NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

HYBRID SIS AND MARKOV CHAIN MONTE CARLO SAMPLING METHODOLOGY FOR GOODNESS-OF-FIT TESTS ON CONTINGENCY TABLES

by

Patrick M. Saluke

September 2018

Thesis Advisor:	Ruriko Yoshida
Second Reader:	W. Matthew Carlyle

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE September 2018	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE

HYBRID SIS AND MCMC SAMPLING METHODOLOGY FOR GOODNESS-OF-FIT TESTS ON CONTINGENCY TABLES

5. FUNDING NUMBERS

6. AUTHOR(S)

Patrick M. Saluke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

The views expressed in this document are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol Number: N/A.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Logistic regression is one of the most popular means of modeling contingency table data due to its ease of use. Simple asymptotic inference (like a χ^2 approximation) for evaluating goodness-of-fit tests, however, may not be valid for sparse datasets having cell counts less than 5. In these cases, we often attempt exact conditional inference via a sampler, such as Markov Chain Monte Carlo (MCMC) or Sequential Importance Sampling (SIS). This paper proposes a hybrid sampling scheme that combines MCMC and SIS to sample sparse, multidimensional contingency tables satisfying fixed marginals when MCMC alone does not guarantee an exhaustive sampling of the conditional state space. To investigate its suitability, the proposed hybrid scheme is applied to an observational dataset from Alzheimer's researcher JA Mortimer measuring the cognitive states of nuns over a 15 year period beginning in 1991. Through the application of our proposed scheme, we find the estimated p-values via a hybrid MCMC and SIS sampler are remarkably similar to the χ^2 asymptotic approximation p-values, even for sparse contingency tables.

14. SUBJECT TERMS

Markov Chain Monte Carlo, MCMC, Sequential Importance Sampling, SIS, sparse, multidimensional contingency table

15. NUMBER OF PAGES 111

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

THIS PAGE INTENTIONALLY LEFT BLANK

HYBRID SIS AND MCMC SAMPLING METHODOLOGY FOR GOODNESS-OF-FIT TESTS ON CONTINGENCY TABLES

Patrick M. Saluke
Lieutenant Commander, United States Navy
MBA, University of Chicago, 2012
B.S., University of Notre Dame, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS ANALYSIS

from the

NAVAL POSTGRADUATE SCHOOL
September 2018

Approved by: Ruriko Yoshida
Thesis Advisor

W. Matthew Carlyle
Second Reader

W. Matthew Carlyle
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Logistic regression is one of the most popular means of modeling contingency table data due to its ease of use. Simple asymptotic inference (like a χ^2 approximation) for evaluating goodness-of-fit tests, however, may not be valid for sparse datasets having cell counts less than 5. In these cases, we often attempt exact conditional inference via a sampler, such as Markov Chain Monte Carlo (MCMC) or Sequential Importance Sampling (SIS). This paper proposes a hybrid sampling scheme that combines MCMC and SIS to sample sparse, multidimensional contingency tables satisfying fixed marginals when MCMC alone does not guarantee an exhaustive sampling of the conditional state space. To investigate its suitability, the proposed hybrid scheme is applied to an observational dataset from Alzheimer's researcher JA Mortimer measuring the cognitive states of nuns over a 15 year period beginning in 1991. Through the application of our proposed scheme, we find the estimated p-values via a hybrid MCMC and SIS sampler are remarkably similar to the χ^2 asymptotic approximation p-values, even for sparse contingency tables.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

BKZ	block Korkin-Zolotarev
CBR	column based reduction
DTMC	discrete time Markov chain
GI	global impairment
IP	integer programming
LB	lower bound
LLL	Lenstra-Lenstra-Lovász
MCMC	Markov chain Monte Carlo
MCI	mild cognitive impairment
MH	Metropolis-Hastings
MLE	maximum likelihood estimator
NPS	Naval Postgraduate School
pmf	probability mass function
RHS	right hand side
SIS	sequential importance sampling
UB	upper bound

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

We begin with the problem of performing a goodness-of-fit test on a *sparse*, multidimensional contingency table. Due to the low cell counts of sparse tables, there is no guarantee on the validity of asymptotic inference. In multidimensionality cases, the curse of dimensionality typically precludes exact inference through enumeration of all contingency tables satisfying the sufficient statistic. To overcome these difficulties, we usually construct an exact conditional hypothesis test by sampling a large number of tables from the conditional state space in order to approximate the distribution of test statistics under H_0.

Markov chain Monte Carlo (MCMC) with the Metropolis-Hastings acceptance criteria provides a method of sampling contingency tables from the conditional state space such that the distribution of tables converges to the true distribution under H_0 by the law of large numbers. Without a Markov basis, however, there is no guarantee that all contingency tables are connected via an *ergodic* Markov chain and thus have a non-zero probability of being sampled via MCMC. To overcome this obstacle, we rely on sequential importance sampling (SIS) to independently sample tables from incongruent regions of the conditional state space. This paper proposes a hybrid scheme combining MCMC and SIS to sample sparse, multidimensional contingency tables satisfying fixed marginals when MCMC alone does not guarantee an exhaustive sampling of the state space.

First, many independent starting tables are sampled from the conditional state space via SIS. Every SIS table is used as an independent starting point from which to initiate MCMC. Each chain of contingency tables resulting from MCMC has the beginning discarded (burn-in) and the remaining tables thinned (thinning). The test statistic is calculated for each surviving table, and the distribution is approximated using the sampled set of test statistics. Our assumption is that by sampling many different SIS starting points from the conditional state space, we are able to sample representatively from all regions of the space even though all contingency tables may not be connected via MCMC. By running MCMC for a sufficient number of iterations, the sampled distribution will converge to the true distribution of test statistics.

To test our sampling scheme, we apply the proposed procedure to a four-dimensional contingency table dataset measuring the cognitive states of nuns. Through the application of our hybrid scheme, we find the estimated p-values via a hybrid MCMC and SIS sampler are similar to the χ^2 asymptotic approximation p-values, even for sparse contingency tables.

Although the proposed hybrid sampler results in similar p-values for goodness-of-fit hypothesis tests as the χ^2 asymptotic approximation, no assumptions are made regarding the distribution of test statistics. This methodology could also be employed to validate the use of a χ^2 approximation. By employing SIS sampling to start various MCMC chains, we also avoid any costly computation of a Markov basis, which many times is unfeasible to compute. The major tradeoff is that a large sample size requires significant amounts of computing power and time compared to an asymptotic distribution assumption. Our conclusions on the suitability of this algorithm are based on its application to one dataset, but we provide a framework that should work for datasets of increased dimensionality.

Acknowledgments

First and foremost, I would like to thank my advisor, Dr. Ruriko "Rudy" Yoshida for the innumerable hours spent discussing and debating every single concept. We probably spent more time on conjectures that got scrapped than ideas that actually made it into the paper. Thank you for your patience and willingness to help me bear the load throughout this process. Along with my junior officer submarine tour, this thesis was one of my toughest trials, but I like to think I gained some glimpses of the hidden universe along with all the pain. I could not have accomplished this without you!

I would also like to thank our department chairman and my second reader, Dr. Matt Carlyle. I appreciate your insights, especially setting the stage for the structure and the technical details that followed.

Through their interaction with Dr. Yoshida, the following helped to make this research possible: Dr. Dick Kryscio and Dr. Zhiheng Xie at the University of Kentucky for their work with this dataset; Dr. David Kahle at the University of Baylor, Dr. Christopher O'Neill at UC Davis, and Dr. Luis Garcia-Puente at Sam Houston State University for their efforts on Macaulay2 and implementing RKZ and LLL column based reduction; and especially Hara san, Dr. Hisayuki Hara at Doshisha University, particularly for his R code for a random multidimensional move satisfying the sufficient statistic.

Since this will likely be my only publication, I would finally like to acknowledge the support of all those I have relied on for unfailing help. Thank you to the Lord for all the blessings, my lovely wife, Janice, my parents, sisters, and brothers, friends and family on the journey, the mentors throughout my naval career, my Navy Human Resources colleagues, the faculty here at NPS, my classmates in the OR department that shared many a commiseration, and the academics, researchers, and scientists that advance the knowledge of our world and made my small contribution possible.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: Introduction

This chapter introduces the background, motivation, and objectives for our analysis. A more exhaustive account of the proposed method and its application to an observed dataset will be provided in subsequent chapters.

1.1 Background

In surveys, medical, and scientific research, investigators frequently use contingency tables to study relationships between the dependent variables and independent response variable (Kateri 2014; Yoshida 2010). The simplest contingency tables, two-way tables, have two dimensions for two variables: one dependent variable, and one independent variable. Two-way contingency tables have been studied for over a century (Fam 2012), and hypothesis tests of two-way tables are often trivial. In this thesis, we attempt to extend hypothesis tests to less tractable contingency tables that contain multiple independent variables.

When analyzing the frequencies within an observed multidimensional contingency table, we often would like to conduct a hypothesis test comparing the relative fit of two models to the observed data. Does adding an additional explanatory factor (or level of complication) significantly improve the model? One of the procedures for model selection is a goodness-of-fit test. The null hypothesis (H_0) is a simpler model with fewer explanatory factors. In the context of this thesis, the null model is a three-way table with two independent variables (explanatory factors). The alternative hypothesis (H_1) expands the table to four dimensions by adding a third independent variable to the null model. The goodness-of-fit test assesses how much *information* about the response is gained by adding the third variable to the model. Is there sufficient evidence to confidently conclude the alternative model is a better explanation of the observed data than the null model?

A typical contingency table has a fixed number of levels for each dimension. Each dimension corresponds to a variable. Each cell represents a unique combination of discrete levels, one for each variable. The cell entries in the table are simply result of binning observations into the appropriate cells based on the observational level of each variable. For standard contingency tables with a large number of observations (events) and small dimensionality (few cells), the cell counts tend to be large (>5).

Hypothesis tests evaluate the abnormality of the observed test statistic (from the observed contingency table) based on where it falls on a distribution of possible test statistics assuming the null hypothesis was true. For standard contingency tables, the hypothesis test just described is easily performed since the distribution of test statistics can be assumed to asymptotically converge to a parametric (χ^2) distribution. There is no guarantee of the validity of the asymptotic assumption for *sparse* tables with low cell counts (<5), however. The χ^2 distribution assumption has been shown to be inaccurate or biased when expected cell counts are small (Haberman 1988). For these special contingency tables, the χ^2 asymptotic assumption for the distribution of test statistics cannot be relied upon to accurately estimate a p-value.

Although asymptotic inference may not be valid for *sparse* tables, an exact distribution of test statistics could still be obtained by enumerating every possible contingency table, calculating its test statistic, and assigning the appropriate probability of occurrence to that table. This is known as exact inference (for example, Fisher's (1922) exact test for 2×2 contingency tables). Unfortunately, it is computationally expensive (if not impossible) to enumerate all possible tables for large, multidimensional contingency tables that could have many possible permutations. Since we are unable to use either asymptotic or exact inference, we pursue approximate inference via sampling.

1.2 Motivation

The ease with which data are now collected, along with increases in computing power, has naturally led to the pervasiveness of contingency tables with more covariates and increased dimensionality. As observations are being spread across more and more dimensions (cells), the likelihood of contending with a *sparse*, multidimensional contingency table is amplified. These contingency tables are often too sparse for asymptotic approximation to be valid and contain too many cells to conduct exact inference through enumerating all possible tables in the conditional state space.

Sampling contingency tables provides an alternative way of generating a distribution for a test statistic without assuming a parametric distribution or enumerating every possible table. Conducting hypothesis tests in this manner is known as approximate inference. The accuracy of approximate inference depends on the degree to which the sampled distribution of test statistics matches the true distribution, which is usually unknown. Markov chain Monte Carlo (MCMC) sampling of contingency tables has been shown to be useful for obtaining hypothesis test p-values and estimating the number of contingency tables in the state space satisfying fixed marginal sums (Besag and Clifford 1989; Chen et al. 2005; Diaconis and Efron 1985; Guo and Thompson 1992).

Unfortunately, a naïve MCMC sampling approach may lead to biased results since it assumes every possible contingency table in the state space is connected via a single MCMC chain. For some models it is non-trivial to prove connectivity, even for two dimensional contingency tables. It can be extremely difficult to prove this condition is met for contingency tables larger than two dimensions. We require a modification of MCMC sampling that is suitable for conditional state spaces in which not all contingency tables are connected. By augmenting MCMC with sequential importance sampling (SIS), it becomes possible to sample contingency tables not connected via MCMC chain.

1.3 Research Objectives

In this thesis, we present an alternative method for hypothesis testing on *sparse*, multidimensional contingency tables, a problematic dataset for asymptotic hypothesis testing. We propose a combination of MCMC and SIS methods to sample the multidimensional contingency table state space in order to *approximate* a test statistic distribution under H_0. In our implementation of MCMC and SIS, our focus will be on an approach for testing the fit of multivariate logistic regression models.

In 2010, Hara et al. (2010) showed an explicit description of a *Markov basis*[1] for multiple Poisson regression models. In the case of bivariate logistic regression models, they also explicitly delineated a set of all moves connecting a Markov chain, provided the sum for each combination of levels of the covariates is positive. To generalize the sampler proposed by Hara et al. to multivariate logistic regression, we create a set of MCMC transitions for multivariate logistic regression models via an algorithm. Currently, no proof exists that

[1] A Markov basis is a finite set of moves on contingency tables such that all feasible contingency tables in the conditional state space are guaranteed to be connected via a Markov chain (Diaconis and Sturmfels 1998).

this set of transitions is able to connect every possible contingency table in the conditional state space through a Markov chain. Solely using MCMC for sampling would likely omit feasible regions of the conditional state space and bias the approximate distribution. In order to conduct the goodness-of-fit hypothesis tests desired without bias, we combine the set of transitions with a hybrid scheme of MCMC and SIS procedures proposed by Kahle et al. (2017b). More specifically, we use MCMC and SIS to sample contingency tables from the conditional state space defined by a fixed sufficient statistic.

After developing the hybrid sampler, we investigate the SIS algorithm closely. We show the SIS procedure might have high rejection rates in general for multivariate logistic regression models. We also observe some characteristics of the test statistic distribution resulting from the scheme and potential causes.

We end this paper with an application test of the MCMC/SIS hybrid scheme to a sparse, multidimensional contingency table dataset that measures the cognitive states of nuns with Alzheimer's disease over time (Mortimer 2012). We use our sampling scheme to assess which factors are associated with higher rates of transition to a state of dementia. We hope to use this research to validate our hybrid methodology as an alternative approach that analysts and researchers could apply to a variety of similar problems.

CHAPTER 2:
Definitions and Literature Review

This chapter defines basic terms for understanding MCMC and SIS and reviews some of the prior research regarding the application of these methods to contingency tables.

2.1 Definitions

The following definitions provide a foundation to understanding hypothesis testing for contingency tables, some of the original research presented in Section 2.2, and the sampling scheme proposed in Chapter 3.

2.1.1 Contingency Table

Contingency tables are used to study relationships between factors in a model (Yoshida 2010). Observations (events) are collected in which the response is recorded along with potential explanatory factors. The observed events are aggregated into a contingency table where each dimension of the table corresponds to one of the factors. The *cell counts* of the table represent the frequency of events in which the response and explanatory factors take on a unique combination of levels. An n-way contingency table is the n-dimensional table resulting from counting the number of events that occur at combinations of two or more discrete criteria (Drton et al. 2009).

2.1.2 Basic Notation

- *Levels* describe the finite number of categories contained within each dimension (factor) of the contingency table. A $2 \times 2 \times 3 \times 4$ table has 2 levels for the first factor, 2 levels for the second, 3 for the third, and 4 levels for the fourth factor.
- A *cell* is defined as a particular event within the contingency table that has a specified *level* for each factor of the table. A $2 \times 2 \times 3 \times 4$ table has 48 cells.
- *Cell counts* are the frequency of observations or events corresponding to a particular cell.

- $x_{\ell ijk}$ is the cell count for ℓ level of the first factor, i level of the second factor, j level of the third factor, and k level of the fourth factor.
- An alternative method for representing individual cells is to *flatten* the contingency table to a one-dimensional vector. A $2 \times 2 \times 3 \times 4$ contingency table $\mathbf{x}$ has 48 cells that could also be denoted as $\mathbf{x} = \{x_1, x_2, ..., x_{48}\}$.

2.1.3 Sparse

Contingency tables with small cell counts are referred to as *sparse* (Agresti 2002). A common rule of thumb is a contingency table is considered sparse if any expected cell count is less than five. As the dimensionality increases in multiway contingency tables, the expected cell counts necessarily decrease as observations are spread across more cells. A χ^2 test statistic distribution may not be valid for sparse contingency tables (Haberman 1988). In this analysis, we consider the contingency tables analyzed to be *sparse* and thus not valid for asymptotic inference using a χ^2 approximation of the null distribution of test statistics.

2.1.4 Log-linear model

Contingency tables are often modeled as log-linear models (Agresti 2002). We build our log-linear model by extending the work of Hara et al. (2010) from bivariate logistic regression to trivariate logistic regression.

Let $A \in \{1, ..., I\}$, $B \in \{1, ..., J\}$, and $C \in \{1, ..., K\}$ represent three covariates (A, B, C) with corresponding ordinal levels. For $i \in \{1, ..., I\}$, $j \in \{1, ..., J\}$, $k \in \{1, ..., K\}$, let random variables X_{1ijk} and X_{2ijk} represent the number of *successes* and *failures* (cell counts) for level (i, j, k). Under a log-linear model, the cell count X_{1ijk} is modeled as a Poisson random variable with parameter λ_{ijk} such that

$$X_{1ijk} \sim \text{Poisson}\left(\lambda_{ijk}\right).$$

Under the generic, saturated log-linear model with three covariates, the parameter λ_{ijk} is described in canonical form as a function of the covariates and interactions:

$$\log(\lambda_{ijk}) = \mu + \lambda_i^A + \lambda_j^B + \lambda_k^C + \lambda_{ij}^{AB} + \lambda_{ik}^{AC} + \lambda_{jk}^{BC} + \lambda_{ijk}^{ABC}. \tag{2.1}$$

An independence model is a special case of the general model in which we assume there are no interactions between the variables and so the values of the interaction term coefficients are all zero.

Our model further assumes the specified form: $\lambda_i^A = i \cdot \alpha$, $\lambda_j^B = j \cdot \beta$, and $\lambda_k^C = k \cdot \gamma$. Under these assumptions, the log-linear model reduces to:

$$\log(\lambda_{ijk}) = \mu + i \cdot \alpha + j \cdot \beta + k \cdot \gamma. \tag{2.2}$$

These assumptions are appropriate when we believe each observation is independent, there are no interactions between variables, and a linear relationship exists between the log odds of the response and the values of the explanatory variables. The log-linear model presented in Equation 2.2, along with the log-likelihood ratio, forms the basis of our test statistic calculation for a hypothesis test.

