285	0.057	0.054
	3	15.285	0.057	0.054
SIS-MCMC for NUN with 2 Age Groups				
Age	Prior State	Test Statistic	P-value	P-value(Asymptotic)
≤ 85	1	2.772	0.569	0.948
	2	15.199	0.061	0.055
	3	17.934	0.010	0.022
> 85	1	9.093	0.074	0.335
	2	9.377	0.289	0.312
	3	11.986	0.182	0.152
SIS-MCMC for NUN with 4 Age Groups				
Age group	Prior State	Statistic	P-value	P-value(Asymptotic)
First Quantile	1	1.597	0.692	0.991
	2	12.117	0.171	0.146
	3	14.077	0.016	0.080
Second Quantile	1	8.267	0.073	0.408
	2	9.108	0.328	0.333
	3	10.336	0.141	0.242
Third Quantile	1	8.730	0.103	0.366
	2	6.186	0.548	0.626
	3	8.272	0.325	0.407
Fourth Quantile	1	6.991	0.058	0.538
	2	9.913	0.184	0.271
	3	6.357	0.637	0.607

All p-values are not significant for models with older participants (older than 83) no matter the prior cognitive state, in both chi-square test and likelihood ratio test. The results show that a multinomial logistic regression model is legitimate to estimate the transition probability matrix between cognitive states for older participants.

As shown in Table 3.7, in the results of likelihood ratio test, when we separate data into four groups by quantile of age, the only significant p-value (0.016) appears in the model with participants younger than 83 (group 1 in 4 age groups part) and from state 3 (global impairment). It suggests we may not use a multinomial logistic

Table 3.8: Contingency table of age group 1 from state 3

Current state	Contingency table			
	Apoe4	no college	college	post college
1	absence	0	1	1
	presence	0	0	0
2	absence	1	1	1
	presence	1	1	0
3	absence	3	3	5
	presence	0	7	0
4	absence	2	2	2
	presence	0	1	5
5	absence	0	1	4
	presence	0	2	0

regression model to calculate the transition probabilities with younger participants (younger than 83) from a global impairment state, or some key covariates or interactions are missing in the model. The results of chi-square test also support this result, which has a p-value of 0.031. However, when we look into this situation, the contingency table in this model is like Table 3.8. The contingency table for transitions from state 3 (global impairment) with participants younger than 83 is so sparse that we may ignore its significance. Regardless of this special case, we can conclude that when we separate all the Nun Study data into four groups based on age quantiles, all test statistics of goodness of fit tests are not significant. The logistic regression model with presence of APOE-4 and education level as covariates can be used to estimate the transition probability matrices.

Based on all these test results, we recommend using multinomial logistic regression with presence of APOE-4 and education level as covariates and separate the Nun Study data into four groups by age quantile.

3.4 Conclusion

In this chapter, we considered a multinomial logistic regression model with two categorical covariates. Two-way contingency tables are used to describe this type of data. We applied a hybrid scheme of sequential importance sampling (SIS) method and Monte Carlo Markov Chain (MCMC) to sample the sets of two-way contingency tables for multinomial logistic regression model. Based on the SIS-MCMC samples, we generated the exact distribution of chi-square goodness-of-fit test statistics and likelihood ratio test statistics. From the exact distribution, we can calculate more accurate p-value of a given dataset.

To apply the SIS-MCMC procedure to the Nun Study data, we considered three different grouping methods for age and fit the data with 3, 6, or 12 models based on the prior cognitive states and grouping methods of age. As the sampling method is based on two-way contingency tables here, when we need a third variable that affects the response variable, we recommend using the stratification of the third variable for the contingency tables. A problem is that the standard of the stratification method is not clear. It should be determined by the dataset and related previous research. As for the Nun Study data, we encountered severe sparse table since we separated data into four groups by age, which results to 12 models to estimate. Considering the sample size and sparseness of the contingency tables, we decided not to separate Nun Study data into more groups. After considering the p-value of both chi-square test and likelihood ratio test in all different cases, we recommend the model with four age groups for transition probability matrix in Nun Study.

In this chapter, we analyzed the multinomial logistic regression model by describing the data as contingency tables. We extended the MCMC method based on Markov basis of binomial logistic regression to multinomial logistic regression models. By now, as this sampling method can only deal with two-way contingency tables, we applied this method to a bivariate logistic regression models. However, in some cases,

more than two factors are believed to affect the response variable. We tried to include a third covariate by stratification ,which is also limited by the sparseness of dataset. More work is needed to deal with the situation with more than two covariates. Also, we are considering categorical covariates in this paper, we can also use two way contingency tables to describe data with continuous covariates by appropriate grouping method.

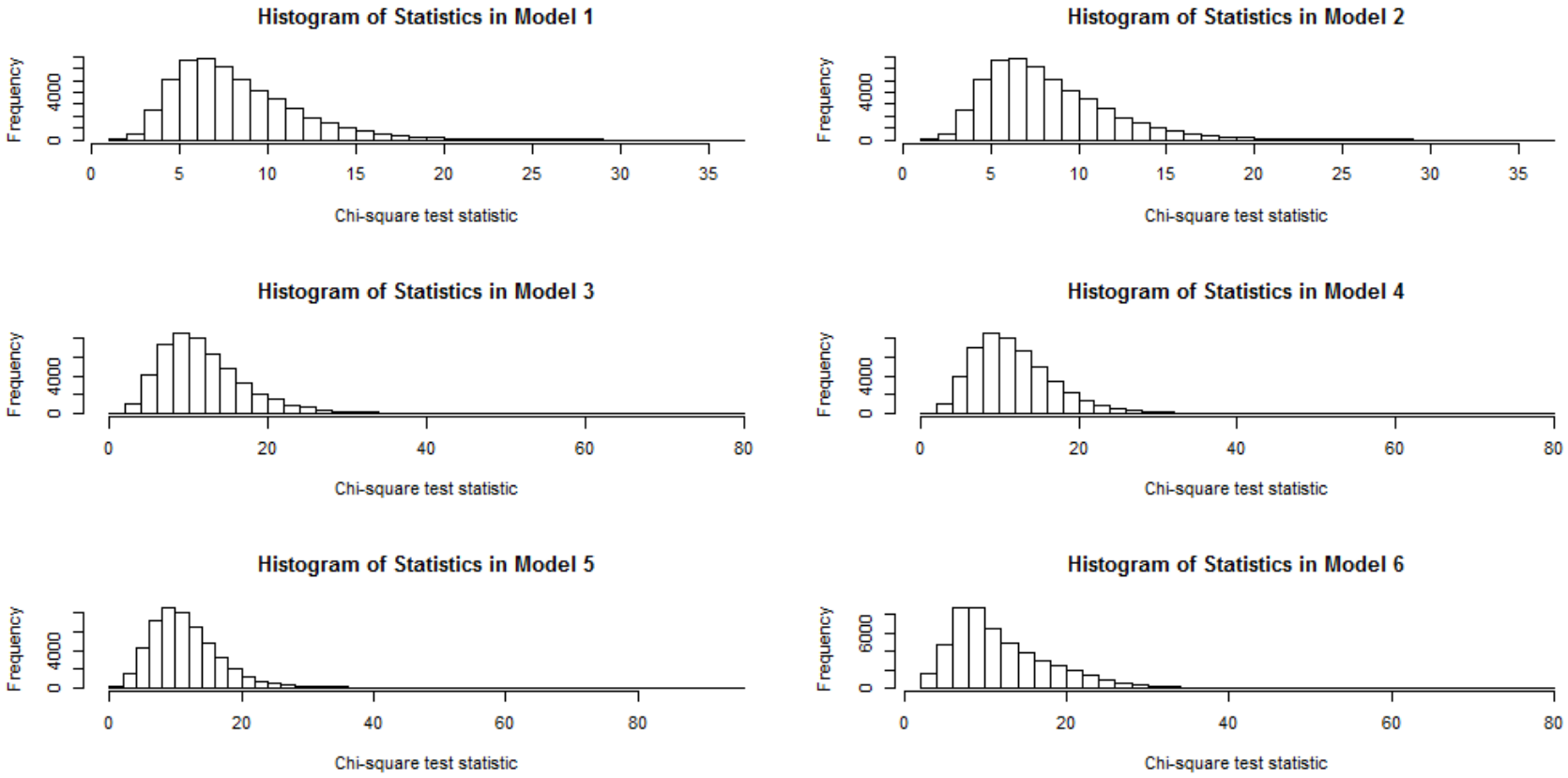

Figure 3.1: Histogram of Chi-square Statistics

Chapter 4 Logistic Regression with Right Censored Ordinal Covariate

4.1 Introduction

The traditional logistic regression model restricts the analysis to observations with complete data ignoring all the observations with incomplete data, such as missing or censored values. However, in many datasets with heavy censoring, much information will be lost when incomplete observations are simply deleted. Right censored data commonly arise in many epidemiology or medical studies. For example, in the study of Alzheimer's disease, the length of time in mild cognitive impairment is right censored whenever a patient is already in that cognitive state at the start of the study.

There are different general approaches for estimating the regression parameters in a model with censored covariates. First, we can just use the censored covariates without adjustment, which is well known to lead to bias in estimates (Atem et al. (2015)). Second, all observations with censored values could be excluded from the analysis. This is the most commonly used method, which is referred as a "complete-case" analysis. By this method, the analysis is only based on those observations with complete covariates, and could yield biased estimates if data are not missing completely at random. Even if the assumption that the censoring is missing completely at random, which is usually violated in observed data, a problem on the complete-case analysis is the loss of efficiency especially when the percentage of censoring gets large.