2.1.5 Log-Likelihood Ratio

Let Θ_1 be the parameter space for the alternative hypothesis and Θ_0 be the parameter space for the null hypothesis. Note that $\Theta_0 \subset \Theta_1$ because we specify that the null hypothesis is nested in the alternative hypothesis.

Also let $L_1(\theta|x)$ be the likelihood function for the alternative and $L_0(\theta|x)$ be the likelihood function for the null hypothesis, where θ is a parameter and x is an observation. Note that L_0 is a special case of L_1 where one of the parameters $\theta_i \in \Theta_1$ is fixed (often to 0). Since $\Theta_0 \subset \Theta_1$ and $L_1(\theta|x)$ is a more general form than $L_0(\theta|x)$, the alternative will in general be more likely than the null,

$$\max_{\theta \in \Theta_1} L_1(\theta|x) \geq \max_{\theta \in \Theta_0} L_0(\theta|x).$$

The likelihood ratio test statistic is

$$\Delta(x) = \frac{\max_{\theta \in \Theta_0} L_0(\theta|x)}{\max_{\theta \in \Theta_1} L_1(\theta|x)} < 1.$$

The log-likelihood ratio test statistic (G^2) is defined as

$$G^2 = -2 \cdot \log(\Delta) = -2 \cdot \left(\log \left(\max_{\theta \in \Theta_0} L_0(\theta|x) \right) - \log \left(\max_{\theta \in \Theta_1} L_1(\theta|x) \right) \right). \tag{2.3}$$

2.1.6 Test Statistic

A test statistic of a contingency table is a statistical measure of distance between the table and the maximum likelihood estimator (MLE) under the given model. Pearson's chi-squared and the log-likelihood ratio are examples of test statistics. For our comparison of two nested binomial models, we take advantage of the equivalency between the log-likelihood ratio and the difference in residual deviances between the null and alternative models.

2.1.7 Bivariate Logit versus Trivariate Logit Test Statistic

Let $\mathbf{T} = (T_{\ell ijk})$ be a contingency table. Using the log-linear model described in Section 2.1.4, the null hypothesis is a nested subset of the alternative hypothesis and fits the contingency table cell counts to a bivariate logistic regression model:

$$H_0 : \log(\lambda_{ijk}) = \mu + i \cdot \alpha + j \cdot \beta. \tag{2.4}$$

The alternative hypothesis fits the contingency table cell counts to a trivariate logistic regression model:

$$H_1 : \log(\lambda_{ijk}) = \mu + i \cdot \alpha + j \cdot \beta + k \cdot \gamma. \tag{2.5}$$

Let G^2 be the log-likelihood ratio test statistic (Section 2.1.5) between the null and alternative hypotheses. Here we set

$$G^2(\mathbf{T}) = 2 \cdot (\text{log max likelihood under } H_1 - \text{log max likelihood under } H_0). \tag{2.6}$$

The log max likelihood under the null in Equation (2.4) is

$$\begin{array}{rcl} \log\big(L_0(\mathbf{T})\big) & = & \sum_{i=1}^{I} \sum_{j=1}^{J} \big(T_{1ijk} \cdot \log(\frac{\exp(\mu^* + i \cdot \alpha^* + j \cdot \beta^*)}{1+\exp(\mu^* + i \cdot \alpha^* + j \cdot \beta^*)}) \\ & + & T_{2ijk} \cdot \log(\frac{1}{1+\exp(\mu^* + i \cdot \alpha^* + j \cdot \beta^*)})\big) \end{array}$$

where μ^*, α^*, β^* are the MLEs under the null hypothesis. The log max likelihood under the alternative in Equation (2.5) is

$$\begin{aligned}\log\big(L_1(\mathbf{T})\big) &= \textstyle\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\big(T_{1ijk}\cdot\log(\frac{\exp(\mu^{**}+i\cdot\alpha^{**}+j\cdot\beta^{**}+k\cdot\gamma^{**})}{1+\exp(\mu^{**}+j\cdot\alpha^{**}+k\cdot\beta^{**}+s\cdot\gamma^{**})})\\ &+ T_{2ijk}\cdot\log(\frac{1}{1+\exp(\mu^{**}+i\cdot\alpha^{**}+j\cdot\beta^{**}+k\cdot\gamma^{**})})\big)\end{aligned}$$

where μ^{**}, α^{**}, β^{**}, γ^{**} are the MLEs under the alternative hypothesis.

2.2 Literature Review

Given a contingency table, calculating the goodness-of-fit test statistic requires fitting two different models (H_0 and H_1) to the contingency table data and calculating the difference between the residual deviances (G^2). The *distribution* of test statistics under H_0 is necessary to conduct a hypothesis test, however. In order to calculate the distribution of test statistics, one must know the conditional distribution of contingency tables. Unfortunately, it is difficult to determine the exact, conditional distribution of multiway contingency tables because there are too many tables to enumerate. Markov chain Monte Carlo simulation provides a means of approximating the distribution by sampling many contingency tables from the conditional state space.

2.2.1 Markov Chain Monte Carlo (MCMC)

A *Markov chain* is a stochastic sequence of events or locations characterized by one-step transition probabilities of moving from one state to another (Dobrow 2016). An important property of Markov chains is the memoryless property that dictates the probability of transitioning to a future state depends *only* on the current state and not the sequence of arriving at that state. In the context of this paper, a Markov chain can be thought of as a random walk on a graph whose vertices are contingency tables satisfying the sufficient statistic (Dobrow 2016). The *sufficient statistic* (**b**) is a vector of parameters (commonly row sums and column sums) that are fixed under H_0 and specify the *conditional state space* ($S_{\mathbf{b}}$), the set of all contingency tables feasible under H_0.

As suggested by the name, MCMC algorithms combine Markov chains and Monte Carlo simulation. These methods provide a means for estimating complex and high-dimensional probability distributions via sampling (Dobrow 2016). MCMC approaches estimate the true distribution of tables by randomly traversing and sampling from the conditional state space ($S_{\mathbf{b}}$) rather than computing the distribution directly. Stepping from one contingency table to another on the random walk is called a *move* ($\mathbf{z}$). A valid move must preserve the sufficient statistic ($\mathbf{b}$). Since each state on the MCMC chain is dependent on the previous state, *burn-in* and *thinning* techniques are often applied to the MCMC chain to reduce dependencies among samples (Link and Eaton 2012).

Diaconis and Sturmfels (1998) described algebraic ways to construct MCMC samplers in discrete exponential families that can be used to conduct hypothesis testing where asymptotic inference or exact enumeration cannot be applied. Their baseline contingency table example was a two-way table characterized by a hypergeometric distribution as the conditional distribution given fixed row and column sums (sufficient statistic) under the independence model. The algorithm they devised proposes a *move* based on a $\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$ shift in the values of 4 cells located by the union of 2 random rows and 2 random columns. An accepted move must preserve the sufficient statistic as well as maintain every cell entry as non-negative. The result of random move after random move is a Markov chain of contingency tables on the conditional state space. Diaconis and Sturmfels proved that by the usual *Metropolis-Hastings* acceptance procedure (Hastings 1970), the algorithm they described gives a connected, aperiodic, reversible (*ergodic*) Markov chain with the stationary distribution equal to the hypergeometric distribution specified by the sufficient statistic (Diaconis and Sturmfels 1998).

In order for an MCMC algorithm to proportionally sample from all contingency tables (states) in the conditional state space, all states must be connected via a Markov chain. With a Markov basis already known, running an MCMC algorithm is efficient in computational time and not memory intensive. Hara et al. (2010) applied the methods of Diaconis and Sturmfels to show the necessary and sufficient conditions with a subset of a Markov basis to sample contingency tables without sampling bias via an ergodic Markov chain under the univariate and the bivariate Poisson regression model with the assumption that all sufficient statistics are strictly positive. In this project, we extend their logic to a trivariate regression model.

MCMC methods are not without drawbacks. The contingency tables resulting from MCMC are not independent. Each subsequent table is dependent on the previous table. In our analysis, burn-in and thinning will be used to reduce dependency. The initial computation of a Markov basis is also problematic. For 3-way contingency tables with fixed 2-margins, De Loera and Onn (2005) showed there is no upper bound on the number of elements in a Markov basis.

In general, with only a subset of a Markov basis (a set of moves), there is no guarantee of the connectivity of all feasible states. To attempt to sample from all areas of the conditional state space, we utilize SIS in addition to MCMC. SIS allows for the independent sampling of the conditional state space. Unlike MCMC, tables sampled need not be connected via a Markov chain. Each independent SIS table will have its own Markov chain in an attempt to sample from all the Markov chains existing in the conditional state space.

2.2.2 Sequential Importance Sampling (SIS)

SIS was first applied to sampling two-way contingency tables under the independence model in Chen et al. (2005). SIS randomly samples a contingency table from the conditional state space ($S_{\mathbf{b}}$) by populating the cell counts of the table one-by-one. The count for each cell is a random variable, therefore the resulting contingency table is also a random variable. In our algorithms for SIS, we use the same procedure described in Chen et al. (2006). The multidimensional contingency table **X** is flattened into vector form for convenience in applying linear algebra techniques. The minimum and maximum feasible values for the first cell count are calculated using integer programming (IP). The cell count for the first cell, denoted as X_1 (random variable), is randomly sampled from the uniform distribution bounded by the IP minimum and maximum. After fixing X_1, the second cell X_2 is sampled in the same manner but conditional on X_1. The entire sampled contingency table is the result of sequentially fixing all the cells in the table. Clearing all the cell counts and sequentially sampling all cells again generates another contingency table. A desirable outcome of SIS is that the second table is generated independently of the first.

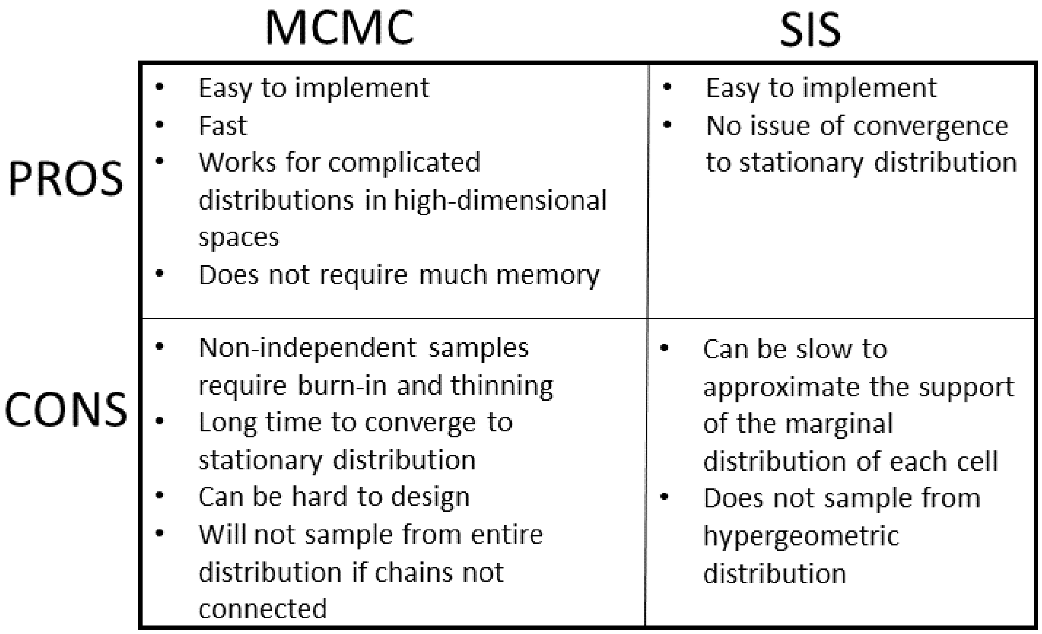
Unlike MCMC, SIS does not require the computation of a Markov basis, which is often difficult to compute (Xi et al. 2013). The SIS procedure independently samples from the conditional distribution, while a Markov chain may require many iterations in order to be independent of the current state. In these two regards SIS overcomes some disadvantages of MCMC, but SIS also presents a new set of problems.

The SIS procedure samples contingency tables independently and close to uniformly. Unfortunately, the appropriate sampling distribution for a contingency table with a fixed sufficient statistic is a hypergeometric distribution, not uniform (Agresti 2002). Another difficulty of SIS is computing the *marginal distribution* (feasible integer values) of each cell count. We typically approximate the marginal distribution with a discrete uniform distribution bounded by the minimum and maximum count values a cell can assume. These integer bounds require non-trivial time to solve via IP. After taking the time to solve for the bounds, there could still be an integer value within the interval that makes the problem infeasible. If this value is randomly sampled, the sufficient statistic will not be satisfied and the SIS algorithm must restart from the beginning. Such rejections are known as *holes*.[2] One of the major disadvantages of SIS is that rejections lead to increased computational time due to resampling.

[2] A *hole* of a *semigroup* is a right hand side (RHS) $\mathbf{b} \in \mathbb{R}^{d_1}$ with $A \in \mathbb{Z}^{d_1 \times d_2}$ such that $P_b \neq \emptyset$ and $P_b \cap \mathbb{Z}^{d_2} = \emptyset$ where the polytope $P_b = \{\mathbf{x} \in \mathbb{R}^{d_2} \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq 0\}$. For the application in this paper, the matrix A is non-negative and therefore the solution $\mathbf{x}$ for the equation $A \cdot \mathbf{x} = \mathbf{b}$ is always bounded (Schrijver 1986). A *hole* indicates sequentially sampling $x_i \in \{x_1, ..., x_{d_2}\}$ such that further random sampling to determine $\mathbf{x}$ is integer infeasible. There may exist a *real* solution, but no integer solution exists given the particular x_i that have already been fixed. Section 4.2 presents an example hole.

2.3 Summary

The advantages and disadvantages of MCMC and SIS previously mentioned are summarized in Figure 2.1. MCMC and SIS methods developed and advanced by previous researchers are the building blocks on which we construct a hybrid sampling scheme in the next chapter for sampling sparse, multidimensional contingency tables. Appropriately sampling the conditional state space allows the distribution of test statistics to be approximated and ultimately permits hypothesis testing for sparse, multidimensional tables where asymptotic inference may not be valid.



	MCMC	SIS
PROS	• Easy to implement • Fast • Works for complicated distributions in high-dimensional spaces • Does not require much memory	• Easy to implement • No issue of convergence to stationary distribution
CONS	• Non-independent samples require burn-in and thinning • Long time to converge to stationary distribution • Can be hard to design • Will not sample from entire distribution if chains not connected	• Can be slow to approximate the support of the marginal distribution of each cell • Does not sample from hypergeometric distribution

Figure 2.1. MCMC and SIS Comparison.

This figure describes the advantages and disadvantages of MCMC and SIS sampling methods. A hybrid approach using both methods can help to overcome the deficiencies in each. Sources: Chen et al. (2006); Steorts (2016).

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

Chapter 3 presents the logic of combining MCMC and SIS into a hybrid algorithm for sampling the conditional state space of sparse, multidimensional contingency tables. An example hypothesis test is used extensively throughout the chapter to aid in understanding the application of the mathematical formulae. The R code supporting the sampler implementation is provided in Appendix C.

3.1 Nun Cognitive Observational Dataset

The nun cognitive observation dataset originally comes from Mortimer (2012), who conducted a study on 672 participants from 1031 Catholic sisters born before 1917 from the School Sisters of Notre Dame religious order. The nuns were asked to voluntarily participate in the study from 1991 to 1993 and were all age 75 years and older at the time of the study. Each nun had cognitive ability recorded for up to 10 unevenly spaced examinations, and the time between examinations ranged from 0.421 to 3.911 years (Wei and Kryscio 2016). After removal of data with missing values, the final dataset had 461 participants with 2480 total observations (Wei and Kryscio 2016). Since its compilation, this study has been used numerous times in academic papers for application of sampling techniques and insights into factors affecting Alzheimer's disease.

The levels of five factors were recorded for each observation: prior cognitive status, current cognitive status, presence of APOE-4 allele[3], highest education level achieved, and age.

- Prior/current cognitive status has five levels: intact cognition (1), mild cognitive impairment (MCI) (2), global impairment (GI) (3), dementia (4), and death (5).
- Presence of APOE-4 allele (APOE4) has two levels: not present (1) and present (2).
- Education has three levels: no college (1), college degree (2), and post graduate degree (3).
- In the dataset, age is a continuous variable. To construct a contingency table, we

[3] APOE-4 allele is a gene present in 10-15% of the population that is associated with increased risk for Alzheimer's and earlier onset (National Institute on Aging 2015).

categorize observations into age quartiles. Level 1 corresponds to age 77-83.6, level 2 to 83.6-87.1, level 3 to 87.1-90.5, and level 4 to 90.5-104.3.

Markov models are often used to model transitions from one cognitive state to another. Tyas et al. (2007) applied a Markov chain model and Wei and Kryscio (2016) applied a Semi-Markov model to the nun dataset. In both investigations, the researchers modeled the transitions between cognitive states as a discrete time Markov chain (DTMC) with a finite state space of five states $(1, 2, 3, 4, 5)$ and a transition probability matrix:

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

Dementia (4) and death (5) are treated as absorbing states. Figure 3.1 shows the possible state transitions. The numbers on each arc correspond to the observed number of transitions from prior state (arc tail) to current state (arc head). We are interested in discerning which factors (APOE4, education, age) contribute to increased transition probabilities to dementia. The complete data set used in our analysis is reported in Table A.1 of Appendix A.

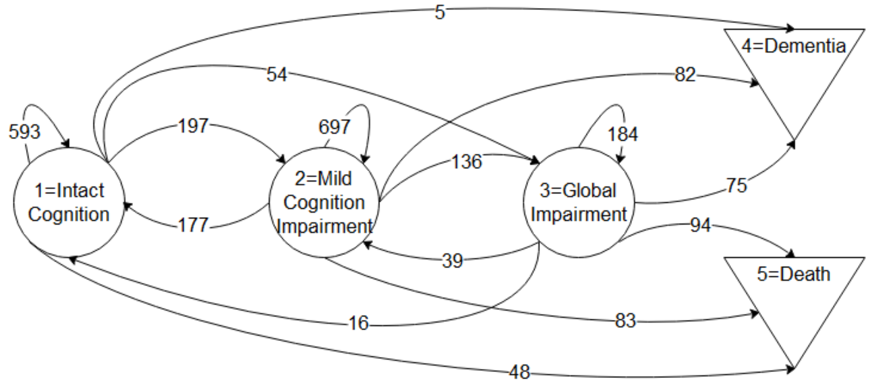


Figure 3.1. Discrete time Markov Chain for Nun Cognitive Dataset.

This figure shows the five states that comprise the Markov state space. Numbers displayed on arcs represent the number of transitions from state i to state j observed in the dataset. Transitions to the same state are permissible. Source: Wei and Kryscio (2016).

3.2 Multivariate Logistic Regression Model

In this thesis, we make use of a Poisson regression model, a log-linear model form of regression analysis, which can be used to model contingency table cell counts (see Section 2.1.4). Under this model, the response variable has a Poisson distribution and the logarithm of its expectation parameter is a linear form of unknown parameters (Equations 2.4, 2.5). A logistic regression is a special case of the Poisson regression model in which the response variable has only two levels. The applications of Poisson regression include predicting hospital admissions (White 2009), estimating the depth of the recruiting market (Monaghan 2016), and our application of investigating the cognitive state transitions of a patient with Alzheimer's disease (Xie 2016).

With the sparse, multidimensional contingency table dataset, we seek to compare the relative fit of the observed data to two nested log-linear models (H_0 and H_1) using a goodness-of-fit hypothesis test. More specifically, we are interested in the associations between transition rates among cognitive states and three factors of interest: the presence of APOE-4 allele, education, and age. Following the basic modeling of Salazar et al. (2007), Wei and Kryscio (2016), and Xie (2016), we model the relationships between variables as a multivariate logistic regression model such that:

$$\log\left(\frac{P_{s4}}{P_{sv}}\right) = \alpha_{sv} + \beta_{1sv}Y_1 + \beta_{2sv}Y_2 + \beta_{3sv}Y_3, \tag{3.1}$$

for $s \in \{1,2,3\}$ and $v \in \{1,2,3\}$. Here $\left(\frac{P_{s4}}{P_{sv}}\right)$ mimics the odds ratio. The numerator is the number of transitions from state s to *dementia* (state 4). The denominator is the number of transitions from state s to state v. Our analysis only examines the relative odds of transitioning to dementia, but the methodology could be applied to any state transitions of interest. The independent variables are denoted as Y_i. $Y_1 \in \{1,2\}$ is the presence of APOE-4 allele, $Y_2 \in \{1,2,3\}$ is education level, and $Y_3 \in \{1,2,3,4\}$ is age quantile. For notation purposes, we designate APOE-4 allele as having I levels, education as having J levels, and age as having K levels. α_{sv} and β_{isv} are the MLE parameters for the model given the observed data.

For each $s \in \{1, 2, 3\}$ and $v \in \{1, 2, 3\}$, we establish the following hypotheses:

$$\begin{aligned} H_0^{isv} : \ \beta_{isv} &= 0, \\ H_1^{isv} : \ \beta_{isv} &\neq 0, \end{aligned}$$

for variable $i \in \{1, 2, 3\}$. Given a null model of the contingency table data with two independent variables, we are determining whether adding the third independent variable improves the model as measured by a statistical threshold.

3.3 Example Setup

The following subsections establish a concrete example to illustrate sequentially proceeding through the necessary steps for a goodness-of-fit hypothesis test. This example is used throughout the chapter to demonstrate the algorithms discussed. Before proceeding to SIS and MCMC, the initial steps are to establish the competing hypotheses and extract the relevant data from the entire dataset.

3.3.1 Establish Hypotheses

Suppose we are interested in testing the following hypotheses based on the model defined by Equation 3.1 for $s = 2$, $v = 2$, and $i = 3$. The null and alternative hypotheses are:

$$H_0^{322} : \log\left(\frac{P_{24}}{P_{22}}\right) = \alpha_{22} + \beta_{122}Y_1 + \beta_{222}Y_2, \tag{3.2}$$

$$H_1^{322} : \log\left(\frac{P_{24}}{P_{22}}\right) = \alpha_{22} + \beta_{122}Y_1 + \beta_{222}Y_2 + \beta_{322}Y_3. \tag{3.3}$$

These hypotheses are testing whether adding the Age factor (Y_3) to a model of the transitions from MCI ($s = 2$) to dementia relative to the transitions from MCI ($s = 2$) to MCI ($v = 2$) significantly improves the model.

3.3.2 Extract Relevant Data to Observed Table ($\mathbf{x_0}$)

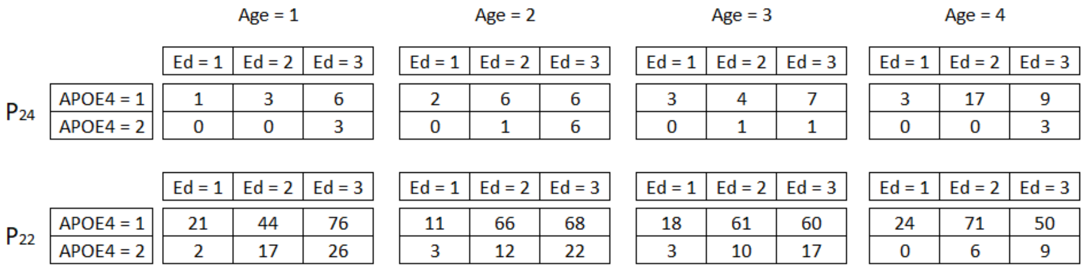
Testing the above hypotheses requires generating a test statistic from a 4-dimensional contingency table for $s = 2$, $v = 2$, and $i = 3$. To conduct the hypothesis test, the test statistic will later be compared to an approximate distribution of test statistics.

First, we construct an $L \times I \times J \times K$ contingency table and assign two *levels* to the first dimension (ℓ) of our contingency table. For the example hypothesis test, $\ell = 1$ corresponds to state transitions where prior state = 2 and current state = 4 (P_{24}). $\ell = 0$ corresponds to state transitions where prior state = 2 and current state = 2 (P_{22}). For the nun dataset, the resulting 4-way contingency table is 2 (transition levels) $\times$ 2 (APOE4) $\times$ 3 (education) $\times$ 4 (age). The are 48 *cells* in this contingency table.

Equivalently, $\ell = 1$ signifies transitioning from MCI to dementia (bad outcome). $\ell = 0$ signifies maintaining the cognitive state of MCI between two contiguous observations (good outcome). This hypothesis tests whether the association between the relative transition probabilities $\left(\frac{P_{24}}{P_{22}}\right)$ and age (Y_3) is significant to the model in the presence of APOE-4 allele (Y_1) and education (Y_2).

To construct the observed contingency table ($\mathbf{x_0}$), the 82 observations that transition from state 2 to state 4 (P_{24}) are collected along with the 697 observations for P_{22}. We use d to denote the dimension of $\mathbf{x_0}$. In this case, $d = \dim(\mathbf{x_0}) = 2 \times 2 \times 3 \times 4$. Table 3.1 shows a 2-dimensional hierarchical representation of the observed contingency table, which is in reality a 4-dimensional contingency table.

Table 3.1. Example Observational Data ($\mathbf{x_0}$).

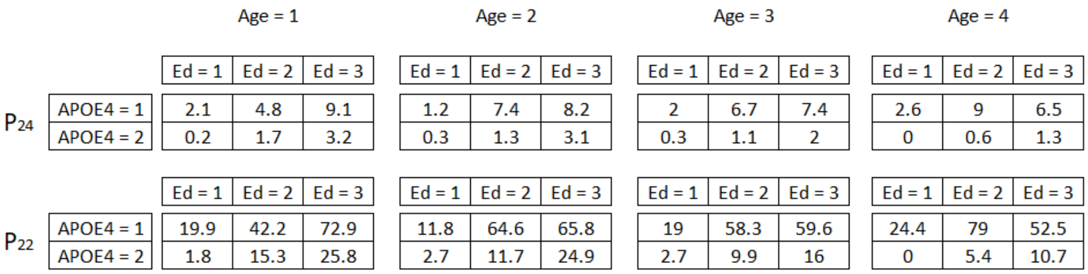


		Age = 1			Age = 2			Age = 3			Age = 4		
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P24	APOE4 = 1	1	3	6	2	6	6	3	4	7	3	17	9
	APOE4 = 2	0	0	3	0	1	6	0	1	1	0	0	3
P22	APOE4 = 1	21	44	76	11	66	68	18	61	60	24	71	50
	APOE4 = 2	2	17	26	3	12	22	3	10	17	0	6	9

This contingency table gives observational counts for P_{24} and P_{22}. The top half shows the 82 observations for P_{24} scattered across 24 cells. Examining a cell at random, we see that for APOE4=1, Ed=1, and Age=1, there was 1 transition from state 2 to state 4. There were 21 transitions from state 2 to state 2 given the same levels for the three factors. The small cell counts, particularly for some of the cells where APOE4 = 2 (present) and Ed = 1 (no college), suggest this is probably a *sparse* contingency table.

In this particular example, $\mathbf{x_0}$ contains 779 observations (total cell counts). Each observation represents a nun transitioning between the corresponding prior cognitive state and current cognitive state. By observation, many of the cells in this four dimensional contingency table contain small numbers (<5). Therefore, asymptotic inference utilizing a χ^2 distribution of test statistics is not assured to be valid. As a check of sparsity, the *expected* cell counts are calculated from the MLE of the null model and displayed in Table 3.2. As can be seen visually, this table meets the *sparse* table criteria discussed in Section 2.1.3.

Table 3.2. Expected Cell Counts under H_0^{322} for Observational Data ($E\left[\mathbf{x_0}\right]$).



		Age = 1			Age = 2			Age = 3			Age = 4		
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P24	APOE4 = 1	2.1	4.8	9.1	1.2	7.4	8.2	2	6.7	7.4	2.6	9	6.5
	APOE4 = 2	0.2	1.7	3.2	0.3	1.3	3.1	0.3	1.1	2	0	0.6	1.3
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P22	APOE4 = 1	19.9	42.2	72.9	11.8	64.6	65.8	19	58.3	59.6	24.4	79	52.5
	APOE4 = 2	1.8	15.3	25.8	2.7	11.7	24.9	2.7	9.9	16	0	5.4	10.7

This table shows the *expected* cell counts (P_{24} and P_{22}) using the MLE of the null model: $H_0^{322} : \log\left(\frac{P_{24}}{P_{22}}\right) = \alpha_{22} + \beta_{122}Y_1 + \beta_{222}Y_2$. The low expected cell counts confirm this is a sparse contingency table.

3.4 Test Statistic Calculation

To generate the test statistic, we first tabulate the data from Table 3.1 into a 779×4 matrix. Each row of the data matrix represents an observation. For an observation, the value in each column represents the observed level for the corresponding factor ($L, Y_1, Y_2,$ and Y_3). For example, there is one observation of (1,1,1,1). This corresponds to a nun that transitioned from state 2 to state 4 (L = 1) with APOE-4 allele not present (Y_1 = 1), no college (Y_2 = 1), and age = 77 - 83.6 (Y_3 = 1). There are 21 identical observations (rows) of (0,1,1,1). This corresponds to a nun that transitioned from state 2 to state 2 (L = 0) with APOE-4 allele not present (Y_1 = 1), no college degree (Y_2 = 1), and age = 77 - 83.6 (Y_3 = 1).

Using Equations 3.2 and 3.3 and the tabulated data, we fit two different generalized linear models, both with binomial distributions and a logit link function, to the observed data. The binomial distribution follows from the assumption that each observation is independent and can take on value $L = 1$ (transition from state 2 to 4) or $L = 0$ (transition from state 2 to 2). The larger model (H_1) contains all three covariates. The null model is a smaller model and contains only 2 covariates. In the example, the null model does not contain Y_3 (age covariate). The following R code computes the MLE for the two binomial logistic models defined in Equations 3.2 and 3.3.

```
res0 <- glm(L ~ Y1+Y2, family = binomial(link="logit"), data=tab)
res1 <- glm(L ~ Y1+Y2+Y3, family = binomial(link="logit"), data=tab)
```

Table 3.3 summarizes the MLE for these two logistic regressions.

Table 3.3. Null and Alternative Model Summary.

Variable	H_0	H_1
β_1 (APOE-4)	-0.01508	0.1091
	(0.30)	(0.31)
β_2 (Ed)	0.08281	0.1404
	(0.18)	(0.18)
β_3 (Age)		0.3270***
		(0.11)
Residual Deviance	524.04	514.81

$^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$

This table reports maximum likelihood parameter estimates under H_0 and H_1 given $\mathbf{x_0}$. Standard errors for each estimate are shown in parentheses. The residual deviance equals $-2\times$ the maximum log-likelihood. The difference in residual deviances is the observed test statistic (G_0^2). Coefficients and residual deviance are computed from the **glm** function in baseline **R**.

The log-likelihood ratio (G^2) test statistic is taken to be the difference in the residual deviances (refer to Section 2.1.5) between the two models. As expected, the larger (more parameters) alternative model has a smaller residual deviance than the null model. $G_0^2 = 524.04 - 514.81 = 9.23$.[4] In order to conduct a goodness-of-fit hypothesis test, we need to

[4] In the algorithms to follow, we denote the observed test statistic as $T(\mathbf{x_0})$ and the distribution of test statistics as $\mathbf{T}(\mathbf{X})$.

assess whether this test statistic is abnormally large. If so, this evidence supports H_1 and indicates Y_3 is a significant factor ($\beta_{322} \neq 0$) and should be added to the model.

The extremeness of the observed test statistic can be easily checked if the distribution of test statistics is known. The distribution of test statistics for standard contingency tables (all expected cell counts > 5) asymptotically approaches a χ^2 distribution with appropriate degrees of freedom. However, the asymptotic assumption is not guaranteed to be valid for *sparse* contingency tables (Haberman 1988). The next alternative is an exact test in which we enumerate all possible contingency tables that satisfy the fixed sufficient statistic (also known as fixed marginals).

3.5 Sufficient Statistic (b)

The sufficient statistic (**b**) is a fixed vector under the null model (H_0). The following elements (SS_n) comprise the sufficient statistic for the observed contingency table ($\mathbf{x_0}$).

- The number of transitions from a state s to state 4 (the sum of all cell counts where $\ell = 1$) is fixed:

$$\mathrm{SS}_1 = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} X_{1ijk}, \text{ where } X_{1ijk} \text{ is } \textit{cell count}. \tag{3.4}$$

- Weighted row sum is fixed:

$$\mathrm{SS}_2 = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} i \cdot X_{1ijk}. \tag{3.5}$$

- Weighted column sum is fixed:

$$\mathrm{SS}_3 = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} j \cdot X_{1ijk}. \tag{3.6}$$

The factor, for example Y_3, that is omitted from the null hypothesis is also omitted from the sufficient statistic. In this particular example, k does not have its *own* sufficient statistic element like i (Y_1) and j (Y_2) do. All sampled contingency tables are required to satisfy these three elements of the sufficient statistic.

- Additionally, for a unique (i, j, k) combination, the sum across both cells $(P_{s4} + P_{sv})$ is fixed to the sum of corresponding observed events.

$$\text{SS}_n = \sum_{\ell=0}^{\ell=1} X_{\ell ijk} = P_{s4}^{ijk} + P_{sv}^{ijk} \quad \forall i, j, k. \tag{3.7}$$

For example, $X_{1111}+X_{0111} = 1 + 21 = 22$. This value is fixed when sampling new contingency tables. Because it is part of the sufficient statistic, $X_{1111} + X_{0111}$ will always equal 22. There is an element in the sufficient statistic for each (i, j, k) combination (24 for the nun dataset). Hara et al. (2010) showed the elements presented in Equations 3.4 - 3.7 constitute the *minimal* sufficient statistic for the bivariate logistic regression model (H_0).

To calculate the sufficient statistic from the observed table, $\mathbf{x_0}$ is flattened from a $2\times2\times3\times4$ array to a 48×1 vector. This simple transformation preserves the count data exactly and allows for easier conceptualization of the sufficient statistic calculations via common 2-dimensional linear algebra.

For a factor with Z levels, a *configuration matrix* is generated as

$$\text{B} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 2 & \dots & Z \end{bmatrix}. \tag{3.8}$$

Following Hara et al. (2010), we consider two configurations $B = (\mathbf{b}_1, \cdots, \mathbf{b}_I)$ and $C = (\mathbf{c}_1, \cdots, \mathbf{c}_J)$, where $\mathbf{b}_i$ and $\mathbf{c}_j$ are column vectors. The Segre product of B and C is defined as

$$B \otimes C = \mathbf{b}_i \oplus \mathbf{c}_j \quad \forall\, i \in \{1, \cdots, I\},\ j \in \{1, \cdots, J\}, \quad \text{where}\, \mathbf{b}_i \oplus \mathbf{c}_j = \begin{pmatrix} \mathbf{b}_i \\ \mathbf{c}_j \end{pmatrix}. \tag{3.9}$$

Suppose we have three factors with levels = 2, 3, and 4, respectively. Using Equations 3.8 and 3.9, the Segre product of B, C, and D is:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 & 1 & 2 & 3 & 4 \end{bmatrix}.$$

The modified Segre product (Z) is obtained by removing the redundant rows (rows 3 and 5) and eliminating the row corresponding to the factor (in this case Y_3) not included in H_0 (row 6). The modified Segre product for our example is:

$$Z^{3x24} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 3 & 3 & 3 & 3 \end{bmatrix}.$$

The row corresponding to Y_3, which has levels = $\{1, 2, 3, 4\}$, has been removed.

A Lawrence lifting of the modified Segre product (Z) yields the appropriate matrix for calculating the sufficient statistic ($\mathbf{b}$). From Xie (2016), the Lawrence lifting matrix (A) of the modified Segre product (Z) with C columns[5] is defined as:

$$A = \Lambda(Z) = \begin{bmatrix} Z & 0 \\ I_C & I_C \end{bmatrix} \tag{3.10}$$

where I_C is the identity matrix of dimension C.

For our example, applying the Lawrence lifting matrix (Equation 3.10) to the observed data ($\mathbf{x_0}$) facilitates the calculation of all 27 elements of the sufficient statistic ($\mathbf{b}$). For the specified H_0 and $\mathbf{x_0}$, the equation $A \cdot \mathbf{x_0} = \mathbf{b}$ yields the following sufficient statistic vector:

$$\mathbf{b} = \begin{bmatrix} 82\ 97\ 196\ 22\ 2\ 47\ 17\ 82\ 29\ 13\ 3\ 72\ 13\ 74\ 28\ 21\ 3\ 65\ 11\ 67\ 18\ 27\ 0\ 88\ 6\ 59\ 12 \end{bmatrix}.$$

[5] C is the number of (i, j, k) combinations. In this example, $C = 24$.

The first three elements in the sufficient statistic correspond to Equations 3.4, 3.5, and 3.6 respectively. Elements 4-27 in the sufficient statistic represent Equation 3.7 applied to each of the 24 (i, j, k) combinations. Element 4 corresponds to fixing the sum of $X_{1111} + X_{0111}$ to 22.

The conditional state space ($S_{\mathbf{b}}$) is the set of all contingency tables satisfying the sufficient statistic ($\mathbf{b}$). If we could enumerate all contingency tables $\mathbf{x}$ satisfying the equation $A \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ and their associated probabilities, we could use exact inference to estimate the hypothesis test p-value. The numerous integer solutions satisfying $A \cdot \mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}$ preclude us from enumerating all possible contingency tables, however. As a result of the impediments to complete enumeration, we resort to randomly sampling contingency tables from the conditional state space. In order to approximate the true conditional distribution of contingency tables, we employ SIS and MCMC.