The third way is to use maximum likelihood estimates under the assumption of censored covariates. Atem et al. (2015) and Austin and Hoch (2004) considered a linear regression model with censored independent variables. For a generalized linear model, there is some literature on the treatment of missing value covariates (Vach and Schumacher (1993), Vach and Blettner (1995)). However, there are limited studies focused on dealing with censored covariates.

This chapter is devoted to the scenario where a logistic regression model would be appropriate if no censored data would have occurred. We also assume that the percentage of a censored covariate is independent of the exact value of that covariate. For simplicity, we only consider the case with one completely observed ordinal covariate and another right censored ordinal covariate. This approach can be extended to more complex cases.

In this chapter, we will introduce the likelihood function based on joint probability and a method to estimate nuisance parameters in the first section. Secondly, we show the performance of our new method comparing with the logistic regression model with only complete cases and logistic regression with penalized likelihood function method with different censoring percentages and sample sizes. Then, we applied our new method to a study of relationship between arteriolosclerosis and patient's length of time in mild cognitive impairment. Lastly, we discuss the new method, and propose some potential future work.

4.2 Method

We assume a logistic regression model for a binary outcome Y (values 0 or 1) given two categorical predictors: X_1 with J possible values and X_2 with K possible values, where X_2 might be right censored for some observations. We assume

$$\mu_{kj}(\boldsymbol{\beta}) := P(Y = 1|X_1 = j, X_2 = k) = \frac{\exp(\beta_0 + \beta_{1j} + \beta_{2k})}{1 + \exp(\beta_0 + \beta_{1j} + \beta_{2k})} \tag{4.1}$$

with restrictions $\beta_{11} = 0$ and $\beta_{21} = 0$, here $j = 2, 3, \cdots, J$ and $k = 2, 3, \cdots, K$. We denote the vector $(\beta_0, \beta_{1j}, \beta_{2k})$ as $\boldsymbol{\beta}$.

As the value of X_2 might be right censored, C_2 is the indicator variable of whether

X_2 is observed or not

$$C_2 := \begin{cases} 1 & \text{if } X_2 \text{ is observed} \\ 0 & \text{if } X_2 \text{ is censored} \end{cases} \tag{4.2}$$

Due to the right censoring of X_2, we define a random variable Z_2 with $2K$ possible values

$$Z_2 := \begin{cases} X_2 & \text{if } X_2 \text{ is observed} \\ K + X_2 & \text{if } X_2 \text{ is censored} \end{cases} \tag{4.3}$$

As suggested by Vach and Schumacher (1993), the conditional probability of occurrence of right censoring is assumed to be:

$$P(C_2 = 1|Y = i, X_1 = j, X_2 = k) = P(C_2 = 1|Y = i, X_1 = j) = q_{ij} \tag{4.4}$$

Hence, we assume that the occurrence of right censored value does not depend on the true value of X_2.

To describe the likelihood of the distribution of X_1 and X_2, we define:

$$\pi_{kj} := P(X_2 = k|X_1 = j) \qquad \tau_j := P(X_1 = j) \tag{4.5}$$

The joint distribution of (Y, X_1, Z_2) is given by:

$$P(Y = i, X_1 = j, Z_2 = k) = \begin{cases} q_{ij} P(Y = i, X_1 = j, X_2 = k) & \text{if } k \leq K \\ (1 - q_{ij}) P(Y = i, X_1 = j, X_2 \geq k - K) & \text{if } k > K \end{cases} \tag{4.6}$$

When $k \leq K$, we have

$$\begin{aligned} & P(Y = i, X_1 = j, X_2 = k) && (4.7) \\ = \; & P(Y = i|X_1 = j, X_2 = k) P(X_2 = k|X_1 = j) P(X_1 = j) && (4.8) \\ = \; & \{\mu_{kj}(\boldsymbol{\beta})\}^i \{1 - \mu_{kj}(\boldsymbol{\beta})\}^{1-i} \pi_{kj} \tau_j && (4.9) \end{aligned}$$

When $k > K$, we set $k = K + k^*$

$$
\begin{aligned}
& P(Y = i, X_1 = j, Z_2 = K + k^*) && (4.10)\\
= & \sum_{k=k^*}^{K} P(Y = i, X_1 = j, X_2 = k) && (4.11)\\
= & \sum_{k=k^*}^{K} P(Y = i|X_1 = j, X_2 = k)P(X_2 = k|X_1 = j)P(X_1 = j) && (4.12)\\
= & \sum_{k=k^*}^{K} \{\mu_{kj}(\boldsymbol{\beta})\}^i \{1 - \mu_{kj}(\boldsymbol{\beta})\}^{1-i} \pi_{kj}\tau_j && (4.13)
\end{aligned}
$$

To summarize, we have the likelihood based on joint distribution as:

$$
P(Y = i, X_1 = j, Z_2 = k) = \begin{cases} q_{ij}\{\mu_{kj}(\boldsymbol{\beta})\}^i \{1 - \mu_{kj}(\boldsymbol{\beta})\}^{1-i} \pi_{kj}\tau_j & \text{if } k \leq K \\ (1 - q_{ij}) \sum_{k=k^*}^{K} \{\mu_{kj}(\boldsymbol{\beta})\}^i \{1 - \mu_{kj}(\boldsymbol{\beta})\}^{1-i} \pi_{kj}\tau_j & \text{if } k > K \text{ and } k = K \end{cases} \quad (4.14)
$$

Given n independent observations (y_r, x_{1r}, z_{2r}), the maximum likelihood estimation of $(\boldsymbol{\beta}, \boldsymbol{\pi})$ and τ, q can be estimated independently. The $(\hat{\boldsymbol{\beta}}^{\boldsymbol{ML}}, \hat{\boldsymbol{\pi}}^{\boldsymbol{ML}})$ result from maximizing:

$$
\begin{aligned}
L = & \prod_{k<K} q_{ij}\{\mu_{kj}(\boldsymbol{\beta})\}^i \{1 - \mu_{kj}(\boldsymbol{\beta})\}^{1-i} \pi_{kj}\tau_j && (4.15)\\
& \times \prod_{k \geq K, k = K + k^*} (1 - q_{ij}) \sum_{k=k^*}^{K} \{\mu_{kj}(\boldsymbol{\beta})\}^i \{1 - \mu_{kj}(\boldsymbol{\beta})\}^{1-i} \pi_{kj}\tau_j && (4.16)
\end{aligned}
$$

To estimate the parameters, we can use the estimated conditional probabilities based on frequency of contingency table as the initial values of the nuisance parameter $\boldsymbol{\pi}$. Then we can calculate the estimated value of $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\pi}})$ by Newton-Raphson algorithm.

If we define the contingency table as $n_{ijk} = \#\{Y = i, X_1 = j, Z_2 = k\}$

$$n_{.jk} = \sum_{i=0}^{1} n_{ijk} \tag{4.17}$$

$$n_{ij+} = \sum_{k=1}^{K} n_{ijk} \tag{4.18}$$

$$n_{ij.} = \sum_{k=1}^{2K} n_{ijk} \tag{4.19}$$

Then the initial $\hat{\boldsymbol{\pi}}^{\mathbf{0}}$ can be estimated by:

$$\hat{\boldsymbol{\pi}}^{\mathbf{0}} = \frac{n_{0j.}n_{0jk}/n_{0j+} + n_{1j.}n_{1jk}/n_{1j+}}{n_{0j.} + n_{1j.}} \tag{4.20}$$

4.3 Simulation Study

Construction of Simulation Data

The goal of this section is to study the performance of estimates from new method based on joint distribution. We include two independent covariates, X_1 and X_2 where X_1 is a binary variable completely observed and X_2 is a censored ordinal variable with three possible values 1, 2 and 3. We have:

- $X_1 \sim Binomial(p = 0.5)$
- $P(X_2|X_1) = \begin{Bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \end{Bmatrix}$
- $p_{i1} = P(X_2 = 1|X_1 = i)$ is chosen from uniform (0,1); p_{i2} is chosen from uniform (0,1-p_{i1}); $p_{i3} = 1 - p_{i1} - p_{i2}$.

The response variable Y is a binomial variable with probability

$$P(Y = 1|X_1 = j, X_2 = k) = \frac{\exp(\beta_0 + \beta_{1j} + \beta_{2k})}{1 + \exp(\beta_0 + \beta_{1j} + \beta_{2k})} \tag{4.21}$$

where $j = 1, 2$ and $k = 1$, 2 and 3. We also set $\beta_0 = -1$ $\beta_{12} = 1$ $\beta_{22} = 1$ and $\beta_{23} = 2$ for given coefficients.

The censoring percentage of X_2 is set as a constant number 30%, 50% or 70%. The sample size of each replicate is from 100 to 1000 and we replicate each scenario 500 times. We also tried simulations with more replicates, the results did not change much.

In logistic regression model with ordinal covariates, maximum likelihood estimates are often inconsistent or fail to converge in the cases of separation or quasi-separation. In logistic regression model with binary response variable, the separation occurs when all the response variables are the same in all observations with a certain value of a covariate. For example, this occurs if all observations with patients age 85 and older all have presence of arteriolosclerosis in a study of relationship between arteriolosclerosis and age, using the logistic regression model. This case is commonly seen in rare variants studies. Firth (1993) introduced a more effective way to deal with the above situation based on penalized likelihood. We also include Firth's model in addition to complete-case logistic model for comparison.

In this simulation part, we calculate the Firth's penalized likelihood by PROC LOGISTIC with option firth in SAS.