3.6 Overall Approach for Hypothesis Test

The following algorithms use the notation introduced in Section 2.1.2. A $2 \times 2 \times 3 \times 4$ contingency table $\mathbf{x}$ is comprised of 48 cells denoted as $\{x_1, x_2, ..., x_{48}\}$. Observed data are presented as lower case; random variables are upper case. Arrays are shown in bold. x_1 is an observed scalar corresponding to the first cell in the table; $\mathbf{x_1}$ is the first sampled contingency table (array). X_1 is a random scalar variable corresponding to the first cell in $\mathbf{X}$, which is itself a random variable (array) composed of $\{X_1, X_2, ..., X_{48}\}$.

Since aymptotic inference and exact enumeration may not be valid for sparse, multidimensional contingency tables, Algorithm 3.6.1 describes an overall sampling approach to conduct an exact conditional hypothesis test. The essential element of the algorithm is Step 2, a method for appropriately sampling the entire conditional state space ($S_{\mathbf{b}}$).

Algorithm 3.6.1 *Overall Sampling Approach for a Goodness-of-Fit Hypothesis Test.*

Input*: H_0 and H_1 as bivariate and trivariate log-linear models, respectively. The observed contingency table $\mathbf{x_0}$. Sample size N and observed test statistic $T(\mathbf{x_0})$.*

Output*: Estimated p-value for the hypothesis test from an approximate distribution of test statistics.*

Algorithm*:*

1. *From the observed contingency table* ($\mathbf{x_0}$) *and the Lawrence lifting matrix A, compute the sufficient statistic* $\mathbf{b}$ *(detailed in Section 3.5).* $S_\mathbf{b}$ *is the conditional state space specified by* $\mathbf{b}$*. That is, a set of all possible contingency tables* ($\mathbf{X_i}$) *satisfying the equation* $A \cdot \mathbf{X_i} = \mathbf{b}, \quad \mathbf{X_i} \geq \mathbf{0}$.
2. *Sample tables* $\mathbf{X_1}, \ldots, \mathbf{X_N}$ *from* $S_\mathbf{b}$ *according to the hypergeometric distribution.*
3. *Compute the test statistic for each table* $T(\mathbf{X_1}), \ldots, T(\mathbf{X_N})$.
4. *Estimate the right-tailed p-value by counting the frequency* $\frac{\mathbb{I}_{T(\mathbf{x_0}) \leq T(\mathbf{x_i})}}{N}$ *for* $i = 1, \ldots N$*, where* $\mathbb{I}$ *is the indicator function.*

After completing step 1 of Algorithm 3.6.1, we seek to sample tables from the conditional state space according to a hypergeometric distribution. MCMC using the Metropolis-Hastings (MH) acceptance ratio samples tables from a hypergeometric distribution, but MCMC alone is not sufficient to approximate the distribution of contingency tables (and test statistics). Without a Markov basis for the conditional state space ($S_\mathbf{b}$), there is no guarantee all states (contingency tables) are connected via by a single MCMC chain. To ensure adequate sampling throughout the conditional state space, we rely on SIS.

3.7 Sequential Importance Sampling (SIS)

Unlike MCMC, SIS provides a means of independently generating contingency tables ($\mathbf{X_i}$) from the conditional state space. Unfortunately, SIS does not sample tables from a hypergeometric distribution, a drawback later rectified with MCMC.

Algorithms 3.7.1, 3.7.2, 3.7.3, and 3.7.4 delineate the steps necessary for our implementation of SIS. As before, let $A \in \mathbb{Z}^{d_1 \times d_2}$ be the Lawrence lifting matrix and $\mathbf{b} \in \mathbb{Z}^{d_1}$ be the sufficient statistic, where d_1, $d_2 \in \mathbb{N}$ and $\mathbb{N}$ is the set of natural numbers ($\mathbb{N} = \{1, 2, \ldots\}$). $\mathbf{X}$ is a random variable (contingency table) composed of individual cell counts $\{X_1, X_2, \ldots X_{d_2}\}$.

Algorithm 3.7.1 *Compute the Lower Bound for X_1 using IP.*

Input*: The matrix $A \in \mathbb{Z}^{d_1 \times d_2}$ and a vector* **b** $\in \mathbb{Z}^{d_1}$ *for the system*

$$\begin{aligned} A \cdot \mathbf{x} =& \ \ \mathbf{b} \\ \mathbf{x} \ \ \geq& \ \ \mathbf{0} \\ \mathbf{x} \ \ \in& \ \mathbb{Z}^{d_2}. \end{aligned}$$

Output*: A lower bound (LB_1) on the individual cell count X_1.*

Algorithm*:*

1. *Run* `Rcplex` *to solve for* **x** *given constraints defined by A* and **b** *and the IP variable constraints on* **x** *with an objective function of*

$$\min x_1.$$

2. *Return LB_1 = the optimal objective value from* `Rcplex`.

Continuing the example begun in Section 3.3, we run Algorithm 3.7.1 once for x_1. Since none of the other random variables $\{X_2, X_3..., X_{48}\}$ of **x** are fixed, X_1 has a large range of possible values. Not surprisingly, LB_1 equals 0. Once cell counts in the contingency table become sequentially fixed, the ranges for each subsequent X_i tend to shrink. Using a similar method as Algorithm 3.7.1, we calculate the upper bound on the first cell in contingency table.

Algorithm 3.7.2 *Compute the Upper Bound for x_1 with IP.*

Input*: The matrix $A \in \mathbb{Z}^{d_1 \times d_2}$ and a vector* **b** $\in \mathbb{Z}^{d_1}$ *for the system*

$$\begin{aligned} A\mathbf{x} =& \ \ \mathbf{b} \\ \mathbf{x} \ \geq& \ \ \mathbf{0} \\ \mathbf{x} \ \in& \ \mathbb{Z}^{d_2}. \end{aligned}$$

Output*: An upper bound (UB_1) on X_1.*

Algorithm*:*

1. *Run* `Rcplex` *with A, $\mathbf{b}$ and the input IP constraints with an objective function of*

$$\max x_1.$$

2. *Return UB_1 = the optimal objective value from* `Rcplex`.

Also as expected, $UB_1 = 22$, the highest possible cell count due to $X_{1111} + X_{0111}$ being fixed at 22 by Equation 3.7 (from observed Table 3.1). A random integer, designated x_1^*, is drawn from a discrete uniform distribution bounded by $[LB_1, UB_1]$. In this scenario, our random draw is $x_1^* = 5$. After fixing cell $X_1 = x_1^*$, Algorithm 3.7.3 updates the system of linear equations ($A \cdot \mathbf{x} = \mathbf{b}$).

Algorithm 3.7.3 *Update Constraints by Reducing A and* $\mathbf{b}$.

Input*: The matrix $A \in \mathbb{Z}^{d_1 \times d_2}$ and a vector $\mathbf{b} \in \mathbb{Z}^{d_1}$ for the system*

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \\ \mathbf{x} &\in \mathbb{Z}^{d_2} \end{aligned}$$

and an additional constraint from random sampling $X_1 = x_1^*$.

Output*: A new system of constraints specified by A' and* $\mathbf{b}'$.

Algorithm*:*

1. *Let A_1 be the first column of A and let A' be the remaining columns of A. $A' \in \mathbb{Z}^{d_1 \times (d_2 - 1)}$. $A' = A - A_1$.*
2. *Set $\mathbf{b}' = \mathbf{b} - A_1 \cdot x_1^*$.*[6]
3. *Return A' and* $\mathbf{b}'$.
4. *Update the constraints to $A' \cdot \mathbf{x}' = \mathbf{b}'$.*

[6] Note that $A \cdot \mathbf{x} = A' \cdot \mathbf{x}'_{\{2 \dots 48\}} + A_1 \cdot x_1^* = \mathbf{b} = \mathbf{b}' + A_1 \cdot x_1^*$ where $\mathbf{x}'_{\{2 \dots 48\}}$ are the cell counts $\{x_2, x_3, \cdots, x_{48}\}$.

Algorithms 3.7.1, 3.7.2, and 3.7.3 provide a method for sequentially sampling x_1 to x_{48} while preserving sufficient statistic $\mathbf{b}$. We iterate through this process until every cell x_i is fixed or the problem becomes infeasible due to finding a *hole*. Algorithm 3.7.4 combines the three algorithms into an SIS procedure for sampling a random contingency table $\mathbf{X_i}$ from the conditional state space ($S_{\mathbf{b}}$). In an attempt to maximize the variety of contingency tables sampled, we modify Algorithm 3.7.3 to allow the cell being updated to be chosen randomly rather than updating cells in order from 1 to 48 .

Algorithm 3.7.4 *Sequential Importance Sampling of the Conditional State Space.*

Input*: The matrix* $A \in \mathbb{Z}^{d_1 \times d_2}$ *and a vector* $\mathbf{b} \in \mathbb{Z}^{d_1}$ *for the system*

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \\ \mathbf{x} &\in \mathbb{Z}^{d_2}. \end{aligned}$$

Output*: A contingency table* $\mathbf{X}$ *sampled via SIS that satisfies the sufficient statistic* $\mathbf{b}$.

Algorithm*:*

```
1. Let $d_2$ be the degree of freedom (i.e., the number of independent variables).
2. Initialize storage vector $\mathbf{Y} = (0, \ldots, 0) \in \mathbb{Z}^{d_2}$.
3. Randomly sample a cell index $j$ from $1$ to $d_2$ without replacement. Do the following:
   (a) Find the lower bound $LB_j$ for $x_j$ by Algorithm 3.7.1.
   (b) Find the upper bound $UB_j$ for $x_j$ by Algorithm 3.7.2.
   (c) Sample $X_j^* \sim \text{Uniform}\left[LB_j, UB_j\right]$.
   (d) Set $Y_j = X_j^*$.
   (e) Update $A = A'$ and $\mathbf{b} = \mathbf{b}'$ by Algorithm 3.7.3.
   (f) Repeat Step 3 until all cells from 1 to $d_2$ are sampled.
4. If $\mathbf{Y}$ does not satisfy the original integer program,[7] then random sampling has encountered a hole, the table $\mathbf{Y}$ is rejected, and the algorithm returns to Step 2. If $\mathbf{Y}$ satisfies the original integer program, then go to Step 5.
5. Return table $\mathbf{X} = \mathbf{Y}$.
```

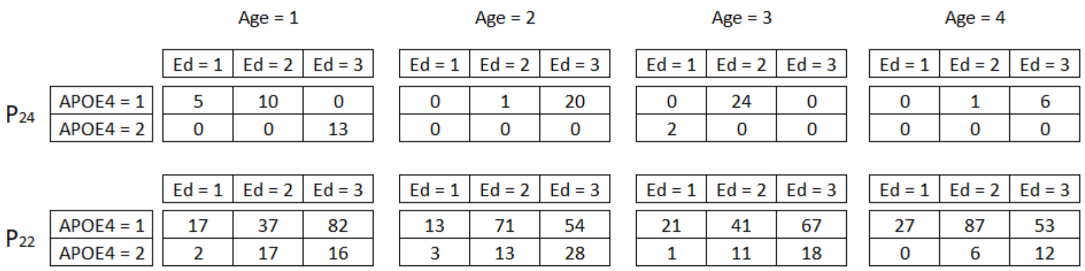
[7] $A \cdot \mathbf{Y} = \mathbf{b}, \quad \mathbf{Y} \geq \mathbf{0}$ and $\mathbf{Y} \in Z^{d_2}$.

After running Algorithm 3.7.4 once, we generate one contingency table ($\mathbf{x_1}$) existing in the conditional state space (sufficient statistic = $\mathbf{b}$).

$$A \cdot \mathbf{x_0} = \mathbf{b} = A \cdot \mathbf{x_1}. \tag{3.11}$$

Table 3.4 shows a randomly sampled contingency table ($\mathbf{x_1}$) from SIS.

Table 3.4. Example SIS Random Contingency Table ($\mathbf{x_1}$).

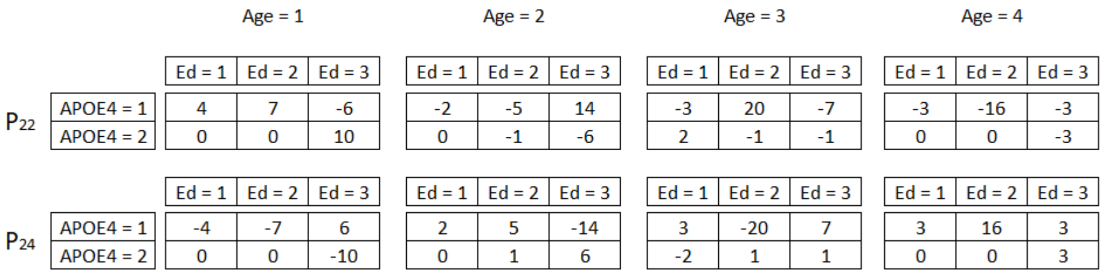


		Age = 1			Age = 2			Age = 3			Age = 4		
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P_{24}	APOE4 = 1	5	10	0	0	1	20	0	24	0	0	1	6
	APOE4 = 2	0	0	13	0	0	0	2	0	0	0	0	0
P_{22}	APOE4 = 1	17	37	82	13	71	54	21	41	67	27	87	53
	APOE4 = 2	2	17	16	3	13	28	1	11	18	0	6	12

This table shows cell counts from one SIS sample. It can be verified through Equation 3.11 that $\mathbf{x_1}$ satisfies the sufficient statistic.

Table 3.4 has substantially different cell counts from Table 3.1, the observed table. The differences in cell counts are shown in Table 3.5. Unlike the large cell count differences resulting from SIS, one iteration of MCMC only fractionally changes the contingency table. This is further discussed in Section 3.8.

Table 3.5. Difference in Cell Counts from a Single SIS Iteration.



		Age = 1			Age = 2			Age = 3			Age = 4		
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P_{22}	APOE4 = 1	4	7	-6	-2	-5	14	-3	20	-7	-3	-16	-3
	APOE4 = 2	0	0	10	0	-1	-6	2	-1	-1	0	0	-3
P_{24}	APOE4 = 1	-4	-7	6	2	5	-14	3	-20	7	3	16	3
	APOE4 = 2	0	0	-10	0	1	6	-2	1	1	0	0	3

This table shows differences in cell counts between the SIS contingency table ($\mathbf{x_1}$) and the observed contingency table ($\mathbf{x_0}$).

This section described a method to independently sample contingency tables from the conditional state space via SIS. Running Algorithm 3.7.4 N times generates N independent contingency tables ($\mathbf{X_i}$) satisfying the sufficient statistic ($\mathbf{b}$). In order to compensate for SIS not sampling tables from a hypergeometric distribution, MCMC is applied next.

3.8 Markov Chain Monte Carlo (MCMC)

MCMC methods facilitate hypergeometric sampling of the conditional state space. Each of the independent SIS contingency tables sampled in Section 3.7 serves as an independent starting point for an MCMC chain. We postulate that with sufficient independent starting points spread throughout the conditional state space, the connectivity problem of MCMC without a Markov basis is overcome.

Starting from the *current* contingency table, Algorithm 3.8.1 lists a sequence of steps to apply a random Markov *move* ($\mathbf{Z}$) to the table and randomly select whether to move to a new contingency table or stay at the current table.

Algorithm 3.8.1 *Metropolis-Hastings Algorithm Applied to Proposed MCMC Move.*

Input*: The starting table* $\mathbf{X_1}$ *and the sample size* N*. Log-linear model* F*. A set of Markov moves (*M*).*

Output*: A set of contingency tables sampled according to the hypergeometric distribution.*

Algorithm*:*

```
1. Set $R = \{\mathbf{X_1}\}$ (from SIS Algorithm 3.7.4).
2. For $i = 2, \cdots, N$:
    (a) Pick a move $\mathbf{Z} \in M$ uniformly.
    (b) Set proposal $\mathbf{X}^* = \mathbf{Z} + \mathbf{X_{i-1}}$.
    (c) Check that $\mathbf{X}^* \geq 0$. (Negative cell counts are prohibited.)
        i. If $\mathbf{X}^* < 0$, $\mathbf{X_i} = \mathbf{X_{i-1}}$. Proceed to Step 2f.
        ii. Else continue to Step 2d.
```

(d) *Compute the acceptance ratio*

$$r = \frac{p(\mathbf{X}^{*}|\mathbf{b})}{p(\mathbf{X_{i-1}}|\mathbf{b})}$$

where $p(\mathbf{X}|\mathbf{b})$ *is the probability mass function (pmf) of a hypergeometric distribution with fixed sufficient statistic* $\mathbf{b}$*. In the case of contingency tables, the acceptance ratio is analogous to the inverse ratio of the cell count factorials:*

$$r = \frac{\prod X_j! \quad \forall X_j \in \mathbf{X_{i-1}}}{\prod X_k! \quad \forall X_k \in \mathbf{X}^{*}}.$$

(e) *Set*

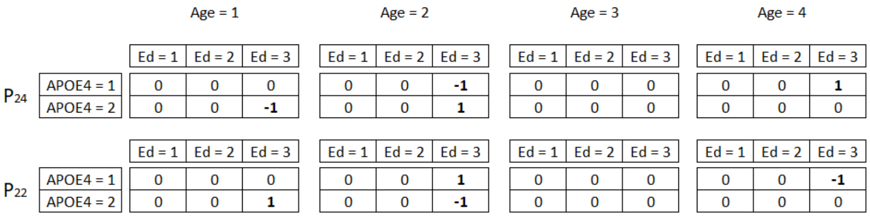
$$\mathbf{X_i} = \begin{cases} \mathbf{X}^{*} & \textit{with probability } \min(r, 1) \\ \mathbf{X_{i-1}} & \textit{else.} \end{cases}$$

(f) *Add element* $\mathbf{X_i}$ *to R.*

3. *Return R, the set of contingency tables from MCMC with MH acceptance applied.*

If it is possible to calculate a Markov basis, set M equal to the Markov basis. In most cases though, like our example, we are unable to enumerate all the elements (possible moves) in the Markov basis. Without a Markov basis, in Step 2a we instead generate a random, multidimensional $\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$ *move* similar to those proposed by Diaconis and Sturmfels (1998). The move must preserve the sufficient statistic $\mathbf{b}$ under H_0. Table 3.6 shows a random move $\mathbf{Z} \in M$. The sufficient statistic is preserved.

Table 3.6. Example Move ($\mathbf{Z}$).

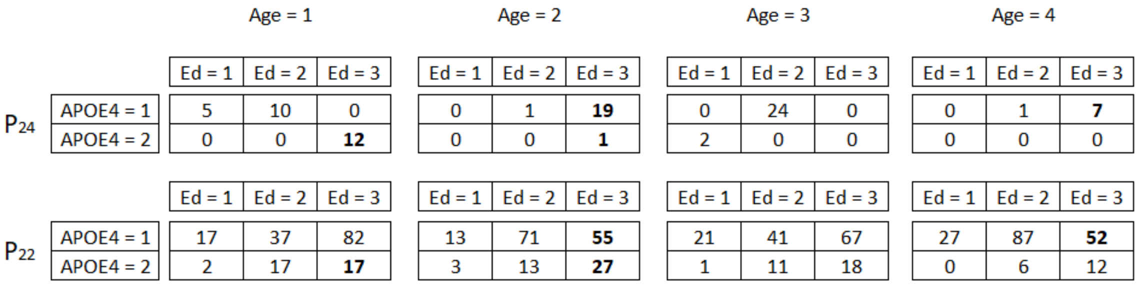


		Age = 1			Age = 2			Age = 3			Age = 4		
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P_{24}	APOE4 = 1	0	0	0	0	0	**-1**	0	0	0	0	0	**1**
	APOE4 = 2	0	0	**-1**	0	0	**1**	0	0	0	0	0	0
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P_{22}	APOE4 = 1	0	0	0	0	0	**1**	0	0	0	0	0	**-1**
	APOE4 = 2	0	0	**1**	0	0	**-1**	0	0	0	0	0	0

This table shows the $\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$ cell manipulations $\mathbf{Z}$ to be applied to $\mathbf{X_1}$. Compared to Table 3.5, one MCMC move results in smaller cell count differences than SIS.

The proposed contingency table $\mathbf{X}^*$ (from Step2b) is strongly dependent on the previous contingency table in the Markov chain, $\mathbf{X_{i-1}}$. Table 3.7 displays the proposed table ($\mathbf{X}^*$).

Table 3.7. Proposed Table ($\mathbf{X}^*$).



		Age = 1			Age = 2			Age = 3			Age = 4		
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P_{24}	APOE4 = 1	5	10	0	0	1	**19**	0	24	0	0	1	**7**
	APOE4 = 2	0	0	**12**	0	0	**1**	2	0	0	0	0	0
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P_{22}	APOE4 = 1	17	37	82	13	71	**55**	21	41	67	27	87	**52**
	APOE4 = 2	2	17	**17**	3	13	**27**	1	11	18	0	6	12

This table shows a proposed contingency table from MCMC (Step 2b of Algorithm 3.8.1).

After generating the proposed table $\mathbf{X}^*$ and checking that all cell counts are non-negative, Step 2d determines the acceptance ratio (r). This ratio measures the relative probabilities of the two contingency tables, $\mathbf{X}^*$ and $\mathbf{X_{i-1}}$. For contingency tables, the probability ratio is the same as the inverse ratio of the cell count factorials for each of the two contingency tables. In the case of our example, $\mathbf{X}^*$ (Table 3.7) is more likely than $\mathbf{X_1}$ (Table 3.4). The acceptance ratio r is therefore greater than 1. From Step 2e, we accept the proposed table and set $\mathbf{X_2} = \mathbf{X}^*$. Step 2f adds $\mathbf{X_2}$ to the set $R = \{\mathbf{X_1}, \mathbf{X_2}\}$.

Depending on the proposed table $\mathbf{X}^*$, there are two additional scenarios to the situation presented in the preceding paragraph (accepting $\mathbf{X}^*$ with probability 1). If $\mathbf{X}^* < 0$, meaning *any* cell $X_i^* \in \mathbf{X}^*$ is negative, then $\mathbf{X}^*$ is rejected and $\mathbf{X_i} = \mathbf{X_{i-1}}$. Proposed table $\mathbf{X}^*$ could also be less probable than current table $\mathbf{X_{i-1}}$. In this case, $0 < r < 1$ and table $\mathbf{X}^*$ will be accepted with probability r.

The process in Step 2 of Algorithm 3.8.1 is repeated until an MCMC chain R is returned containing N contingency tables. One connected MCMC chain is generated from each SIS table sampled.

3.9 Hybrid MCMC and SIS Sampling Algorithm

Algorithm 3.9.1 combines SIS and MCMC to generate an approximate, hypergeometric distribution of test statistics ($\mathbf{T}$) given $\mathbf{x_0}$ and $\mathbf{b}$.

Algorithm 3.9.1 *Approximate Hypothesis Test using MCMC and SIS Sampling.*

Input*: The observed contingency table* $\mathbf{x_0}$ *and sufficient statistic* $\mathbf{b}$. *The desired number of SIS samples K. The sample size N for each MCMC chain. The number of burn-in samples B. The thinning interval Q. Model for H_0 (F_0) and model for H_1 (F_1).*

Output*: An empirical distribution of log-likelihood ratio test statistics* $(\mathbf{T}(\mathbf{X}))$ *from tables in the conditional state space sampled according to a hypergeometric distribution.*

Algorithm*:*

1. *SIS: With Algorithm 3.7.4, independently sample K starting tables. The sample is denoted as* $\{\mathbf{X_{11}}, ..., \mathbf{X_{1K}}\}$.
2. *For each $k = 1, \cdots, K$:*
   (a) *MCMC: Sample N many tables with Algorithm 3.8.1 and starting table* $\mathbf{X_{1k}}$. *The Markov chain* $\mathbf{R_k} = \{\mathbf{X_{1k}}, ..., \mathbf{X_{Nk}}\}$.
   (b) *Initialize the vector of test statistics* $\mathbf{T_k} = \emptyset$.
   (c) *Burn-In: Eliminate the first B tables from* $\mathbf{R_k}$ *to minimize dependence of MCMC on SIS starting point* $\mathbf{X_{1k}}$. *$m = B + 1$.*
   (d) *While $m \leq N$:*
      i. *Use sampled contingency table* $\mathbf{X_{mk}}$ *from* $\mathbf{R_k}$.
      ii. *Compute the maximum log likelihood (residual deviance) under H_0 model.*
      iii. *Compute maximum log likelihood (residual deviance) under H_1 model.*
      iv. *Compute the log-likelihood ratio test statistic (G^2) as the difference in the residual deviance (refer to Section 3.4).*
      v. *Add G^2 to* $\mathbf{T_k}$.
      vi. *Thinning: Advance m by Q, the thinning interval to minimize dependence between successive test statistics.*
   (e) *Return* $\mathbf{T_k}$.
3. *The test statistic distribution is approximated by combining test statistics from all K Markov chains.* $\mathbf{T} = \{\mathbf{T_1}, ..., \mathbf{T_K}\}$.

To complete the example initiated in Section 3.3, Algorithm 3.9.1 is run with parameters: $K = 100$ SIS tables, MCMC chain length $N = 4400$, $B = 400$ burn-in tables eliminated from beginning of each MCMC chain, record test statistics for every $Q = 20$ tables sampled. To sample 440000 contingency tables and test statistics (4400 MCMC iterations for 100 SIS starting points) requires approximately 2 hours of run time (`Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz and 16GB RAM`). Applying *burn-in* removes the first 400 samples from each Markov chain. After thinning, a sample size of 20000 test statistics ($\mathbf{T}$) remains: 200 test statistics ($\mathbf{T_k}$) from each MCMC chain emanating from the 100 SIS starting points. In the final step, the distribution of test statistics $\mathbf{T}$, along with the observed test statistic from Section 3.4, is used to calculate an estimated p-value. Using the proposed hybrid sampler, the estimated p-value for the example goodness-of-fit hypothesis test is 0.008. Chapter 4 further expands upon this result.

3.10 Summary

Figure 3.2 summarizes a notional flow chart from observed contingency table data ($\mathbf{x_0}$) to hypothesis test result. Since asymptotic and exact methods are not guaranteed to be valid for sparse, multidimensional contingency tables, this chapter proposed a method for estimating the null distribution of test statistics under H_0 using a sampler combining SIS and MCMC.

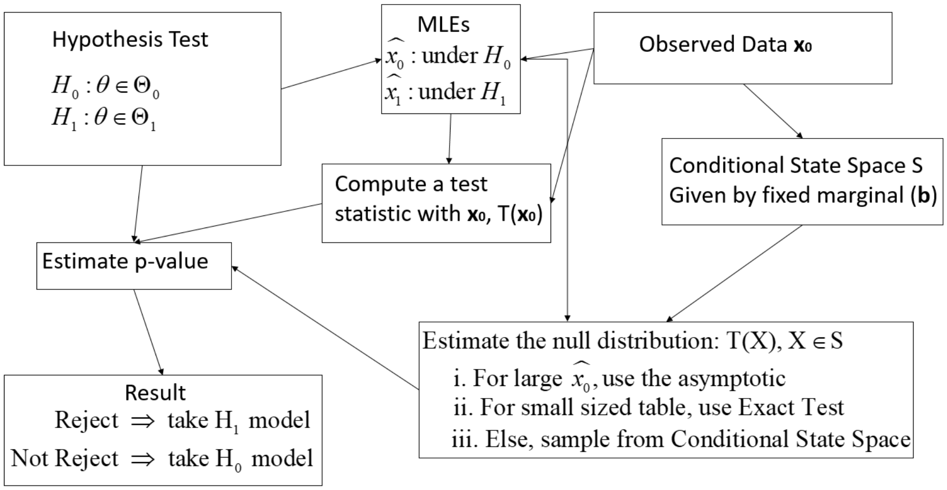

Hypothesis Test
$H_0 : \theta \in \Theta_0$
$H_1 : \theta \in \Theta_1$
MLEs
$\widehat{x_0}$: under H_0
$\widehat{x_1}$: under H_1
Observed Data $\mathbf{x_0}$
Conditional State Space S
Given by fixed marginal (**b**)
Compute a test
statistic with $\mathbf{x_0}$, T($\mathbf{x_0}$)
Estimate p-value
Estimate the null distribution: T(X), X ∈ S
i. For large $\widehat{x_0}$, use the asymptotic
ii. For small sized table, use Exact Test
iii. Else, sample from Conditional State Space
Result
Reject ⇒ take H_1 model
Not Reject ⇒ take H_0 model

Figure 3.2. Notional Flow Chart for Goodness-of-Fit Testing.

This figure shows the steps necessary to perform a goodness-of-fit test given null (H_0) and alternative (H_1) models and observed contingency table data ($\mathbf{x_0}$).

In Section 3.3.1, Equations 3.2 and 3.3 propose null and alternative hypotheses to test the statistical significance of factor Y_3 (age) on particular state transition ratios $\left(\log\left(\frac{P_{24}}{P_{22}}\right)\right)$ in the presence of Y_1 (APOE-4 allele) and Y_2 (education). To evaluate the hypotheses, we extract the relevant observed contingency table data ($\mathbf{x_0}$) from the nun cognitive observational dataset in Section 3.3.2.

The observed table ($\mathbf{x_0}$) and null and alternative hypotheses allow for the calculation of MLEs. With the MLEs, Section 3.4 describes computing the maximum log-likelihood ratio test statistic $\left(T_{\mathbf{x_0}}\right)$ from the difference in the residual deviance of the two models.

The sufficient statistic ($\mathbf{b}$) governing the conditional state space ($S_{\mathbf{b}}$) is calculated from the Lawrence lifting of the modified Segre product in Section 3.5. Hara et al. (2010) showed this sufficient statistic is the minimal necessary sufficient statistic for the bivariate logistic regression model (H_0). After a discussion of p-value estimation from a distribution of test statistics, Sections 3.7 and 3.8 detail SIS and MCMC methods of sampling from the conditional state space.

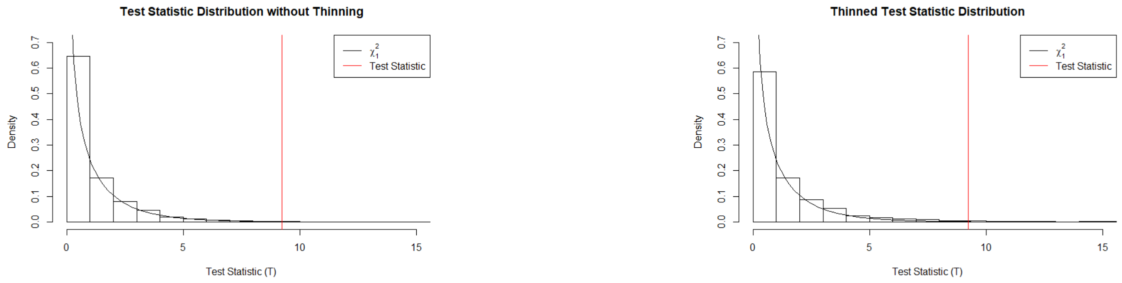
Section 3.9 combines the independent sampling of SIS with the hypergeometric sampling of MCMC to approximate the test statistic distribution ($\mathbf{T}(\mathbf{X})$). The established techniques of *burn-in* and *thinning* are employed to reduce the correlation among tables sampled using MCMC. Applying Algorithm 3.9.1 enables us to approximate the null distribution of test statistics $\mathbf{T}(\mathbf{X})$. Finally, with an observed test statistic $T_{\mathbf{x_0}}$ and a distribution of test statistics $\mathbf{T}(\mathbf{X})$, an approximate p-value is estimated.

CHAPTER 4:
Results and Insights

In this chapter, we apply the sampler described in Chapter 3 to conduct 27 goodness-of-fit hypothesis tests on the nun cognitive dataset. Each hypothesis test is performed using three different variations of the SIS algorithm. The chief observations, results, and insights from the employment of the hybrid scheme are discussed.

4.1 Analysis of Example Problem

Figure 4.1 presents sampled test statistic distribution ($\mathbf{T}(\mathbf{X})$) histograms generated from applying the hybrid sampler (Algorithm 3.9.1) to the example hypothesis test in Chapter 3 ($s = 2, v = 2, i = 3$). In Figure 4.1a, burn-in is applied ($B = 400$) to each Markov chain without any subsequent thinning ($Q = 1$), yielding a sample size of 400000. Figure 4.1b has both burn-in and thinning ($Q = 20$) applied, reducing the sample size to 20000 test statistics. For comparison to the asymptotic approximation, a chi-squared density line with one degree of freedom (the difference in model parameters between H_0 and H_1) is overlayed on each histogram. The observed test statistic is shown in red. The similarity between thinned and unthinned samples calls into question whether the reduction in auto-correlation from thinning justifies the loss of information (Link and Eaton 2012).



(a) No Thinning

(b) Thinned Sample

Figure 4.1. Approximate Test Statistic Distribution: $s = 2, v = 2, i = 3$.

The histograms depict a sampled distribution of test statistics $T(\mathbf{X})$ from the hybrid sampling scheme. Figure 4.1a has no thinning applied. Figure 4.1b has burn-in *and* a thinning interval of 20, resulting in 1/20 the sample size.

The p-value is estimated by counting the proportion of test statistics from the distribution of test statistics ($\mathbf{T}(\mathbf{X})$) that are as extreme or more extreme (larger) than the observed test statistic ($T_{\mathbf{x_0}}$) of 9.23 (Step 4 of Algorithm 3.6.1). As expected from the similarity of their approximate distributions, using either the thinned and unthinned distribution of test statistics gives similar p-value estimates. Without thinning, we calculate a p-value of 0.0075. Using the thinned sampling distribution, the p-value is 0.0076. Employing the common asymptotic assumption for contingency tables that the log-likelihood ratio test statistic follows a χ^2 distribution, the estimated p-value is 0.0024. Although the asymptotic p-value is roughly three times smaller than the p-value estimated from hybrid sampling, the result of the hypothesis test at the $\alpha = 0.01$ level is still to reject H_0 in favor of H_1.

Despite the observed data being a sparse contingency table with 21 of 48 expected cell counts less than 5 (see Table 3.2), the distribution of test statistics from the hybrid SIS and MCMC scheme closely resembles a χ^2 distribution. The tail of the approximate distribution is slightly fatter that the asymptotic tail but not obviously so. The asymptotic distribution assumption seems to be robust, even for sparse contingency tables. This result coincides with guidance from Conover (1999) for the relaxation of conditions to apply the asymptotic assumption to contingency table distributions.

4.2 Hole Example

Staying with the example problem, this section briefly presents one occurrence of a *hole*, a table rejected because there is no longer an integer feasible solution.

Suppose we have the Lawrence lifting matrix A and the sufficient statistic $\mathbf{b}$ from Section 3.5. We proceed through the SIS algorithm (Algorithm 3.7.4) by randomly selecting cell j and then uniformly sampling an integer X_j^*. Further suppose we have sequentially sampled the 28 cells shown in Table 4.1.

After several iterations of Algorithm 3.7.3, A and $\mathbf{b}$ are reduced to yield Equation 4.1 of the form $A' \cdot \mathbf{x} = \mathbf{b}'$, $\mathbf{x} \geq 0$. Although Equation 4.1 is feasible, there is no integer solution. The sampling path delineated in Table 4.1 results in a *hole*. This table is discarded. The SIS algorithm starts again from the beginning, increasing the computational time.

Table 4.1. Cell (X_j) Sampling Sequence Resulting in a Hole.

j (random column)	LB_j	UB_j	x_j^*
42	3	18	13
4	0	10	2
40	3	11	9
7	0	13	4
36	22	28	22
18	5	5	5
33	34	72	43
34	13	13	13
37	17	21	20
20	0	0	0
3	0	7	1
2	0	0	0
30	29	29	29
13	1	1	1
5	0	29	17
43	24	27	26
38	3	3	3
9	29	29	29
44	0	0	0
27	46	46	46
8	0	0	0
24	0	0	0
32	3	3	3
22	0	0	0
12	6	6	6
16	2	2	2
39	61	65	63
45	86	88	87

This table shows a notional random sampling sequence for performing SIS given the observed contingency table $\mathbf{x_0}$ (Table 3.1) and null hypothesis (Equation 3.2).

$$\begin{bmatrix}
1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0\\
1&2&2&1&2&1&1&1&1&1&0&0&0&0&0&0&0&0&0&0\\
1&3&2&3&1&2&3&1&2&3&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}
\mathbf{x} =
\begin{bmatrix}
15\\15\\39\\22\\2\\0\\15\\65\\0\\9\\0\\0\\0\\74\\0\\0\\0\\2\\0\\67\\0\\1\\0\\1\\6\\59\\12
\end{bmatrix}. \tag{4.1}$$

4.3 Discrete Distributions

Because of the integer requirement, all contingency table data produce discrete distributions of test statistics. With enough possible contingency tables in the conditional state space, the discrete distribution appears almost continuous as was shown in Figure 4.1. The nun cognitive dataset contains only five transitions from state 1 (intact cognition) to state 4 (dementia). As a result of few observations, the nine hypothesis tests in which $s = 1$ are based on an observed contingency table that is even more sparse than $s = 2$ or $s = 3$ hypothesis tests. The small values contained in the sufficient statistic (a function of the low P_{14} transitions) limit the number of feasible contingency tables. The resulting distribution of test statistics does not resemble a smooth, contiguous distribution as was the case in Section 4.1.