Simulation Results

With the same censoring percentage, the mean squared error of the estimates of all coefficients decrease when the sample size increases from 100 to 1000 across all three methods. With the same sample size, the mean squared error of estimates of all coefficients increase as the censoring percentage increases from 30 percent to 70 percent across all three methods, which make sense as we have more information and observations to estimate the model when the data suffering from censoring less.

When we compare the performance of the new method (Table 4.1) with logistic regression using only complete cases (Table 4.2), in general, the estimates of the new method have lower mean squared error. When the sample size is between 100 to 500, the mean squared error of the new method performs a lot better than the logistic regression model with complete cases. When the sample size is large (800 or 1000), complete case logistic regression model can give a better the estimate of β_0 in our simulation results. In these cases, the logistic regression model has relatively large sample size. For example, if the sample size is 1000, with 50% censoring, there are still 500 observations can be used in logistic regression model.

We also compare the new method with the Firth logistic regression model (4.3), which reduces the bias of the estimates. When the sample size is small, 100 through 500, the new method has better estimates comparing to Firth logistic model, but the Firth model performs better than the complete-case logistic regression. As the sample size increases, the Firth model outperformed in some estimations but the new method is not far behind.

To test the robustness of the assumption that the percentage of censoring is independent with the exact value of the covariate (equation 4.4), we also consider the situation that

$$P(censoring|X_2 = 0) = 30\% \tag{4.22}$$

$$P(censoring|X_2 = 1) = 50\% \tag{4.23}$$

$$P(censoring|X_2 = 2) = 70\% \tag{4.24}$$

The corresponding simulation results are in Table 4.5. With different sample size from 100 to 800, the estimates of new methods are more accurate than the estimates of complete-case logistic model or Firth logistic model. It shows that our new method is quite robust to the censoring mechanism.

To conclude, we showed our new method can estimate the coefficients with smaller mean squared error in most cases when the sample size is from 100 to 800 comparing to complete-case logistic regression model and logistic regression model with penalized likelihood. Our new method has similar performance with other methods when the sample size is relatively large, in which case the logistic regression model only based on complete cases has relatively large sample size and enough information.

Table 4.1: Mean Squared Error of Estimated Coefficients with New Method

Censor	Sample Size	Mean Squared Error			
		β_0	β_{12}	β_{22}	β_{23}
30%	100	0.337	0.375	0.666	0.592
	200	0.166	0.177	0.360	0.386
	300	0.120	0.134	0.242	0.302
	500	0.072	0.078	0.152	0.193
	800	0.054	0.050	0.124	0.177
	1000	0.047	0.047	0.098	0.161
50%	100	0.412	0.451	0.712	0.589
	200	0.215	0.205	0.422	0.380
	300	0.143	0.147	0.306	0.252
	500	0.118	0.082	0.225	0.200
	800	0.072	0.062	0.153	0.141
	1000	0.065	0.053	0.123	0.140
70%	100	0.463	0.374	0.943	0.705
	200	0.331	0.228	0.521	0.488
	300	0.255	0.200	0.532	0.364
	500	0.192	0.133	0.317	0.220
	800	0.160	0.118	0.261	0.177
	1000	0.146	0.098	0.245	0.165

Table 4.2: Mean Squared Error of Estimated Coefficients with Logistic Regression (Complete Cases)

Censor	Sample Size	Mean Squared Error			
		β_0	β_{12}	β_{22}	β_{23}
30%	100	0.419	0.534	0.795	0.733
	200	0.191	0.230	0.411	0.496
	300	0.138	0.163	0.275	0.395
	500	0.086	0.101	0.165	0.220
	800	0.049	0.054	0.119	0.127
	1000	0.041	0.051	0.088	0.120
50%	100	0.546	0.784	0.817	0.912
	200	0.284	0.384	0.542	0.684
	300	0.148	0.223	0.366	0.427
	500	0.123	0.134	0.244	0.316
	800	0.055	0.076	0.166	0.189
	1000	0.052	0.061	0.116	0.158
70%	100	0.769	1.004	1.259	1.231
	200	0.449	0.571	0.709	0.911
	300	0.314	0.448	0.641	0.634
	500	0.200	0.239	0.408	0.388
	800	0.117	0.161	0.236	0.319
	1000	0.091	0.102	0.184	0.242

4.4 Application and Results

The above method is applied to a study of the relationship between the presence of arteriolosclerosis and time a patient remains in mild cognitive impairment (MCI) status. The data were drawn from the SMART database. In this application, the response variable (Y) is the presence of arteriolosclerosis. The predictor of interest (X_2) is the time of a patient in MCI, which is a right censored variable as some of the patients were already in the MCI state when entered in this study. The variable time of a patient in MCI status is discretized as a ordinal variable with three levels: $0-4$ years, $4-8$ years and more than 8 years. Several choices are available for the control variable (X_1), we select whether the death age is larger than 85 or not (npdage85) and whether the patient is a female or not (female) here. Both of them are binary variables. In our dataset, we have a thousand observations with observed

Table 4.3: Mean Squared Error of Estimated Coefficients with Firth Logistic Regression

Censor	Sample Size	Mean Squared Error			
		β_0	β_{12}	β_{22}	β_{23}
30%	100	0.365	0.452	0.750	0.770
	200	0.184	0.230	0.415	0.462
	300	0.123	0.150	0.288	0.373
	500	0.091	0.102	0.165	0.223
	800	0.045	0.050	0.116	0.120
	1000	0.040	0.050	0.084	0.117
50%	100	0.511	0.723	0.988	1.079
	200	0.249	0.356	0.530	0.661
	300	0.142	0.203	0.361	0.477
	500	0.109	0.118	0.227	0.290
	800	0.056	0.075	0.159	0.192
	1000	0.050	0.059	0.116	0.162
70%	100	0.773	1.061	1.448	1.482
	200	0.416	0.581	0.777	0.985
	300	0.273	0.367	0.562	0.662
	500	0.170	0.213	0.387	0.383
	800	0.107	0.144	0.235	0.304
	1000	0.083	0.095	0.183	0.238

survive time in MCI and 384 observations with right censored stay time in MCI.

Table 4.6 is a summary table of the dataset used here. From that table, we can see that the proportion of censored time of a patient in MCI status is relatively large in this dataset, which is $358/1358 = 26.3\%$ of all observations. If traditional methods of logistic regression is applied, 26.3% of the data will be ignored.

As the control variable is a binary variable and the right censored variable time in MCI has three levels, we have $J = 2$ and $K = 3$ here. With the constraints $\beta_{11} = \beta_{21} = 0$, the estimated parameter and corresponding confidence intervals from Proc NLP are shown in Table 4.7.

The obvious advantage of fitting a logistic regression model with right censored

Table 4.4: Estimated Variance of Coefficients Based on New Method

Censor	Sample Size	Estimated Variance			
		β_0	β_{12}	β_{22}	β_{23}
30%	100	0.319	0.417	0.879	0.973
	200	0.171	0.207	0.447	0.408
	300	0.114	0.131	0.285	0.329
	500	0.070	0.077	0.143	0.191
	800	0.053	0.050	0.115	0.169
	1000	0.046	0.047	0.088	0.137
50%	100	0.463	0.546	1.387	0.936
	200	0.184	0.211	0.501	0.421
	300	0.122	0.145	0.340	0.287
	500	0.100	0.084	0.211	0.195
	800	0.058	0.061	0.136	0.112
	1000	0.053	0.053	0.103	0.116
70%	100	0.532	0.579	1.720	1.149
	200	0.275	0.266	0.735	0.514
	300	0.199	0.187	0.493	0.381
	500	0.136	0.129	0.269	0.168
	800	0.111	0.119	0.174	0.131
	1000	0.096	0.096	0.143	0.103

covariate is that all data can be used here. By allowing right censored in time of patients in MCI, we can use 358 more observations, which is 35.8% compared to the complete dataset.

From Table 4.6, it can be shown that as the survive time of patients in MCI increases, the percentage of arteriolosclerosis increases both in non-censored cases (78%, 96%, 99%) and censored cases (82%, 95%, 100%). We want to use a logistic regression model to verify this statement. The estimated parameters reflect that trend too. With the death age indicator (npdage85) as the control variable, the odds ratio between time in MCI between 4 to 8 years and time in MCI less than 4 years is $\exp(1.578) = 4.84$ and the odds ratio between time in MCI longer than 8 years and time in MCI between 4 to 8 years is $\exp(2.232 - 1.578) = 1.92$. With gender as the control variable, the odds ratio between time in MCI between 4 to 8 years and time

Table 4.5: Mean Squared Error of Estimated Coefficients with Varying Censoring Percentage

Sample Size	Mean Squared Error				Mean Squared Error(Complete)				Mean Squared Error(Firth)			
	β_0	β_{12}	β_{22}	β_{23}	β_0	β_{12}	β_{22}	β_{23}	β_0	β_{12}	β_{22}	β_{23}
100	0.375	0.384	0.803	0.594	0.534	0.623	0.962	0.975	0.490	0.623	0.917	1.094
200	0.154	0.160	0.401	0.286	0.195	0.292	0.448	0.575	0.167	0.255	0.445	0.628
300	0.117	0.124	0.312	0.267	0.138	0.195	0.351	0.520	0.126	0.168	0.321	0.583
500	0.072	0.082	0.173	0.194	0.092	0.128	0.209	0.386	0.082	0.118	0.220	0.400
800	0.051	0.055	0.125	0.140	0.053	0.085	0.159	0.249	0.048	0.078	0.139	0.249
1000	0.043	0.047	0.120	0.112	0.045	0.069	0.102	0.182	0.042	0.064	0.095	0.178