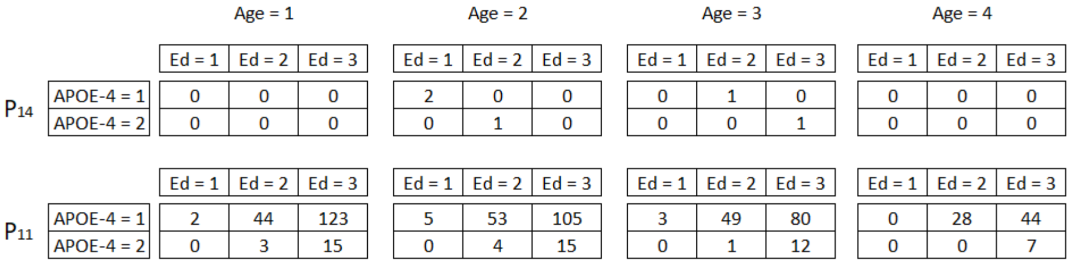
Consider the same proposed hybrid sampling scheme for conducting goodness-of-fit tests, but now applied to $s = v = i = 1$. The null and alternative hypotheses are:

$$H_0^{111} : \log\left(\frac{P_{14}}{P_{11}}\right) = \alpha_{11} + \beta_{211}Y_2 + \beta_{311}Y_3, \tag{4.2}$$

$$H_1^{111} : \log\left(\frac{P_{14}}{P_{11}}\right) = \alpha_{11} + \beta_{111}Y_1 + \beta_{211}Y_2 + \beta_{311}Y_3. \tag{4.3}$$

The observed contingency table from the nun dataset is presented as Table 4.2. There are only 5 total transitions from state 1 to 4 (P_{14}), and $P_{14} = 0$ for $Age = 1$ and $Age = 4$. The first three elements of the sufficient statistic (**b**) are: 5 $\left(\sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{4} X_{1ijk}\right)$, 9 $\left(\sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{4} j \cdot X_{1ijk}\right)$, and 12 $\left(\sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{4} k \cdot X_{1ijk}\right)$ (Section 3.5). Relative to the example discussed in Section 4.1, there are far fewer feasible contingency tables that can satisfy these first three elements of the sufficient statistic.

Table 4.2. Extremely Sparse Observational Data ($\mathbf{x_0}$): $s = v = i = 1$.



		Age = 1			Age = 2			Age = 3			Age = 4		
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P14	APOE-4 = 1	0	0	0	2	0	0	0	1	0	0	0	0
	APOE-4 = 2	0	0	0	0	1	0	0	0	1	0	0	0
		Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3	Ed = 1	Ed = 2	Ed = 3
P11	APOE-4 = 1	2	44	123	5	53	105	3	49	80	0	28	44
	APOE-4 = 2	0	3	15	0	4	15	0	1	12	0	0	7

This table gives observational transitions for P_{14} and P_{11}. There are very few (5) P_{14} transitions, limiting the number of feasible contingency tables in the conditional state space.

Figure 4.2 shows a histogram of test statistics corresponding to the initial distribution of 100 SIS starting points before MCMC is applied. There are only five feasible contingency tables for this hypothesis test, including the observed table. In this instance, SIS has nearly uniformly sampled the left three contingency tables.

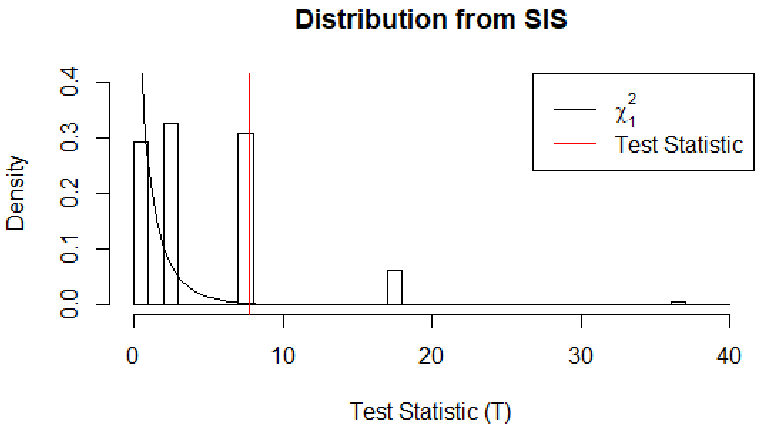
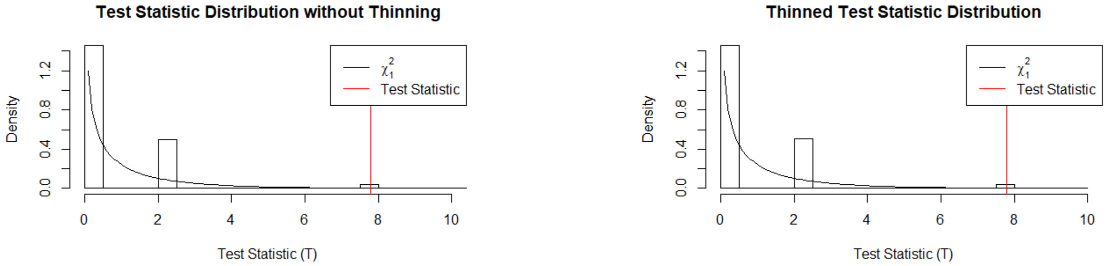


Figure 4.2. SIS-Only Test Statistic Distribution: $s = v = i = 1$.

Few feasible contingency tables exist. After SIS, applying MCMC sampling will increase probability of tables closer to MLE (small test statistic) and reduces probability of contingency tables far from MLE.

With few feasible tables, it is impossible for the sampled distribution to approximately match a continuous asymptotic distribution. Figure 4.3 compares the approximate distribution from the hybrid sampling approach with the asymptotic distribution. Only three contingency tables have non-trivial probability. Again, we observe little difference between the thinned and unthinned sampling distributions. The effect of MH and MCMC in favoring transitions to tables with higher likelihood can be seen by comparing Figure 4.2 with Figure 4.3. The estimated p-value based on hybrid sampling is 0.005, and the p-value from asymptotic inference is 0.005. Both methods would reject H_0 in favor of H_1 at $\alpha = 0.01$.



(a) No Thinning

(b) Thinned Sample

Figure 4.3. Approximate Test Statistic Distribution: $s = v = i = 1$.

Although this histogram does show a decaying probability, there are too few contingency tables satisfying the sufficient statistic to produce a histogram resembling a continuous distribution. The sample sizes are 400000 and 20000, respectively.

4.4 Hypergeometric and Triangle Distributions

Up to this point, we have exclusively used uniform sampling within the SIS algorithm to randomly generate individual cell counts. We also modify Step 3c of Algorithm 3.7.4 to experiment with triangle and hypergeometric cell sampling and compare the resultant distribution of SIS test statistics. Assuming a uniform, hypergeometric, or triangle distribution within SIS does not affect MCMC sampling. Only the SIS table that begins an MCMC chain is affected by the distribution choice.

For the hypergeometric distribution, a random sample is drawn such that $X_j^* \sim \text{Hypergeometric}(K, m, n)$. K = lower bound (LB) + upper bound (UB) = the number of balls to be drawn from the urn. $m = UB$ = the number of white balls in the urn.

$n = UB$ = the number of black balls in the urn. Therefore, the minimum white balls drawn from the urn will be LB and the maximum will be UB. Compared to the uniform distribution that has equal probability for each integer, hypergeometric distribution results in more random draws from the middle of the interval $[LB, UB]$ than from the boundaries.

For the triangle distribution, a random sample is drawn such that $X_j^* \sim \text{Triangle}(a, b, c)$, where a = minimum = LB, b = maximum = UB, and c = mode, which we take to be the maximum likelihood estimate. If $a > c$, we reset the distribution minimum to the mode ($a = c$); if $b < c$, we reset the maximum to the mode ($b = c$). Unlike hypergeometric, the triangle distribution is a continuous distribution. To adequately weight each integer, we add a continuity correction by reducing a by 0.50 and increasing b by 0.49. The resulting random draw X_j^* is rounded to the nearest integer within $[LB, UB]$. If $X_j^* < LB$, $X_j^* = LB$. If $X_j^* > UB$, $X_j^* = UB$. In this way, for maximum likelihood estimates outside the possible range of X_j^*, the high probability of drawing a number near the maximum likelihood estimate is added entirely to the nearest bound rather that spread evenly over the interval.

To compare uniform, triangle, and hypergeometric sampling schemes, we reconstruct the example problem with $s = 2$, $v = 2$, and $i = 3$ and replace the uniform distribution with hypergeometric and triangle distributions (in Step 3c of SIS Algorithm 3.7.4). Figure 4.4 shows a histogram of the 100 SIS starting points from each sampling method. For this hypothesis test, there are no discernible differences in the distribution of SIS tables.

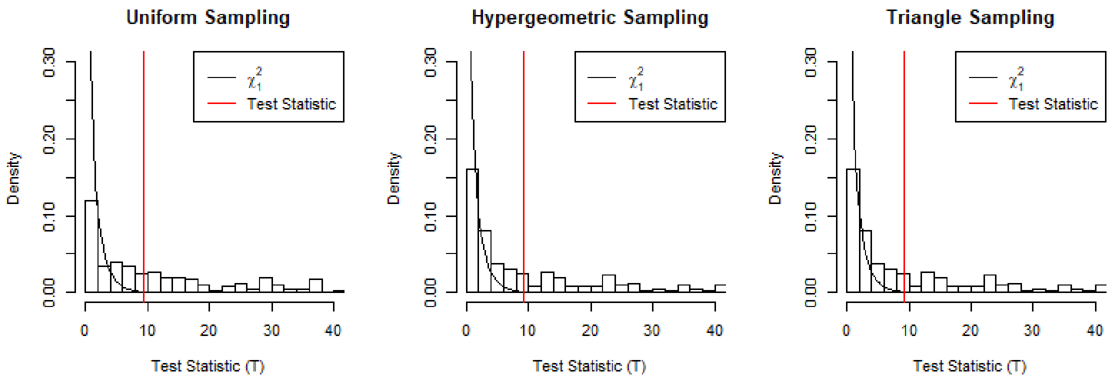


Figure 4.4. Initial SIS Test Statistics: $s = 2$, $v = 2$, $i = 3$.

100 independent contingency tables from SIS. The sampling type indicates the assumed distribution when sampling individual cells with bounds from IP.

After applying an MCMC algorithm to each of the independent starting tables, the distribution of test statistics for each scheme is shown in Figure 4.5. Since the SIS portion of sampling yielded similar results and the MCMC algorithm is identical in each case, it is not surprising the three different sampling methods produce similar distributions. The approximate p-values are 0.0076, 0.0030, and 0.0074 for uniform, hypergeometric, and triangle sampling, respectively. The p-value for an asymptotic approximation is 0.0024. Comparing Figure 4.4 to Figure 4.5, the SIS algorithm samples more from large test statistic values in the right tail of the distribution than MCMC with MH applied.

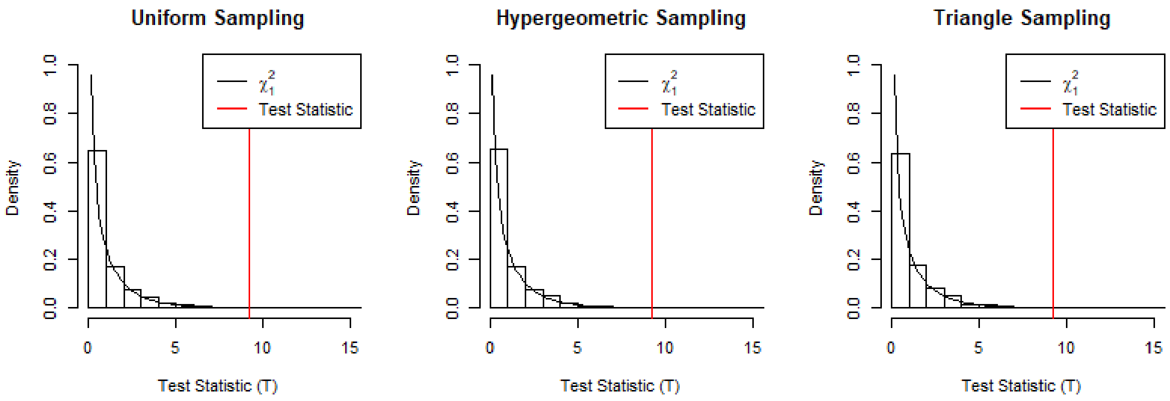


Figure 4.5. Sampling Method Comparison: $i = 3$, $s = 2$, $v = 2$.

Uniform, Triangle, and Hypergeometric sampling methods for SIS all produce similar distributions and p-values from the hybrid SIS and MCMC sampling scheme. After burn-in and thinning, the histogram sample size is 20000.

To better illustrate the effect of a sampling scheme within the SIS algorithm, we reproduce Figure 4.2 for testing the hypothesis for $s = v = i = 1$ but with hypergeometric and triangle sampling schemes also used in the SIS algorithm. The three most frequently sampled tables via SIS are displayed. Recall from Section 4.3 that this hypothesis test and corresponding sufficient statistic had few feasible contingency tables. Figure 4.6 shows that in this instance, uniform sampling of cells produces a roughly uniform distribution of contingency tables across the three possibilities. Hypergeometric sampling has higher probability on the middle table, and triangle sampling places more probability on the table with the lowest test statistic. This illustrates that different sampling methods for SIS can produce different distributions of contingency tables. The sampled distribution, however, after the MCMC portion of the hybrid sampler has been applied is almost identical for all three methods (see Figure B.1 in Appendix B).

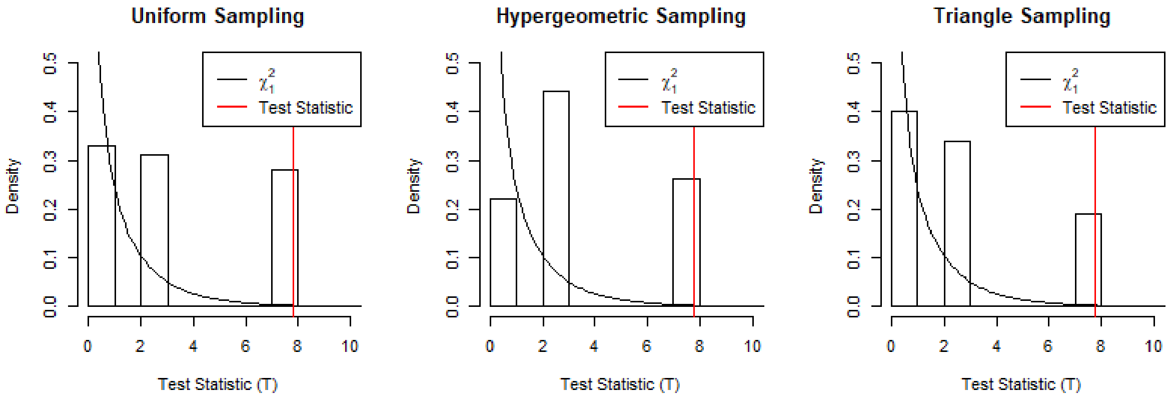


Figure 4.6. Initial SIS Test Statistics: $i = s = v = 1$.

There are few SIS initial starting points regardless of the sampling method. The SIS sample size for each histogram is 200.

4.5 Goodness-of-Fit Test Results

The following sampling parameters are used for implementing the SIS/MCMC hybrid sampling scheme (Algorithm 3.9.1). $K = 100$ independent SIS sampled tables. $N = 4400$ contingency tables in each MCMC chain, beginning with an initial table from SIS. $B = 400$ burn-in tables eliminated from beginning of each MCMC chain, and we record test statistics for every $Q = 20$ tables sampled. An approximate test statistic distribution ($\mathbf{T}(\mathbf{X})$) is generated for all $s \in \{1, 2, 3\}$, $v \in \{1, 2, 3\}$, and $i \in \{1, 2, 3\}$ combinations and for the three distribution assumptions of uniform, hypergeometric, and triangle. After burn-in and thinning, each approximate test statistic distribution for a given (s, v, i) has a sample size of 20000 test statistics.

Recall the null and alternative hypotheses being tested via a goodness-of-fit hypothesis test for i, s, v:

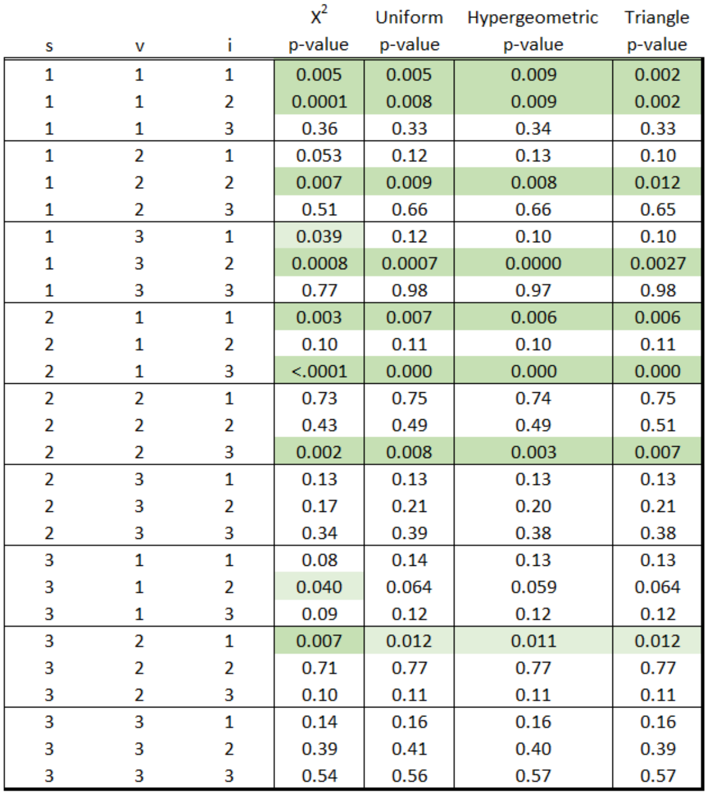
$$\log\left(\frac{P_{s4}}{P_{sv}}\right) = \alpha_{sv} + \beta_{1sv}Y_1 + \beta_{2sv}Y_2 + \beta_{3sv}Y_3.$$

$$H_0^{isv} : \beta_{isv} = 0,$$

$$H_1^{isv} : \beta_{isv} \neq 0.$$

The results of the 27 goodness-of-fit tests are summarized in Table 4.3. The test statistic distributions from which the estimated p-values are derived are presented in Appendix B.

Table 4.3. Goodness-of-Fit Test Results for (s, v, i).