Table 4.6: Summary Table of Time in MCI and Arteriosclerosis

Censoring	Time in MCI							
	0-4		4-8		8+		ALL	
	N	% Arter=1	N	% Arter=1	N	% Arter=1	N	% Arter=1
Not Censored	788	78%	145	96%	67	99%	1000	82%
Censored	194	82%	103	95%	61	100%	358	89%

Table 4.7: Parameter Estimation and Confidence Interval

Parameter estimate with death age as control						
Parameter	Estimate	Approx Std Err	Lower Bound 95% CI	Upper Bound 95% CI	t Value	P-value
β_0	0.804	0.127	0.556	1.053	6.341	0.000
β_{12}	0.746	0.160	0.433	1.059	4.673	0.000
β_{22}	1.578	0.385	0.824	2.332	4.101	0.000
β_{23}	2.232	0.889	0.491	3.974	2.512	0.012
Parameter estimate with gender as control						
Parameter	Estimate	Approx Std Err	Lower Bound 95% CI	Upper Bound 95% CI	t Value	P-value
β_0	0.909	0.117	0.679	1.139	7.748	0.000
β_{12}	0.638	0.156	0.333	0.944	4.092	0.000
β_{22}	1.710	0.377	0.972	2.448	4.541	0.000
β_{23}	2.559	0.903	0.788	4.329	2.832	0.005

in MCI less than 4 years is $\exp(1.71) = 5.53$ and the odds ratio between time in MCI longer than 8 years and time in MCI between 4 to 8 years is $\exp(2.559 - 1.71) = 2.34$. It also can be shown that all parameters are statistically significant. From this point of view, we can conclude that after controlling by gender and death age, the longer a patient stays in MCI, the more likely that the patient has arteriolosclerosis.

Also if we consider the conditional probabilities $\boldsymbol{\pi}$, which are nuisance parameters in this model, the initial values of conditional probabilities based on equation 4.20 are:

$$\hat{\boldsymbol{\pi}}^{\mathbf{0}} = \begin{Bmatrix} 0.855 & 0.083 & 0.062 \\ 0.837 & 0.158 & 0.144 \end{Bmatrix}$$

The estimated conditional probabilities for the model are:

$$\hat{\boldsymbol{\pi}}^{\boldsymbol{ML}} = \begin{Bmatrix} 0.792 & 0.119 & 0.089 \\ 0.636 & 0.191 & 0.173 \end{Bmatrix}$$

We can conclude that comparing to $\hat{\boldsymbol{\pi}}^{\boldsymbol{ML}}$, the initial probability is a good starting estimation of the conditional probabilities.

In the Newton-Raphson method calculated by Proc NLP, there are different choices to estimate variance of coefficients. In this application, we are calculating the variance of coefficients by the equations below:

$$\begin{aligned} Cov &= \frac{nobs}{d} JJ(f)^{-1} && (4.25) \\ JJ(f) &= J(f)^T J(f) && (4.26) \\ J(f) &= (\nabla f_1, \cdots, \nabla f_m) = (\frac{\partial f_i}{\partial x_j}) && (4.27) \\ d &= nobs - df && (4.28) \end{aligned}$$

where f_i is the joint likelihood function for each observation. *df* is the number of parameters and *nobs* is set to the number of observations in the data set times the number of functions estimated.

To validate the calculation of the estimated variance of coefficients, we also estimate the variance by using the bootstrap algorithm, which relies on random sampling with replacement from the real dataset. The bootstrap gives us similar results compared to the variance estimated by Newton-Raphson algorithm through equations above. The comparison of those two methods are summarized in Table 4.8.

Table 4.8: Estimated Variance by Bootstrap

Estimated variance with death age as control		
Parameter	Bootstrap Algorithm	Newton-Raphson method
β_0	0.0967	0.0944
β_{12}	0.1214	0.1206
β_{22}	0.2551	0.2565
β_{23}	0.2949	0.4714
Estimated variance with gender as control		
Parameter	Bootstrap Algorithm	Newton-Raphson method
β_0	0.0802	0.0812
β_{12}	0.1300	0.1313
β_{22}	0.2621	0.2600
β_{23}	0.3599	0.4946

4.5 Conclusion and Discussion

In this chapter, we introduced a new method to fit the logistic regression model with right censored covariate based on joint probability. The calculation is based on maximum likelihood estimates and Newton-Raphson method. We also showed a good estimate of initial value of nuisance parameter.

We set up different simulation scenarios to show the accuracy of the new method compared to traditional logistic regression model with complete cases and logistic regression with penalized likelihood function(Firth). Simulation results showed our new method outperformed both existing methods when sample size is relatively small or medium (100-500) with different censoring percentages. When sample size is relatively large, for example 800 or 1000, the traditional logistic regression model with complete cases can capture enough information to converge to the real value of parameters. In such cases, the new method had similar performance comparing to traditional or penalized method. In the simulation part, we also tested the robustness of the new method to the assumption of the censoring mechanism that the probability of censoring is independent with the value of censored variable, the results showed that the new method is more accurate for estimating coefficients when the censoring probability depend on the value of censored variable.

Another important advantage of our new method is that it can deal with sparse datasets better. In the case of sample size is relatively small comparing to the total number of combination of two covariates, sparse table happened frequently in simulation datasets. For example, in our simulation with sample size 100 and censoring probability 30%, almost 20% of the replicates contain rare observations for some certain combination of complete and censored covariates, in which case that the traditional logistic regression model had non-convergence in estimates of coefficients. While with the new methods, the rate of non-convergence is similar to the rate with penalized likelihood method (Firth), which has been showed performed well in sparse

dataset by Heinze and Schemper (2002).

In the application part, we applied our new method to a study of relationship between the presence of arteriolosclerosis and time in MCI. Our results showed that it is more likely a patient had presence arteriolosclerosis with longer time stay in MCI.

Chapter 5 Future Research

In this dissertation, we proposed a modified goodness-of-fit test for multinomial logistic regression model with clustering effect. We showed the modified test statistic can preserve the type-I error better than traditional statistic for longitudinal data. We also studied the power of the new statistic by simulation. Different alternative hypothesis can be used in power analysis, and we studied the case of missing squared term in this dissertation. Missing of interaction and other more complex alternatives can also be tested in the future.

In the second chapter, we applied SIS-MCMC algorithm to multinomial logistic regression model with two categorical covariates. This method combined the advantages of both sequential importance sampling and MCMC. However, we can only deal with the two-way contingency table as the Markov bases is large and infeasible for three-way tables. Also, it is not clear how to involve the third covariate, especially when that variable is continuous as the age in Nun study. We found a satisfactory solution by stratifying the data using age intervals.

Some extensions and generalizations can be applied on our new method of logistic regression model with censored covariate. Firstly, our method is aimed at logistic regression model with right censored covariates. A obvious extension would be dealing with logistic regression model with left censored covariates or interval censored covariates. Different setting of Z_2 (as defined in equation 4.3) should be applied based on different censoring types.

Another potential generalization is that our likelihood function based joint distribution can only deal with one censored covariate in this chapter. It can be extended to datasets with more than one censored covariates or even mixture of censoring and missing covariates. We also only include one complete categorical covariate here, it

is of interest to have more than one complete covariate in the problem.

Appendix

R Code for Goodness-of-fit Test

```
library(ResourceSelection)
library(MASS)
library(mnlogit)
#### Multinomial Case
MultiGOF = function(B, pat, df, X2tu, X1v, mu, rho, K, beta1, beta2, beta3, betas){
Result = matrix(NA,B,7)
for (b in 1:B){
X3 = rep(1, pat)  # for the intercept
#X2 = c(1:pat)/X2tu  # cluster cov
X2 = sample(c(1:9), pat, replace=TRUE)/X2tu
X = NA
ID = 0
for (i in 1:pat){
k = sample(c(1,2,3,4,5),1)  # number of obs in each patient
sigma = matrix(rho, k, k)
diag(sigma) = X1v
X1 = mvrnorm(1, rep(mu, k), sigma)
temp = cbind(X1, X2[i], X3[i])
ID = c(ID, rep(i, k))
X = rbind(X, temp)
}
X = X[-1,]
ID = ID[-1]

beta1s = c(beta1, betas[1])
beta2s = c(beta2, betas[2])
beta3s = c(beta3, betas[3])
X_sq = cbind(X, (X[,1])^2)

# Generate Y
eta = cbind(X_sq%*%beta1s, X_sq%*%beta2s, X_sq%*%beta3s)
eeta = exp(eta)
P = eeta/apply(eeta, 1, sum)

Y = P[,1]
for (i in 1:nrow(P)){
r = runif(1)
if(r<P[i,1]){Y[i] = 1}
else if (r<P[i,1]+P[i,2]){Y[i]=2}
else {Y[i]=3}
}
#table(Y)
Y = as.matrix(Y)
X0 = X
X = X_sq
```

```
# fit the model
simu = cbind(Y,X)
simu = cbind(simu, c(1:nrow(simu)))
ss1 = as.matrix(rep(1,K))
s1 = kronecker(simu, ss1)
ss = as.matrix(rep(c(1:K), nrow(X)))
s1 = cbind(s1, ss)
s11 = as.matrix(s1[,1] == s1[, ncol(s1)])
simu = cbind(s1, s11)
colnames(simu) = c("Y","X1","X2","X3","X1s","index","choice","mode")
simu = as.data.frame(simu)
simuraw = as.data.frame(cbind(Y,X))
colnames(simuraw) = c("Y","X1","X2","X3","X1s")
simuraw = cbind(simuraw, c(1:nrow(simuraw)))
colnames(simuraw)[ncol(simuraw)] = "index"
simuraw = cbind(simuraw,m1$fitted.values)
colnames(simuraw)[ncol(simuraw)] = "fit"
simu = cbind(simu, as.vector(t(m1$probabilities)))
colnames(simu)[ncol(simu)] = "fit"
### Test statistic ###
## Ordered by p1
yhat=subset(simu, choice==1)$fit #grouped by p1
G = 10
cutyhat = cut(yhat, breaks = quantile(yhat, probs=seq(0,1, 1/G)), include
    .lowest=TRUE)
Egroup = list()
Ogroup = split(subset(simu, mode==1), cutyhat)
O=E=ph=array(NA, dim=c(G,K))
for (k in (1:K)){
Egroup[[k]] = split(subset(simu, choice==k), cutyhat)
for (j in (1:G)){
O[j,k] = sum(subset(Ogroup[[j]], choice==k)$mode)
E[j,k] = sum(subset(Egroup[[k]][[j]])$fit)
ph[j,k] = sum(subset(Egroup[[k]][[j]])$fit -(subset(Egroup[[k]][[j]])$
    fit)^2)
ng = nrow(Ogroup[[j]])
pbargk = E[j,k]/ng
ph[j,k] = ph[j,k]/(ng*pbargk*(1-pbargk))
}
}
C1 = sum((O-E)^2/E)
J1 = sum((O-E)^2/E/ph)
pc1 = 1-pchisq(C1, df)
pj1 = 1-pchisq(J1, df)
##### Include the cluster effect here ####
# vi from simple method
simuraw = cbind(simuraw, ID)
colnames(simuraw)[ncol(simuraw)] = "ID"
m = rep(NA,K)
p = m
n = m
x = m
r = rep(0,K)
for (i in 1:K){
```

```
gg = subset(simuraw, Y==i)
m[i] = length(unique(simuraw$ID))
n[i] = nrow(simuraw)
x[i] = nrow(gg)
p[i] = x[i]/n[i]
for (cl in unique(simuraw$ID)){
temp = subset(simuraw, ID==cl)
nij = nrow(temp)
tempx = subset(temp, Y==i)
xij = nrow(tempx)
r[i] = r[i] + (xij-nij*p[i])^2
}
}
v = m/(m-1)*r/n/n
po = rep(0,K)
for (i in 1:K){
temp = subset(simu, choice==i)
po[i] = mean(temp$fit)
}

delta1 = sum(v/po)*n[1]/(K-1)
Cc1 = C1/delta1
Jc1 = J1/delta1
pcc1 = 1-pchisq(Cc1, df)
pjc1 = 1-pchisq(Jc1, df)
Result[b,] = c(pc1, pj1, pcc1, pjc1, delta1, C1, J1)
}
```