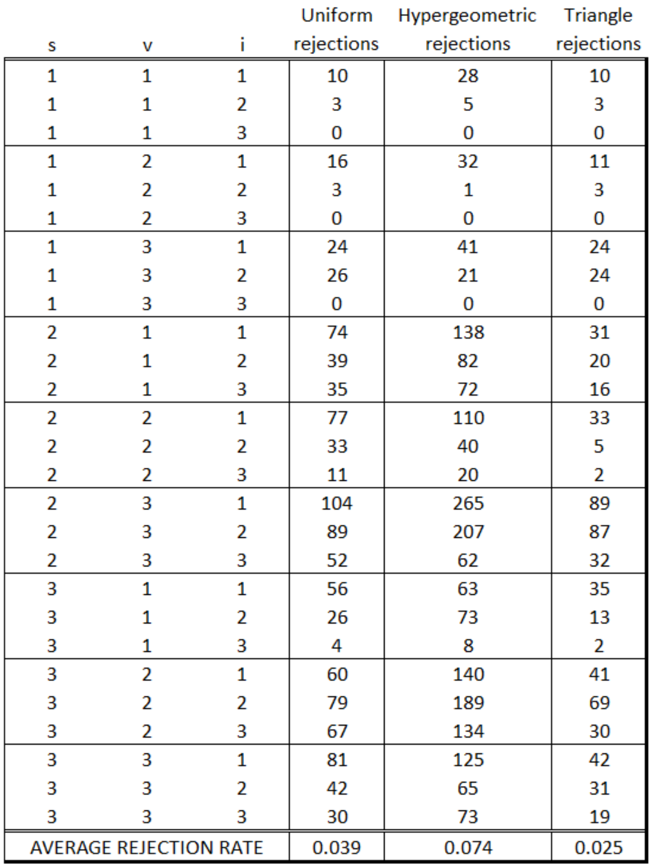


			X^2	Uniform	Hypergeometric	Triangle
s	v	i	p-value	p-value	p-value	p-value
1	1	1	0.005	0.005	0.009	0.002
1	1	2	0.0001	0.008	0.009	0.002
1	1	3	0.36	0.33	0.34	0.33
1	2	1	0.053	0.12	0.13	0.10
1	2	2	0.007	0.009	0.008	0.012
1	2	3	0.51	0.66	0.66	0.65
1	3	1	0.039	0.12	0.10	0.10
1	3	2	0.0008	0.0007	0.0000	0.0027
1	3	3	0.77	0.98	0.97	0.98
2	1	1	0.003	0.007	0.006	0.006
2	1	2	0.10	0.11	0.10	0.11
2	1	3	<.0001	0.000	0.000	0.000
2	2	1	0.73	0.75	0.74	0.75
2	2	2	0.43	0.49	0.49	0.51
2	2	3	0.002	0.008	0.003	0.007
2	3	1	0.13	0.13	0.13	0.13
2	3	2	0.17	0.21	0.20	0.21
2	3	3	0.34	0.39	0.38	0.38
3	1	1	0.08	0.14	0.13	0.13
3	1	2	0.040	0.064	0.059	0.064
3	1	3	0.09	0.12	0.12	0.12
3	2	1	0.007	0.012	0.011	0.012
3	2	2	0.71	0.77	0.77	0.77
3	2	3	0.10	0.11	0.11	0.11
3	3	1	0.14	0.16	0.16	0.16
3	3	2	0.39	0.41	0.40	0.39
3	3	3	0.54	0.56	0.57	0.57

For each (s, v, i) combination, the estimated p-value from the SIS (using either uniform, hypergeometric, or triangle distribution assumptions) and MCMC sampler is reported. The p-value using a χ_1^2 asymptotic assumption is also stated. Light shading indicates statistical significance at the $\alpha = 0.05$ level. Dark shading indicates statistical significance at the $\alpha = 0.01$ level.

Rejection data presented in Table 4.4 was gathered by sampling 1000 independent SIS tables for each (s, v, i) combination and distribution assumption. Depending on the (s, v, i) combination and distribution assumption, rejections for 1000 SIS tables ranged from 0 to 265. Triangle sampling had the lowest average rejection rate at 2.5 rejections per 100 tables, followed by uniform sampling at 3.9 rejections per 100 tables. Hypergeometric sampling, which samples more frequently from the center of the lower and upper bounds, had 7.4 rejections per 100 tables, roughly twice the rate of uniform and hypergeometric. We observe that when one sampling method had a large number of rejections, the other sampling methods also tended to have many rejections. For instance, $s = 2$, $v = 3$, $i = 1$ yielded the most rejections of any case for all three distribution schemes.

Table 4.4. Rejections by Interval Distribution Sampling Scheme for (s, v, i).



s	v	i	Uniform rejections	Hypergeometric rejections	Triangle rejections
1	1	1	10	28	10
1	1	2	3	5	3
1	1	3	0	0	0
1	2	1	16	32	11
1	2	2	3	1	3
1	2	3	0	0	0
1	3	1	24	41	24
1	3	2	26	21	24
1	3	3	0	0	0
2	1	1	74	138	31
2	1	2	39	82	20
2	1	3	35	72	16
2	2	1	77	110	33
2	2	2	33	40	5
2	2	3	11	20	2
2	3	1	104	265	89
2	3	2	89	207	87
2	3	3	52	62	32
3	1	1	56	63	35
3	1	2	26	73	13
3	1	3	4	8	2
3	2	1	60	140	41
3	2	2	79	189	69
3	2	3	67	134	30
3	3	1	81	125	42
3	3	2	42	65	31
3	3	3	30	73	19
AVERAGE REJECTION RATE			0.039	0.074	0.025

For each (s, v, i) combination, the number of rejections resulting from generating 1000 SIS starting points via Algorithm 3.7.4.

4.6 Insights

This section presents multiple insights distilled from our research and application to the nun cognitive observational dataset.

4.6.1 Asymptotic Assumption

The hybrid sampling scheme proposed in Chapter 3 produces estimated p-values similar to the asymptotic approximation p-values (refer to Table 4.3). The hybrid sampling method is generally more conservative with larger p-values. This result occurs despite the contingency tables being sparse and, in the case of $s = 1$, producing test statistic distributions with only a handful of possible values. We infer this as evidence of the applicability of the χ^2 asymptotic approximation even to relatively sparse contingency tables. Contrary to our initial conjectures, the asymptotic approximation works well for this dataset of sparse, multidimensional contingency tables.

4.6.2 Run Time

Each individual hypothesis test for one sampling method (uniform, hypergeometric, or triangle) took approximately 50 minutes to sample 100 SIS starting tables and 4400 MCMC iterations for each starting table. Individual cases ranged in time from around 30 minutes to 2 hours. To run all 27 hypothesis tests with 3 different sampling methods requires about 3 days of run time on a personal computer. Invoking the asymptotic approximation enables all of these tests to be run in less than a second, offering a huge time advantage over the proposed hybrid sampler.

4.6.3 Holes

In establishing the lower and upper bounds of the feasible region for cell X_j^*, we solve two IP systems using a minimum and a maximum objective function. Therefore, both the lower bound (minimum) and upper bound (maximum) are known to be integer feasible solutions to the linear program and not holes. As more and more X_j^* cells become fixed through sampling and the solution set gets smaller, the interval between lower and upper bounds for X_j^* narrows. If a hole still exists in the interval, it becomes increasingly likely to be encountered as the sampling range shrinks.

The hypergeometric distribution samples more frequently near the middle of the range (relative to uniform and triangle), and we observe hypergeometric sampling locating holes more frequently (refer to Table 4.4). Uniform distribution has equal sampling probability for each integer in the interval, including the lower and upper bounds, which are known to be non-holes. If the mode (based on MLE) lies outside the $[LB, UB]$ interval, we constructed our triangle distribution to add all the probability lying outside of the bounds to the nearest boundary. Consequently, the triangle distribution method we employ has an even greater probability than the uniform distribution of sampling from one of the two bounds. As a result, we observe triangle sampling has the fewest rejections.

4.6.4 Sampling Method

From the perspective of estimated p-values, uniform, hypergeometric, or triangle distribution assumptions for the sampling a cell X_j^* have little effect on the estimated p-values. Most of the sampling distribution differences between the three methods are eliminated after MCMC has been run for 4400 iterations and the initial 400 tables (which includes the SIS table) are removed by burn-in.

4.6.5 Application to Cognitive Impairment

We broadly interpret the results for the eight cases in which H_0 is rejected in favor of H_1 at the $\alpha = .01$ level. APOE-4 allele, education, and age all show up as statistically significant in the context of our goodness-of-fit test for different combinations of s, v, i. Table 4.5 displays coefficient estimates for each case where H_0 was rejected at the $\alpha = 0.01$ level.

Education ($i = 2$) appears to have the greatest effect when the prior cognitive status is intact cognition. We conclude education is a significant factor in predicting the relative transition rate from intact cognition to either intact cognition, MCI, or GI relative to transition from intact cognition to dementia. In each case, higher education is associated with a lower odds ratio ($\beta_2 < 0$) of transitioning to dementia relative to transitioning to a less severe state.

Age ($i = 3$) is found to be significant in the relative transition rates to dementia where previous cognitive status is MCI and current status is intact cognition or MCI. Older nuns are more likely to transition from MCI to dementia ($\beta_3 > 0$) compared to younger nuns.

Table 4.5. Maximum Likelihood Estimates of β_i for (s, v, i).

s	v	i	β_1 (APOE4)	β_2 (Ed)	β_3 (Age)
1	1	2	3.68***	**-3.28*****	0.53
			(1.29)	(0.97)	(0.58)
1	2	2	2.18**	**-2.15****	0.30
			(1.07)	(0.89)	(0.45)
1	3	2	2.88*	**-3.34****	0.21
			(1.51)	(1.37)	(0.75)
2	1	3	1.28***	-0.38*	**0.68*****
			(0.43)	(0.23)	(0.13)
2	2	3	0.11	0.14	**0.33*****
			(0.31)	(0.18)	(0.11)
1	1	1	**3.68*****	-3.28***	0.53
			(1.29)	(0.97)	(0.58)
2	1	1	**1.28*****	-0.38*	0.68***
			(0.43)	(0.23)	(0.13)
3	2	1	**1.40****	0.12	0.32
			(0.57)	(0.31)	(0.19)

$^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$ t-statistic

This table reports Maximum Likelihood parameter Estimates for H_0 and H_1 given $\mathbf{x_0}$. Standard errors for each estimate are shown in parentheses. Coefficient estimates computed by **glm** function in baseline R.

Finally, we conclude the presence APOE-4 allele ($i = 1$) is associated with increased probability of transitioning to dementia (bad outcome) relative to transitioning to a good outcome ($\beta_1 > 0$). The contribution of APOE-4 to the model is found to be statistically significant at the $\alpha = .05$ level for the following transitions: from intact cognition to maintaining intact cognition, from MCI to intact cognition, and from GI to MCI. For all of these scenarios, the presence of the APOE-4 allele is associated with an increase in the relative odds of transitioning to dementia.

CHAPTER 5:
Conclusions

This chapter summarizes the sampling methodology described in Chapter 3 and its application to the nun cognitive observational dataset presented in Chapter 4. We discuss some benefits and shortcomings of the hybrid sampling approach for approximate inference. Finally, ideas for expansion and future work are suggested.

5.1 Summary

We begin with the problem of performing a goodness-of-fit test (to compare two models of the data) on a *sparse*, multidimensional contingency table. Due to the low cell counts, there is no guarantee on the validity of asymptotic inference. The multidimensionality of the contingency table also in general precludes exact enumeration of all contingency tables satisfying the sufficient statistic. Our solution is to conduct an approximate hypothesis test by sampling a large number of tables from the approximate distribution.

MCMC with the Metropolis-Hastings acceptance criteria provides a method of sampling contingency tables from the conditional state space such that the distribution of tables converges to the true distribution under H_0 by the law of large numbers. This method is only valid if a Markov chain exists that connects all feasible contingency tables. Without a Markov basis, there is no guarantee that contingency tables in the conditional state space are connected via an *ergodic* Markov chain. To overcome this obstacle, we rely on SIS to sample tables independently from the conditional state space under H_0. The distribution of SIS tables does not, however, yield an accurate hypergeometric distribution of contingency tables.

Algorithm 3.9.1 in Section 3.9 combines SIS and MCMC into a hybrid algorithm that generates an approximate distribution of contingency tables and test statistics for sparse, multidimensional contingency tables under H_0. First, many independent starting tables are sampled from the conditional state space via SIS. An MCMC chain is then initiated at each starting point. Each chain of test statistics has the beginning discarded (*burn-in*) and has the remaining test statistics thinned (*thinning*). The surviving test statistics from each MCMC chain are combined to form an approximate distribution of test statistics. Our assumption is that by sampling many different SIS starting points from the conditional state space, we are able to representatively sample from all regions of the conditional state space even though all contingency tables may not be connected via MCMC. By running MCMC for a sufficient number of iterations, we assume the sampled distribution will converge to the true distribution of test statistics.

In Chapter 4, we apply this sampling approach to a dataset of nuns experiencing different levels of cognitive impairment. The previous and current cognitive state along with the presence of APOE-4 allele, highest education level achieved, and age were recorded for each nun. Our goodness-of-fit hypothesis tests whether adding a third factor (H_1) statistically reduced the model's residual deviance compared to a simpler model with only the other two factors (H_0). Generating an approximate distribution through a hybrid sampler, we conclude the presence of APOE-4 allele ($i = 1$) is associated with higher rates of transition to dementia relative to transition rates to a better cognitive state. We also find that nuns with higher education levels ($i = 2$) were less likely to transition from intact cognition ($s = 1$) to dementia relative to transitioning from intact cognition to GI ($v = 3$), MCI ($v = 2$), or maintaining intact cognition ($v = 1$). Lastly, age ($i = 3$) is found to be significant in a model of the relative transition rates to dementia where previous cognitive status is MCI ($s = 2$) and current status is intact cognition ($v = 1$) or MCI ($v = 2$). Lower age is associated with increased likelihood of transitioning from MCI to intact cognition or maintaining MCI relative to transitioning to dementia. Older nuns are more likely to transition from MCI to dementia compared to younger nuns. Using three different methods (uniform, hypergeometric, and triangle) of sampling individual cells (X_j^*) within SIS, we observe the triangle distribution method results in the fewest rejections (*holes*).

5.2 Benefits and Shortcomings

The most significant benefit of the hybrid sampling approach is that no assumptions are made regarding the distribution of test statistics. Specifically, the test statistics are not assumed to have an asymptotic χ^2 distribution. This makes the approximate distribution method generally more conservative than asymptotic inference. Comparing two models via a goodness-of-fit test also circumvents the assumptions of t-tests on the model maximum likelihood estimates: homoskedastic, normally distributed, and independent residuals.

By employing SIS sampling to start various MCMC chains, we avoid the requirement of the conditional state space to be entirely accessible via a single Markov chain. Additionally, the hybrid sampling approach does not necessitate possession of Markov basis, which many times is unfeasible to compute. Enumerating all contingency tables satisfying the sufficient statistic is also not necessary for the hybrid approach.

There are also several drawbacks to approximate inference through a hybrid sampling approach. The SIS sample size must be large enough to ensure the entire conditional state space is accessible via the subsequent Markov chains. We currently have no way of knowing if this condition is definitively true. MCMC sample size must be large enough to not only establish independence from the starting point of the Markov chain but also to converge to the stationary distribution. Accurate representation through an approximate distribution requires a sample size of thousands if not hundreds of thousands.

A large sample size requires significant amounts of computing power and time. Even with the `cplex` solver, IP problems can be slow to solve, compounding the speed drawback. Each SIS contingency table requires solving $\ell \times i \times j \times k \times 2$ (min/max) IP problems. In the case of the nun data, that amounts to 96 IP solutions for each SIS contingency table, not including the discarded IP solutions resulting from holes. The existence of holes when conducting SIS only adds to the computation time. It is difficult to predict which systems will result in increased frequency of holes, although our analysis shows that holes will exist regardless of the sampling distribution assumption (see Table 4.4).

In the implementation on a personal computer (`Intel(R) Core(TM) i7-6500U CPU @ 2.50GHz and 16GB RAM`), the generation of 440,000 test statistics to conduct a single hypothesis test takes approximately 50 minutes. Extending our algorithm to 27 hypothesis tests with 3 different distribution assumption trials each, even for a relatively small 2×2×3×4 contingency table, requires around 72 hours to generate all 81 approximate distributions. Based on the congruence between the approximated and asymptotic distribution of test statistics for the nun cognitive dataset hypothesis tests, the additional computational time required for the hybrid sampler was not justified.

5.3 Future Work

The sampling scheme presented in Chapter 3 can be generalized to an arbitrary number of dimensions. The key to extending to higher dimensional contingency tables is constructing an algorithm to generate random $\begin{pmatrix} +1 & -1 \\ -1 & +1 \end{pmatrix}$ *moves* across the multiple dimensions that preserve the sufficient statistic (**b**). A perfect algorithm would be able to generate every possible move that preserves the sufficient statistic and each move would be equally likely, but these conditions are difficult to prove. We recommend implementing the SIS and MCMC hybrid sampling method on a higher dimensional dataset to verify its suitability. It is also interesting to catalogue scenarios where the hybrid sampling scheme diverges from the asymptotic approximation. The similarity between estimated p-values for sparse contingency tables from hybrid sampling and those from asymptotic approximation is a surprising result of this research.

We attempted to eliminate the possibility of *holes* in the linear system via column based reduction (CBR) basis transformation through the Lenstra-Lenstra-Lovász (LLL) (Lenstra et al. 1982) and block Korkin-Zolotarev (BKZ) (Schnorr 1987) lattice basis reduction algorithms. We utilized the `LLL` function of the `m2r` R package from David Kahle which connects the algebraic geometry algorithms of `Macaulay2` (Grayson and Stillman 2017) to R (Kahle et al. 2017a). Unfortunately, our $A \cdot \mathbf{x} = \mathbf{b}$ system is too large for the current version of `Macaulay2` to solve, although we successfully achieved hole reduction by applying CBR to smaller, toy problems. As implementations of the LLL algorithm improve, the reduction of holes through forward and backward CBR could be investigated.

We believe the sampling scheme presented provides a viable means of conducting approximate inference for hypothesis tests on sparse, multidimensional tables. The algorithm could also be employed a limited number of times to check the validity of the asymptotic approximation assumption for a given dataset. The most obvious application is for medical data, military or otherwise, that has multiple dimensions of patient factors and a limited sample size. However, this methodology should function for *any* military contingency table in which these conditions are met, such as costly weapons testing with multiple independent factors, simulation analysis when the number of simulations is limited due to time or cost, or wargaming results under different starting conditions.

APPENDIX A:
Nun Cognitive Observational Dataset

Table A.1 shows the complete nun cognitive observational dataset used for our analysis.