R Code SIS-MCMC algorithm

```
library(iterpc)
library(plyr)
library(ggplot2)
library(parallel)
######################  Move  ############################
# moveAB in section 4
moveAB = function(J,K){
check = 0 # to make sure not return all 0s
while(check==0){
move1 = matrix(0,J,K)
j = rep(0,4)
while(j[4]<1||j[4]>J){
j = sample(J,3,replace=T)
j = c(j,(j[3]+j[2]-j[1]))
}
k = rep(0,4)
while(k[4]<1||k[4]>K){
k = sample(K,3,replace=T)
k = c(k,(k[3]+k[2]-k[1]))
}
e1 = move1
e1[j[1],k[1]] = 1
e2 = move1
e2[j[2],k[2]] = 1
```

```
e3 = move1
e3[j[3],k[3]] = 1
e4 = move1
e4[j[4],k[4]] = 1
move1 = e1-e2-e3+e4
move2 = -move1
move = list(move1,move2)
check = sum((unlist(move))^2)
}
return(move)
}
# move0 in section 2
move0 = function(J){
move = matrix(0,2,J)
j = sort(sample(c(2:(J-1)),2))
j = c(j[1]-1,j[1],j[2],j[2]+1)
sig = sample(c(1,-1),1)
move[,j[1]] = c(1,-1)
move[,j[2]] = -c(1,-1)
move[,j[3]] = -c(1,-1)
move[,j[4]] = c(1,-1)
move = sig*move
return(move)
}
# moveA in section 2
moveA = function(J){
move = matrix(0,2,J)
j = c(1,2)
while(j[2]-j[1]==1){
j = sort(sample(c(1:J),2))
}
j2 = sample(c(c((j[1]+1):(j[2]-1)),c((j[1]+1):(j[2]-1))),1)
j3 = j[1]+j[2]-j2
j = c(j[1],min(j2,j3),max(j2,j3),j[2])
sig = sample(c(1,-1),1)
move[,j[1]] = c(1,-1)
move[,j[2]] = -c(1,-1)
move[,j[3]] = -c(1,-1)
move[,j[4]] = c(1,-1)
move = sig*move
return(move)
}

########## MCMC Part ###############
# Hypergeometric probability
Prob2 = function(X){ # X is a list with 2 matrix
p = 1
J = nrow(X[[1]])
K = ncol(X[[1]])
for (i in 1:J){ # Seperate by row??
Xt = rbind(X[[1]][i,],X[[2]][i,])
c = apply(Xt,2,sum)
temp = 1
for (j in 1:K){
```

```
temp = temp*choose(c[j],Xt[1,j])
}
p = p*(temp/choose(sum(c),sum(Xt[1,])))
}
return(as.numeric(p))
}

Problog2 = function(X){
#X is a list with 2 matrix
J = nrow(X[[1]])
K = ncol(X[[1]])
xi = c(sum(X[[1]]),sum(X[[2]]))
xj = rep(0,J)
xk = rep(0,K)
for (i in 1:2){
xj = xj + apply(X[[i]],1,sum)
xk = xk + apply(X[[i]],2,sum)
}
logv = -2*sum(log(c(1:sum(xi))))
X2 = X
for (i in 1:2){
X2[[i]][X2[[i]]==0] = 1
logv = logv + sum(log(c(1:xi[i])))
}
for (j in 1:J){
logv = logv + sum(log(c(1:xj[j])))
}
for (k in 1:K){
logv = logv + sum(log(c(1:xk[k])))
}
for (i in 1:2){
for (j in 1:J){
for (k in 1:K){
logv = logv - sum(log(c(1:X2[[i]][j,k])))
}
}
}
return(logv)
}

Problog25 = function(X){
#X is a list with 2 matrix
J = nrow(X[[1]])
K = ncol(X[[1]])
logv=0
for (i in 1:2){
for (j in 1:J){
for (k in 1:K){
if(X[[i]][j,k]==0){logv = logv
}else{logv = logv + sum(log(c(1:X[[i]][j,k])))}
}
}
}
return(logv)
```

```
}

# Hypergeometric probability
Prob = function(X){
col = apply(X,2,sum)
row = apply(X,1,sum)
return(prod(factorial(col))*prod(factorial(row))/factorial(sum(X))/prod(
    factorial(X)))
}

Problog = function(X){
col = apply(X,2,sum)
X2 = rbind(X, col)
row = apply(X2,1,sum)
X2 = cbind(X2,row)
X2[X2 == 0] = 1
logv = -sum(log(c(1:sum(X))))
for (i in 1:nrow(X)){
for (j in 1:ncol(X)){
logv = logv-sum(log(c(1:X2[i,j])))
}
}
for (i in 1:nrow(X)){logv = logv + sum(log(c(1:X2[i,ncol(X2)])))}
for (j in 1:ncol(X)){logv = logv + sum(log(c(1:X2[nrow(X2),j])))}
return(logv)
}

# Metroplis Hastings algorithm for 2 covariate
MH2 = function(X0,Burn=1000,S=1000,block=100){
J = nrow(X0[[1]])
K = ncol(X0[[1]])
X = X0
Xc = X
# Burn in
for (i in 1:Burn){
movem = moveAB(J,K)
Xc[[1]] = X[[1]] + movem[[1]]
Xc[[2]] = X[[2]] + movem[[2]]
u = runif(1,0,1)
if((max(abs(Xc[[1]])-Xc[[1]])>0)||(max(abs(Xc[[2]])-Xc[[2]])>0)){X = X}
else if(u<exp(Problog2(Xc)-Problog2(X))){X = Xc}
}
# Sample
Sample = list()
for (j in c(1:(S*block))){
movem = moveAB(J,K)
Xc[[1]] = X[[1]] + movem[[1]]
Xc[[2]] = X[[2]] + movem[[2]]
u = runif(1,0,1)
if((max(abs(Xc[[1]])-Xc[[1]])>0)||(max(abs(Xc[[2]])-Xc[[2]])>0)){X = X}
else if(u<exp(Problog2(Xc)-Problog2(X))){X = Xc}
Sample[[j]] = X
#if(is.integer(j/block)){Sample[[j/block]] = X}
}
```

```
Result = Sample[seq(from=block, to=S*block, by=block)]
return(Result)
}

# Metroplis Hastings algorithm for 1 covariate
MH = function(X0, Burn=1000, S=1000, block=100){
J = ncol(X0)
X = X0
# Burn in
for (i in 1:Burn){
movem = move(J)
Xc = X + movem
u = runif(1,0,1)
#print(list(X, Xc, u, i))
if(max(abs(Xc)-Xc)>0){X = X}
else if (u<exp(Problog(Xc)-Problog(X))){X = Xc}
}
Sample = list()
for (j in c(1:(S*block))){
movem = move(J)
Xc = X + movem
u = runif(1,0,1)
if(max(abs(Xc)-Xc)>0){X = X}
else if (u<exp(Problog(Xc)-Problog(X))){X = Xc}
Sample[[j]] = X
#if(is.integer(j/block)) {Sample[[j/block]] = X}
}
Result = Sample[seq(from=block, to=S*block, by=block)]
return(Result)
}

MH5 = function(X, Burn, S, block, N=5){
## Function that generate MCMC results for 5 choice contingency table 2
    covariates
## Need use function Problog25
Problog25 = function(X){
#X is a list with 2 matrix
J = nrow(X[[1]])
K = ncol(X[[1]])
logv=0
for (i in 1:2){
for (j in 1:J){
for (k in 1:K){
if(X[[i]][j,k]==0){logv = logv
}else{logv = logv + sum(log(c(1:X[[i]][j,k])))}
}
}
}
return(logv)
}

# Burn in
for (i in 1:Burn){
Xn = sample(N,2)
```

```
Xs2 = list(X[[Xn[1]]],X[[Xn[[2]]]])
J = nrow(Xs2[[1]])
K = ncol(Xs2[[1]])
movem = moveAB(J,K)
Xc = list(X[[1]],X[[2]])
Xc[[1]] = Xs2[[1]] + movem[[1]]
Xc[[2]] = Xs2[[2]] + movem[[2]]
u = runif(1,0,1)
 if((max(abs(Xc[[1]])-Xc[[1]])>0)||(max(abs(Xc[[2]])-Xc[[2]])>0)){X = X
}else if(u<exp(Problog25(Xs2)-Problog25(Xc))){
X[[Xn[1]]] = Xc[[1]]
X[[Xn[2]]] = Xc[[2]]
#print(c(i,exp(Problog25(Xs2)-Problog25(Xc))))
}
}
# Sample
Sample = list()
 for (j in c(1:(S*block))){
Xn = sample(N,2)
Xs2 = list(X[[Xn[1]]],X[[Xn[[2]]]])
J = nrow(Xs2[[1]])
K = ncol(Xs2[[1]])
movem = moveAB(J,K)
Xc = list(X[[1]],X[[2]])
Xc[[1]] = Xs2[[1]] + movem[[1]]
Xc[[2]] = Xs2[[2]] + movem[[2]]
u = runif(1,0,1)
 if((max(abs(Xc[[1]])-Xc[[1]])>0)||(max(abs(Xc[[2]])-Xc[[2]])>0)){X = X
}else if(u<exp(Problog25(Xs2)-Problog25(Xc))){
X[[Xn[1]]] = Xc[[1]]
X[[Xn[2]]] = Xc[[2]]
#print(c(i,exp(Problog25(Xs2)-Problog25(Xc))))
}
 Sample[[j]] = X
}
 Result = Sample[seq(from=block,to=S*block,by=block)]
 return(Result)
}

########## From Sample A, calculate La Lb ##################
LaLb = function(A,J=8,K=7)
La = rep(0,length(A))
Lb=La
Lanova = La
 for (i in 1:length(A)){
datat = A[[i]]
s = datat[[1]]
a = datat[[1]]+datat[[2]]
coro = c(0,0,0)
 for(j in 1:J){
 for(k in 1:K){
temp = matrix(c(0,j,k),a[j,k],3,byrow=T)
 if(s[j,k]>0) {temp[1:s[j,k],1]=1}
coro = rbind(coro,temp)
```

```
}
}
#print(i)
colnames(coro) = c("case","J","K")
coro2 = as.data.frame(coro)
logitJK <- glm(case ~ J + K, data = coro2, family = "binomial")
logitJ <- glm(case ~ J, data = coro2, family = "binomial")
logitK <- glm(case ~ K, data = coro2, family = "binomial")
logitanova <- glm(case ~ factor(J) + factor(K) , data = coro2, family =
    "binomial")

Lb[i] = as.numeric(-2*(logLik(logitJ)-logLik(logitJK)))
La[i] = as.numeric(-2*(logLik(logitK)-logLik(logitJK)))
Lanova[i] = as.numeric(-2*(logLik(logitJK)-logLik(logitanova)))
}
return(list(La,Lb,Lanova))
}

# From one covariate marginal to get two covariate
# function for n balls k bins choose
library(plyr)
nkballs <- function(n,k) {
tmp <- sample(k, n, replace=TRUE)
nka = tabulate(tmp)
if (length(nka)<k){nka = c(nka, rep(0,k-length(nka)))}
return(nka)
}
# this one can make sure at least 1 in each category
nkballs2 <- function(n,k) {
tmp <- sample(k, n-k, replace=TRUE)
nka = tabulate(tmp)
if (length(nka)<k){nka = c(nka, rep(0,k-length(nka)))}
return(nka+1)
}

# one question is if each attempts > 0, if so
# the failure matrix should use nkballs2
marg2table = function(aa,kk){
#kk is 7 for beta, 8 for alpha
A = list()
tt = list()
for (i in 1:length(aa)){
if (kk == 7) {
tt[[1]] = t(sapply(aa[[i]][1,], nkballs, k=kk))
tt[[2]] = t(sapply(aa[[i]][2,], nkballs2, k=kk))
A[[i]] = tt
} else {
tt[[1]] = sapply(aa[[i]][1,], nkballs, k=kk)
tt[[2]] = sapply(aa[[i]][2,], nkballs2, k=kk)
A[[i]] = tt
}
}
return(A)
}
```

```
########## NUN Part ###############
NUN_MCMC = function(X0,A0,b0,sn,I0=2,J0=3,N0=5,Burn=2,S=100,block=1000){
source("SISfunction.r")
source("MCMCfunction.r")
#sn number of starting points
# X0 A0 b0 are given data
# I0 J0 N0 are given dimension
startm = matrix(NA,nrow(X0),sn)
i = 1
while (i<=sn){
startm[,i] = SIS(X0,A0,b0)
if (sum((A0%*%startm[,i]-b0)^2)==0) {
i=i+1
}
}
MCout = list()
for (c in 1:sn){
C0 = startm[,c] #starting column
Xstart = list()
ind = 1
for(k in 1:N0){
Xstart[[k]] = matrix(NA,I0,J0)
for (i in 1:I0){
for (j in 1:J0){
Xstart[[k]][i,j]=C0[ind]
ind = ind+1
}
}
}
MCout[[c]] = MH5(X=Xstart,Burn=Burn,S=S,block=block,N=N0)
}
return(MCout)
}

#Transfer matrix to real data
trans = function(temp,sn,sp,N0,I0=2,J0=3){
# sn is the index of starting point
# S is the index of sample
# Transfer a list of 5 matrix to long format logistic data
t = c(NA,NA,NA)
trans0 = temp[[sn]][[sp]]
for(n in 1:N0){
for(i in 1:I0){
for(j in 1:J0){
if(trans0[[n]][i,j]>0){
tt = matrix(c(n,i,j),trans0[[n]][i,j],3,byrow=T)
t = rbind(t,tt)
}
}
}
}
mcmc_sample=t[-1,]

a = as.matrix(rep(1,N0))
```

```
n1 = kronecker(as.matrix(mcmc_sample),a)
aa = as.matrix(rep(c(1:N0),nrow(n1)/N0))
n1 = cbind(n1,aa)
n11 = as.matrix(n1[,1] == n1[,4])
n1 = cbind(n1,n11)
colnames(n1) = c("case","J","K","choice","mode")
mcmc_sample2 = as.data.frame(n1)
return(mcmc_sample2)
}

HLfunction = function(temp,sn,S,I,J,K){
fmjk <- formula(mode ~ 1|factor(J)+factor(K))
HL=matrix(NA,sn,S)
for(p in 1:sn){
for(q in 1:S){
simu = trans(temp,p,q,N0=K) ###!!Changed for N0=4!!!!!!!!
simuraw = subset(simu,mode==1)
fm1 = formula(mode ~ 1|factor(J)+factor(K))
m1 = mnlogit(fm1,simu,"choice")

simuraw = cbind(simuraw,m1$fitted.values)
colnames(simuraw)[ncol(simuraw)] = "fit"
simu = cbind(simu,as.vector(t(m1$probabilities)))
colnames(simu)[ncol(simu)] = "fit"
G = J*I
Egroup = list()
simuraw$fac =with(simuraw,interaction(factor(J),factor(K)),drop=TRUE)
Ogroup = split(simuraw,simuraw$fac)
O=E=ph=array(NA,dim=c(G,K))
for (k in (1:K)){
simuk = subset(simu,choice==k)
simuk$fac =with(simuk,interaction(factor(J),factor(K)),drop=TRUE)
Egroup[[k]] = split(simuk,simuk$fac)
for (j in (1:G)){
O[j,k] = sum(subset(Ogroup[[j]],choice==k)$mode)
E[j,k] = sum(subset(Egroup[[k]][[j]])$fit)
ph[j,k] = sum(subset(Egroup[[k]][[j]])$fit -(subset(Egroup[[k]][[j]])$
    fit)^2)
ng = nrow(Ogroup[[j]])
pbargk = E[j,k]/ng
ph[j,k] = ph[j,k]/(ng*pbargk*(1-pbargk))
}
}
C1 = sum((O-E)^2/(E+1e-8))
HL[p,q] = C1
}
}
return(HL)
}

HLPfunction = function(temp,sn,S,I,J,K){
# Calculate Pigeon's statistic
fmjk <- formula(mode ~ 1|factor(J)+factor(K))
Sys.time()
```

```
HL=matrix(NA,sn,S)
for(p in 1:sn){
for(q in 1:S){
simu = trans(temp,p,q,N0=K) ###!!Changed for N0=4!!!!!!!!
simuraw = subset(simu,mode==1)
fm1 = formula(mode ~ 1|factor(J)+factor(K))
m1 = mnlogit(fm1,simu,"choice")

simuraw = cbind(simuraw,m1$fitted.values)
colnames(simuraw)[ncol(simuraw)] = "fit"
simu = cbind(simu,as.vector(t(m1$probabilities)))
colnames(simu)[ncol(simu)] = "fit"
G = J*I
Egroup = list()
simuraw$fac =with(simuraw,interaction(factor(J),factor(K)),drop=TRUE)
Ogroup = split(simuraw,simuraw$fac)
O=E=ph=array(NA,dim=c(G,K))
for (k in (1:K)){
simuk = subset(simu,choice==k)
simuk$fac =with(simuk,interaction(factor(J),factor(K)),drop=TRUE)
Egroup[[k]] = split(simuk,simuk$fac)
for (j in (1:G)){
O[j,k] = sum(subset(Ogroup[[j]],choice==k)$mode)
E[j,k] = sum(subset(Egroup[[k]][[j]])$fit)
ph[j,k] = sum(subset(Egroup[[k]][[j]])$fit -(subset(Egroup[[k]][[j]])$
    fit)^2)
ng = nrow(Ogroup[[j]])
pbargk = E[j,k]/ng
ph[j,k] = ph[j,k]/(ng*pbargk*(1-pbargk))
}
}
C1 = sum((O-E)^2/(E+1e-8))
J1 = sum((O-E)^2/(E+1e-8)/(ph+1e-8))
HL[p,q] = J1
}
}
return(HL)
}

Pvaluefunction = function(HL,temp,I,J,K,sn,S){
# Pvalue for HL's statistic
library(mnlogit)
simuraw = list()
for (i in 1:K){simuraw[[i]] = t(x1[[i]])}
# make sure x1 is the original data
temp2 = temp
temp2[[1]][[1]] = simuraw
simu = trans(temp2,1,1,N0=K)
simuraw = subset(simu,mode==1)
fm1 = formula(mode ~ 1|factor(J)+factor(K))
m1 = mnlogit(fm1,simu,"choice")
simuraw = cbind(simuraw,m1$fitted.values)
colnames(simuraw)[ncol(simuraw)] = "fit"
simu = cbind(simu,as.vector(t(m1$probabilities)))
```

```
colnames(simu)[ncol(simu)] = "fit"
G = J*I
Egroup = list()
simuraw$fac =with(simuraw, interaction(factor(J), factor(K)), drop=TRUE)
Ogroup = split(simuraw, simuraw$fac)
O=E=ph=array(NA, dim=c(G,K))
for (k in (1:K)){
simuk = subset(simu, choice==k)
simuk$fac =with(simuk, interaction(factor(J), factor(K)), drop=TRUE)
Egroup[[k]] = split(simuk, simuk$fac)
for (j in (1:G)){
O[j,k] = sum(subset(Ogroup[[j]], choice==k)$mode)
E[j,k] = sum(subset(Egroup[[k]][[j]])$fit)
ph[j,k] = sum(subset(Egroup[[k]][[j]])$fit -(subset(Egroup[[k]][[j]])$
    fit)^2)
ng = nrow(Ogroup[[j]])
pbargk = E[j,k]/ng
ph[j,k] = ph[j,k]/(ng*pbargk*(1-pbargk))
}
}
C1 = sum((O-E)^2/(E+1e-8))
Pvalue = sum(HL>=C1)/sn/S
#Cc1 = sum((O-E)^2/(E+1e-8)/ph)
return(c(Pvalue,C1))
}

PvaluePfunction = function(HLP, temp, I, J, K, sn, S){
# Pvalue for Pigeon's statistic
library(mnlogit)
simuraw = list()
for (i in 1:K){simuraw[[i]] = t(x1[[i]])}
temp2 = temp
temp2[[1]][[1]] = simuraw
simu = trans(temp2, 1, 1, N0=K)
simuraw = subset(simu, mode==1)
fm1 = formula(mode ~ 1|factor(J)+factor(K))
m1 = mnlogit(fm1, simu, "choice")
simuraw = cbind(simuraw, m1$fitted.values)
colnames(simuraw)[ncol(simuraw)] = "fit"
simu = cbind(simu, as.vector(t(m1$probabilities)))
colnames(simu)[ncol(simu)] = "fit"
G = J*I
Egroup = list()
simuraw$fac =with(simuraw, interaction(factor(J), factor(K)), drop=TRUE)
Ogroup = split(simuraw, simuraw$fac)
O=E=ph=array(NA, dim=c(G,K))
for (k in (1:K)){
simuk = subset(simu, choice==k)
simuk$fac =with(simuk, interaction(factor(J), factor(K)), drop=TRUE)
Egroup[[k]] = split(simuk, simuk$fac)
for (j in (1:G)){
O[j,k] = sum(subset(Ogroup[[j]], choice==k)$mode)
E[j,k] = sum(subset(Egroup[[k]][[j]])$fit)
ph[j,k] = sum(subset(Egroup[[k]][[j]])$fit -(subset(Egroup[[k]][[j]])$
```

```
        fit)^2)
ng = nrow(Ogroup[[j]])
pbargk = E[j,k]/ng
ph[j,k] = ph[j,k]/(ng*pbargk*(1-pbargk))
}
}
C1 = sum((O-E)^2/(E+1e-8))
J1 = sum((O-E)^2/(E+1e-8)/(ph+1e-8))
Pvalue = sum(HLP>=J1)/sn/S
return(c(Pvalue,J1))
}

LRTfunction = function(temp,sn,S,I,J,K){
fmjk <- formula(mode ~ 1|factor(J)+factor(K))
LRT=matrix(NA,sn,S)
for(p in 1:sn){
for(q in 1:S){
simu = trans(temp,p,q,N0=K) ###!!Changed for N0=4!!!!!!!!
simuraw = subset(simu,mode==1)
fm1 = formula(mode ~ 1|factor(J)+factor(K))
m1 = mnlogit(fm1,simu,"choice")

simuraw = cbind(simuraw,m1$fitted.values)
colnames(simuraw)[ncol(simuraw)] = "fit"
simu = cbind(simu,as.vector(t(m1$probabilities)))
colnames(simu)[ncol(simu)] = "fit"

simuraw$fac =with(simuraw,interaction(factor(J),factor(K)),drop=TRUE)
Ogroup = split(simuraw,simuraw$fac)
N = matrix(NA,length(Ogroup),1)
NY = matrix(NA,length(Ogroup),K)
for (i in 1:length(Ogroup)){
for (j in 1:K){
N[i] = nrow(Ogroup[[i]])
NY[i,j] = sum(Ogroup[[i]]$choice==j)
}
}
l2 = 0
for (i in 1:length(Ogroup)){
for (j in 1:K){
if (NY[i,j]!=0){
l2 = l2+NY[i,j]*log(NY[i,j]/N[i])
}
}
}
l1 = m1$logLik
LRT[p,q] = -2*(l1-l2)
}
}
return(LRT)
}

PLRTfunction = function(LRT,temp,I,J,K,sn,S){
# Pvalue for LRT's statistic
```

```
library(mnlogit)
simuraw = list()
for (i in 1:K){simuraw[[i]] = t(x1[[i]])}
temp2 = temp
temp2[[1]][[1]] = simuraw
simu = trans(temp2,1,1,N0=K)
simuraw = subset(simu,mode==1)
fm1 = formula(mode ~ 1|factor(J)+factor(K))
m1 = mnlogit(fm1,simu,"choice")
simuraw = cbind(simuraw,m1$fitted.values)
colnames(simuraw)[ncol(simuraw)] = "fit"
simu = cbind(simu,as.vector(t(m1$probabilities)))
colnames(simu)[ncol(simu)] = "fit"
simuraw$fac =with(simuraw,interaction(factor(J),factor(K)),drop=TRUE)
Ogroup = split(simuraw,simuraw$fac)
N = matrix(NA,length(Ogroup),1)
NY = matrix(NA,length(Ogroup),K)
for (i in 1:length(Ogroup)){
for (j in 1:K){
N[i] = nrow(Ogroup[[i]])
NY[i,j] = sum(Ogroup[[i]]$choice==j)
}
}
l2 = 0
for (i in 1:length(Ogroup)){
for (j in 1:K){
if (NY[i,j]!=0){
l2 = l2+NY[i,j]*log(NY[i,j]/N[i])
}
}
}
l1 = m1$logLik
LRT0 = -2*(l1-l2)
Pvalue = sum(LRT>=LRT0)/sn/S
return(c(Pvalue,LRT0))
}
```

SAS Code for Logistic Regression with Censored Covariate

```
%macro llnlpsimu(num,x1=,x2=,y=,covtype=,ww11=,ww21=,ww12=,ww22=,);
proc nlp data=simu&num vardef=df phes cov=&covtype out=simures&num;
max ll;
parms beta0=0,beta12=0, beta22=0, beta23=0,w11=&ww11,w21=&ww21,w12=&
    ww12,w22=&ww22;
/*p11=0.9, p21=0.73, p12=0.06, p22=0.14 */
*bounds p11>1e-12, p12>1e-12, p21>1e-12, p22>1e-12, p11<1, p12
    <1, p21<1, p22<1;
/* p11=0.9;p21=0.73;p12=0.06;p22=0.14;*/
p11=exp(w11)/(1+exp(w11));
p12=exp(w12)/(1+exp(w12));
p21=exp(w21)/(1+exp(w21));
p22=exp(w22)/(1+exp(w22));
if &x2<4 then
do;
```

```
14 if &x1=0 and &x2=1 then
15 do;
16 mu=1/(1+exp(-beta0 -0-0));
17 y = mu**&Y*(1-mu)**(1-&Y)*p11;
18 end;
19 else if &x1=1 and &x2=1 then
20 do;
21 mu=1/(1+exp(-beta0-beta12 -0));
22 y = mu**&Y*(1-mu)**(1-&Y)*p21;
23 end;
24 else if &x1=0 and &x2=2 then
25 do;
26 mu=1/(1+exp(-beta0 -0-beta22));
27 y = mu**&Y*(1-mu)**(1-&Y)*p12;
28 end;
29 else if &x1=1 and &x2=2 then
30 do;
31 mu=1/(1+exp(-beta0-beta12-beta22));
32 y = mu**&Y*(1-mu)**(1-&Y)*p22;
33 end;
34 else if &x1=0 and &x2=3 then
35 do;
36 mu=1/(1+exp(-beta0 -0-beta23));
37 y = mu**&Y*(1-mu)**(1-&Y)*(1-p11-p12);
38 end;
39 else if &x1=1 and &x2=3 then
40 do;
41 mu=1/(1+exp(-beta0-beta12-beta23));
42 y = mu**&Y*(1-mu)**(1-&Y)*(1-p21-p22);
43 end;
44 end;
45 else if &x2=4 then
46 do;
47 if &x1=0 then
48 do;
49 mu1=1/(1+exp(-beta0 -0-0));
50 mu2=1/(1+exp(-beta0 -0-beta22));
51 mu3=1/(1+exp(-beta0 -0-beta23));
52 y = mu1**&Y*(1-mu1)**(1-&Y)*p11+mu2**&Y*(1-mu2)**(1-&Y)*p12+mu3**&Y*(1-
       mu3)**(1-&Y)*(1-p11-p12);
53 end;
54 else if &x1=1 then
55 do;
56 mu1=1/(1+exp(-beta0-beta12 -0));
57 mu2=1/(1+exp(-beta0-beta12-beta22));
58 mu3=1/(1+exp(-beta0-beta12-beta23));
59 y = mu1**&Y*(1-mu1)**(1-&Y)*p21+mu2**&Y*(1-mu2)**(1-&Y)*p22+mu3**&Y*(1-
       mu3)**(1-&Y)*(1-p21-p22);
60 end;
61 end;
62 else if &x2=5 then
63 do;
64 if &x1=0 then
65 do;
```

```
mu1=1/(1+exp(-beta0-0-0));
mu2=1/(1+exp(-beta0-0-beta22));
mu3=1/(1+exp(-beta0-0-beta23));
y = mu2**&Y*(1-mu2)**(1-&Y)*p12+mu3**&Y*(1-mu3)**(1-&Y)*(1-p11-p12);
end;
 else if &x1=1 then
do;
mu1=1/(1+exp(-beta0-beta12-0));
mu2=1/(1+exp(-beta0-beta12-beta22));
mu3=1/(1+exp(-beta0-beta12-beta23));
y = mu2**&Y*(1-mu2)**(1-&Y)*p22+mu3**&Y*(1-mu3)**(1-&Y)*(1-p21-p22);
end;
end;
 ;
 ll=log(y);
run;
%mend llnlpsimu;
/*Import data*/
%macro create(n);
%do i = 1 %to &n;
data simu&i;
set simu;
where rep = &i;
run;
data _null_;
set simu&i;
call symput("lps11",lp11);
call symput("lps21",lp21);
call symput("lps12",lp12);
call symput("lps22",lp22);
run;
%put mean of x is &lps11 &lps21 &lps12 &lps22 ;
%llnlpsimu(&i,x1=x1,x2=z2,y=y,covtype=5,ww11=&lps11,ww21=&lps21,ww12=&
      lps12,ww22=&lps22);
data out&i; set simures&i; if _obs_=1;run;
proc logistic data=simu&i descending outest=log&i;
class x2(ref="1") x1(ref="0")/ param=ref ;
model y = x2 x1;
where C=0;
run;
proc logistic data=simu&i descending outest=logf&i;
class x2(ref="1") x1(ref="0")/ param=ref ;
model y = x2 x1/firth;
where C=0;
run;
data out&i; set out&i; rep=&i; run;
data log&i; set log&i; rep=&i; run;
data logf&i; set logf&i; rep=&i; run;
%end;
%mend create;
%create(500)
```

Bibliography

A Mortimer, J. (2012). The nun study: risk factors for pathology and clinical-pathologic correlations. *Current Alzheimer Research*, 9(6):621–627.

Abner, E. L., Schmitt, F., Nelson, P., Lou, W., Wan, L., Gauriglia, R., Dodge, H., Woltjer, R., Yu, L., Bennet, D., et al. (2015). The statistical modeling of aging and risk of transition project: Data collection and harmonization across 11 longitudinal cohort studies of aging, cognition, and dementia. *Observational studies*, 1(2015):56.

Atem, F., Qian, J., Maye, J. E., Johnson, K. A., and Betensky, R. A. (2015). Linear regression with a randomly censored covariate: application to an alzheimer's study. Technical report, Technical report.

Austin, P. C. and Hoch, J. S. (2004). Estimating linear regression models in the presence of a censored independent variable. *Statistics in medicine*, 23(3):411–429.

Berkelaar, M., Eikland, K., Notebaert, P., et al. (2004). lpsolve: Open source (mixed-integer) linear programming system. *Eindhoven U. of Technology*.

Besag, J. and Clifford, P. (1989). Generalized monte carlo significance tests. *Biometrika*, 76:633–642.

Chen, Y., Diaconis, P., Holmes, S., and Liu, J. (2005a). Sequential monte carlo methods for statistical analysis of tables. *American Statistical Association*, 100(469):109–120.

Chen, Y., Diaconis, P., Holmes, S. P., and Liu, J. S. (2005b). Sequential monte carlo methods for statistical analysis of tables. *Journal of the American Statistical Association*, 100(469):109–120.

Chen, Y., Dinwoodie, I., Dobra, A., and Huber, M. (2005c). Lattice points, contingency tables, and sampling. In *Integer points in polyhedra—geometry, number theory, algebra, optimization*, volume 374 of *Contemp. Math.*, pages 65–78. Amer. Math. Soc., Providence, RI.

Chen, Y., Dinwoodie, I., and Sullivant, S. (2006a). Sequential importance sampling for multiway tables. *Ann. Statist.*, 34(1):523–545.

Chen, Y., Dinwoodie, I. H., and Sullivant, S. (2006b). Sequential importance sampling for multiway tables. *The Annals of Statistics*, 34:523–545.

D. Kahle, R. Y. and Garcia-Puente, L. (2015). Hybrid schemes for exact conditional inference in discrete exponential families. *Submitted to Annals of Institute of Statistical Mathematics*.

De Loera, J. and Onn, S. (2005). Markov bases of three-way tables are arbitrarily complicated. *J. Symb. Comput.*, 41(2):173–181.

De Loera, J. and Onn, S. (2005). Markov bases of three-way tables are arbitrarily complicated. *Journal of Symbolic Computation*, 41:173–181.

Diaconis, P. and Sturmfels, B. (1998). Algebraic algorithms for sampling from conditional distributions. *The Annals of Statistics*, 26(1):363–397.

Dobra, A. and Fienberg, S. (2010). The generalized shuttle algorithm. In Gibilisco, P., Riccomagno, E., Rogantin, M., and Wynn, H., editors, *Algebraic and geometric methods in statistics*, pages 135–156. Cambridge University Press.

Esiri, M., Wilcock, G., and Morris, J. (1997). Neuropathological assessment of the lesions of significance in vascular dementia. *Journal of Neurology, Neurosurgery & Psychiatry*, 63(6):749–753.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1):27–38.

Hara, H., Takemura, A., and Yoshida, R. (2010). On connectivity of fibers with positive marginals in multiple logistic regression. *Journal of Multivariate Analysis*, 101(4):909–925.

Heinze, G. and Schemper, M. (2002). A solution to the problem of separation in logistic regression. *Statistics in medicine*, 21(16):2409–2419.

Ighodaro, E. T., Abner, E. L., Fardo, D. W., Lin, A.-L., Katsumata, Y., Schmitt, F. A., Kryscio, R. J., Jicha, G. A., Neltner, J. H., Monsell, S. E., et al. (2016). Risk factors and global cognitive status related to brain arteriolosclerosis in elderly individuals. *Journal of Cerebral Blood Flow & Metabolism*, page 0271678X15621574.

Kumar, V., Abbas, A. K., and Aster, J. C. (2012). *Robbins basic pathology*. Elsevier Health Sciences.

Pantoni, L., Garcia, J. H., and Brown, G. G. (1996). Vascular pathology in three cases of progressive cognitive deterioration. *Journal of the neurological sciences*, 135(2):131–139.

Vach, W. and Blettner, M. (1995). Logistic regression with incompletely observed categorical covariatesinvestigating the sensitivity against violation of the missing at random assumption. *Statistics in Medicine*, 14(12):1315–1329.

Vach, W. and Schumacher, M. (1993). Logistic regression with incompletely observed categorical covariates: a comparison of three approaches. *Biometrika*, 80(2):353–362.

Vita

Education

- **Ph.D.**, Statistics, University of Kentucky, Lexington, KY 2011-present
- **M.S.**, Statistics, University of Kentucky, Lexington, KY 2011-2014
- **B.S.**, Statistics, Shandong University, Jinan, China 2007-2011

Experience

- **Research Assistant** University of Kentucky, 2014-present
- **Teaching Assistant** University of Kentucky, 2011-2014
- **Internship** Kentucky Department for Energy Development and Independence, 2012