Table A.1. Nun Cognitive Observational Dataset.

Prior State	Current State	APOE	Educ	Age	Observations
1	1	1	1	1	2
1	1	1	1	2	5
1	1	1	1	3	3
1	1	1	2	1	44
1	1	1	2	2	53
1	1	1	2	3	49
1	1	1	2	4	28
1	1	1	3	1	123
1	1	1	3	2	105
1	1	1	3	3	80
1	1	1	3	4	44
1	1	2	2	1	3
1	1	2	2	2	4
1	1	2	2	3	1
1	1	2	3	1	15
1	1	2	3	2	15
1	1	2	3	3	12
1	1	2	3	4	7
1	2	1	1	1	2
1	2	1	1	2	3
1	2	1	1	3	2
1	2	1	1	4	1
1	2	1	2	1	29
1	2	1	2	2	11
1	2	1	2	3	13

Continuation of Table A.1					
Prior State	Current State	APOE	Educ	Age	Observations
1	2	1	2	4	14
1	2	1	3	1	32
1	2	1	3	2	24
1	2	1	3	3	26
1	2	1	3	4	18
1	2	2	2	1	5
1	2	2	2	2	3
1	2	2	2	3	3
1	2	2	2	4	1
1	2	2	3	1	5
1	2	2	3	2	3
1	2	2	3	3	2
1	3	1	2	1	3
1	3	1	2	2	2
1	3	1	2	3	6
1	3	1	2	4	6
1	3	1	3	1	6
1	3	1	3	2	11
1	3	1	3	3	9
1	3	1	3	4	5
1	3	2	2	2	2
1	3	2	3	1	1
1	3	2	3	3	2
1	3	2	3	4	1
1	4	1	1	2	2
1	4	1	2	3	1
1	4	2	2	2	1
1	4	2	3	3	1
1	5	1	1	2	1
1	5	1	1	4	1
1	5	1	2	1	2

Continuation of Table A.1					
Prior State	Current State	APOE	Educ	Age	Observations
1	5	1	2	3	4
1	5	1	2	4	5
1	5	1	3	1	5
1	5	1	3	2	7
1	5	1	3	3	11
1	5	1	3	4	6
1	5	2	2	4	1
1	5	2	3	1	1
1	5	2	3	2	2
1	5	2	3	3	1
1	5	2	3	4	1
2	1	1	1	1	4
2	1	1	1	2	4
2	1	1	1	4	1
2	1	1	2	1	24
2	1	1	2	2	21
2	1	1	2	3	8
2	1	1	2	4	7
2	1	1	3	1	37
2	1	1	3	2	21
2	1	1	3	3	21
2	1	1	3	4	14
2	1	2	2	1	4
2	1	2	2	2	1
2	1	2	2	3	2
2	1	2	2	4	1
2	1	2	3	1	4
2	1	2	3	2	2
2	1	2	3	3	1
2	2	1	1	1	21
2	2	1	1	2	11

Continuation of Table A.1					
Prior State	Current State	APOE	Educ	Age	Observations
2	2	1	1	3	18
2	2	1	1	4	24
2	2	1	2	1	44
2	2	1	2	2	66
2	2	1	2	3	61
2	2	1	2	4	71
2	2	1	3	1	76
2	2	1	3	2	68
2	2	1	3	3	60
2	2	1	3	4	50
2	2	2	1	1	2
2	2	2	1	2	3
2	2	2	1	3	3
2	2	2	2	1	17
2	2	2	2	2	12
2	2	2	2	3	10
2	2	2	2	4	6
2	2	2	3	1	26
2	2	2	3	2	22
2	2	2	3	3	17
2	2	2	3	4	9
2	3	1	1	1	5
2	3	1	1	2	1
2	3	1	1	3	3
2	3	1	1	4	4
2	3	1	2	1	2
2	3	1	2	2	5
2	3	1	2	3	17
2	3	1	2	4	24
2	3	1	3	1	4
2	3	1	3	2	6

Continuation of Table A.1					
Prior State	Current State	APOE	Educ	Age	Observations
2	3	1	3	3	16
2	3	1	3	4	15
2	3	2	1	1	1
2	3	2	1	2	1
2	3	2	2	1	4
2	3	2	2	2	10
2	3	2	2	3	3
2	3	2	2	4	3
2	3	2	3	1	2
2	3	2	3	2	3
2	3	2	3	3	4
2	3	2	3	4	3
2	4	1	1	1	1
2	4	1	1	2	2
2	4	1	1	3	3
2	4	1	1	4	3
2	4	1	2	1	3
2	4	1	2	2	6
2	4	1	2	3	4
2	4	1	2	4	17
2	4	1	3	1	6
2	4	1	3	2	6
2	4	1	3	3	7
2	4	1	3	4	9
2	4	2	2	2	1
2	4	2	2	3	1
2	4	2	3	1	3
2	4	2	3	2	6
2	4	2	3	3	1
2	4	2	3	4	3
2	5	1	1	3	1

Continuation of Table A.1					
Prior State	Current State	APOE	Educ	Age	Observations
2	5	1	1	4	7
2	5	1	2	1	5
2	5	1	2	2	4
2	5	1	2	3	9
2	5	1	2	4	16
2	5	1	3	1	2
2	5	1	3	2	8
2	5	1	3	3	4
2	5	1	3	4	11
2	5	2	1	1	1
2	5	2	1	4	1
2	5	2	2	3	2
2	5	2	2	4	3
2	5	2	3	1	1
2	5	2	3	2	2
2	5	2	3	3	4
2	5	2	3	4	2
3	1	1	2	1	1
3	1	1	2	2	1
3	1	1	2	3	3
3	1	1	2	4	1
3	1	1	3	1	1
3	1	1	3	2	3
3	1	1	3	3	2
3	1	1	3	4	1
3	1	2	2	2	1
3	1	2	3	3	1
3	1	2	3	4	1
3	2	1	1	1	1
3	2	1	1	2	2
3	2	1	1	3	1

Continuation of Table A.1					
Prior State	Current State	APOE	Educ	Age	Observations
3	2	1	1	4	2
3	2	1	2	1	1
3	2	1	2	2	3
3	2	1	2	3	9
3	2	1	2	4	5
3	2	1	3	1	1
3	2	1	3	2	2
3	2	1	3	3	2
3	2	1	3	4	5
3	2	2	1	1	1
3	2	2	2	1	1
3	2	2	2	2	1
3	2	2	3	2	1
3	2	2	3	4	1
3	3	1	1	1	3
3	3	1	1	2	4
3	3	1	1	3	6
3	3	1	1	4	15
3	3	1	2	1	3
3	3	1	2	2	3
3	3	1	2	3	8
3	3	1	2	4	39
3	3	1	3	1	5
3	3	1	3	2	16
3	3	1	3	3	17
3	3	1	3	4	19
3	3	2	1	2	1
3	3	2	1	3	2
3	3	2	1	4	2
3	3	2	2	1	7
3	3	2	2	2	7

Continuation of Table A.1

Prior State	Current State	APOE	Educ	Age	Observations
3	3	2	2	3	18
3	3	2	2	4	2
3	3	2	3	2	1
3	3	2	3	3	3
3	3	2	3	4	3
3	4	1	1	1	2
3	4	1	1	2	3
3	4	1	1	4	2
3	4	1	2	1	2
3	4	1	2	2	2
3	4	1	2	3	5
3	4	1	2	4	18
3	4	1	3	1	2
3	4	1	3	3	4
3	4	1	3	4	10
3	4	2	1	4	2
3	4	2	2	1	1
3	4	2	2	2	2
3	4	2	2	3	4
3	4	2	2	4	5
3	4	2	3	1	5
3	4	2	3	2	1
3	4	2	3	3	3
3	4	2	3	4	2
3	5	1	1	2	1
3	5	1	1	4	6
3	5	1	2	1	1
3	5	1	2	2	2
3	5	1	2	3	5
3	5	1	2	4	21
3	5	1	3	1	4

Continuation of Table A.1					
Prior State	Current State	APOE	Educ	Age	Observations
3	5	1	3	2	10
3	5	1	3	3	13
3	5	1	3	4	12
3	5	2	2	1	2
3	5	2	2	2	5
3	5	2	2	3	5
3	5	2	2	4	2
3	5	2	3	2	1
3	5	2	3	3	1
3	5	2	3	4	3
End of Table					

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B:
Test Statistic Distributions

Figures B.1 to B.27 show histograms for the sampled test statistic distribution for each (s, v, i) combination and sampling method.

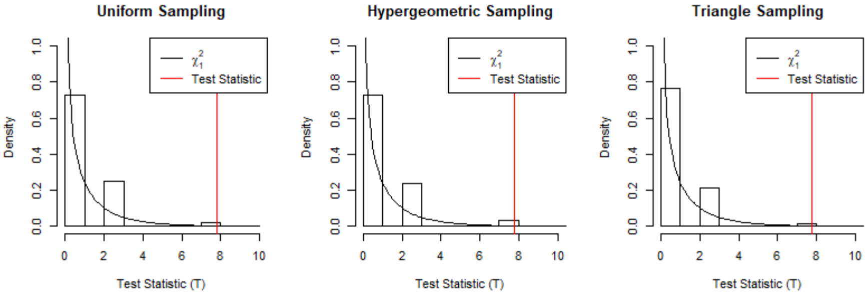


Figure B.1. Hybrid Sampling Test Statistic Distribution: $s = 1, v = 1, i = 1$.

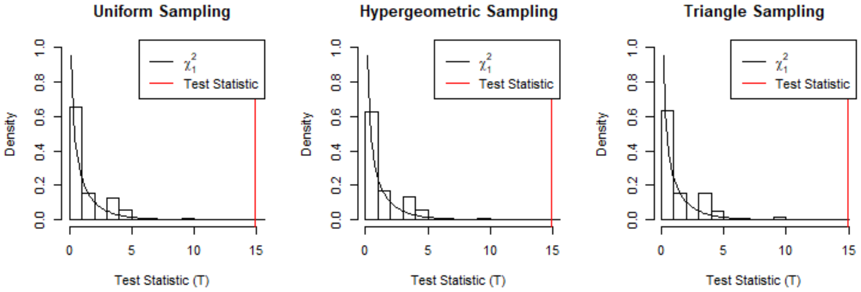


Figure B.2. Hybrid Sampling Test Statistic Distribution: $s = 1, v = 1, i = 2$.

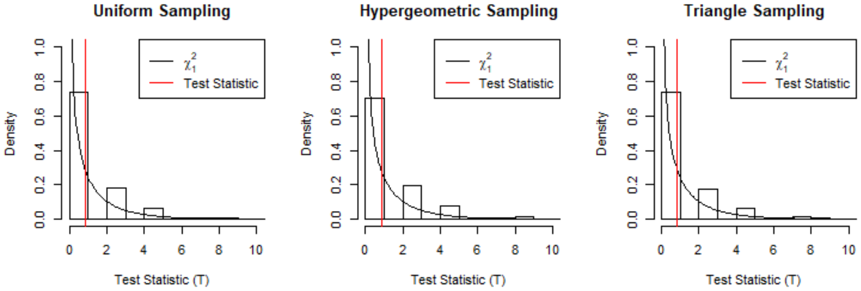


Figure B.3. Hybrid Sampling Test Statistic Distribution: $s = 1, v = 1, i = 3$.

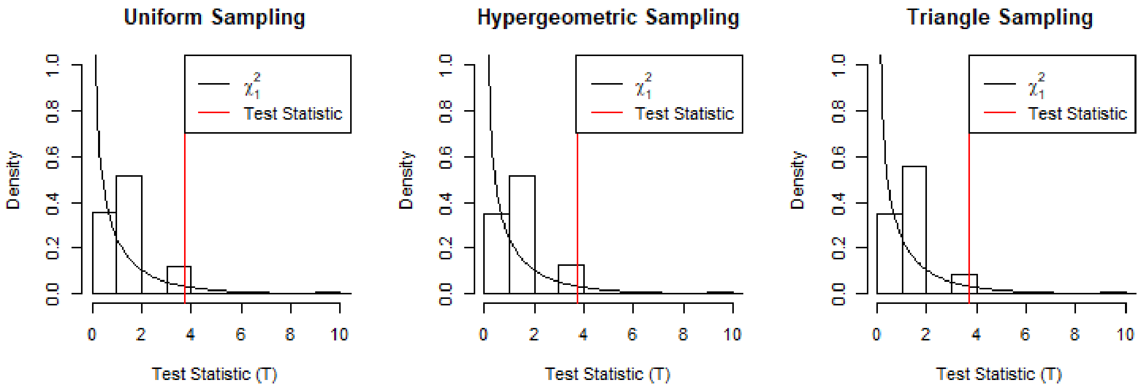


Figure B.4. Hybrid Sampling Test Statistic Distribution: $s = 1, v = 2, i = 1$.

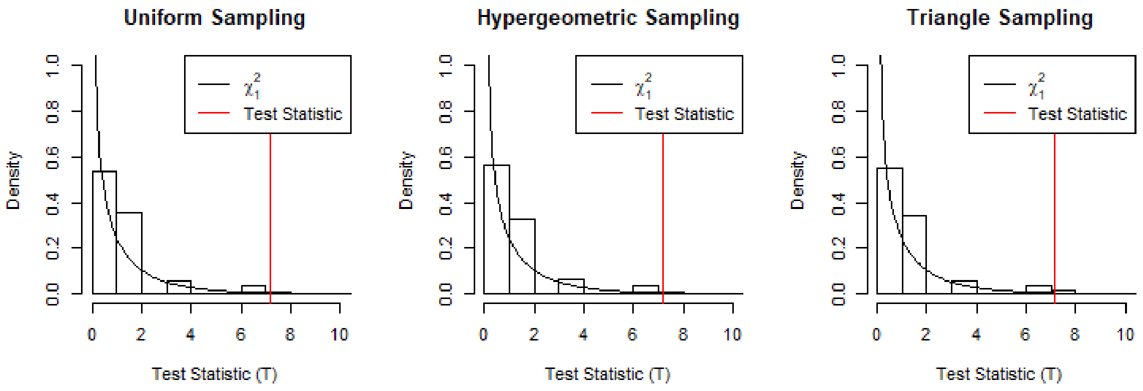


Figure B.5. Hybrid Sampling Test Statistic Distribution: $s = 1, v = 2, i = 2$.

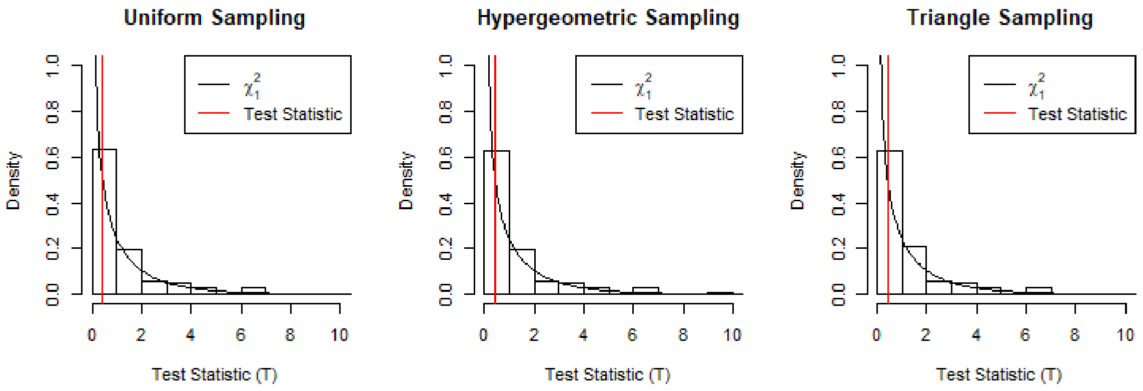


Figure B.6. Hybrid Sampling Test Statistic Distribution: $s = 1, v = 2, i = 3$.

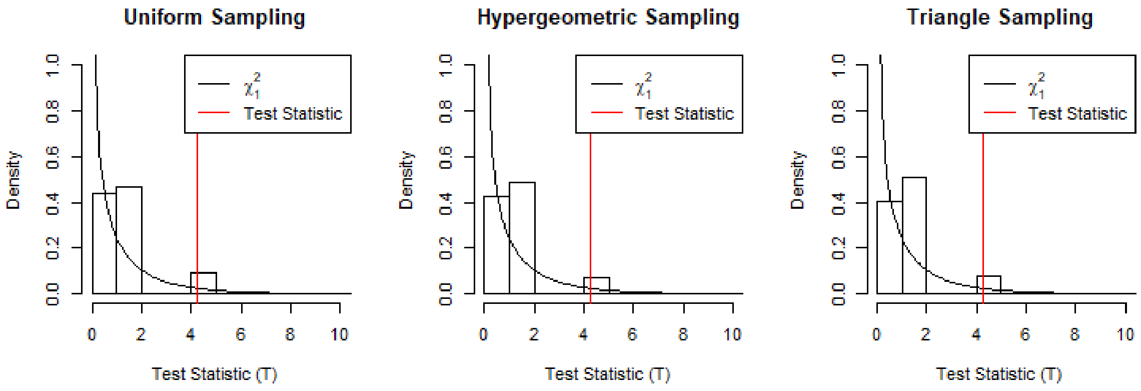


Figure B.7. Hybrid Sampling Test Statistic Distribution: $s = 1, v = 3, i = 1$.

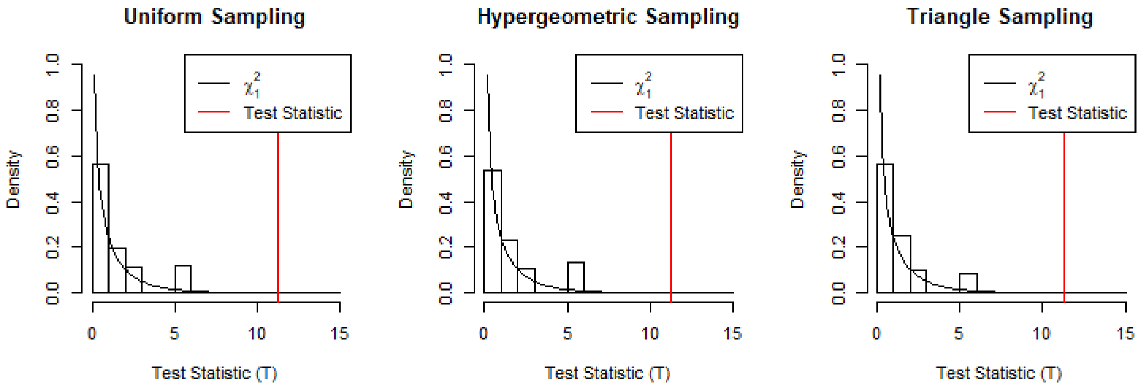


Figure B.8. Hybrid Sampling Test Statistic Distribution: $s = 1, v = 3, i = 2$.

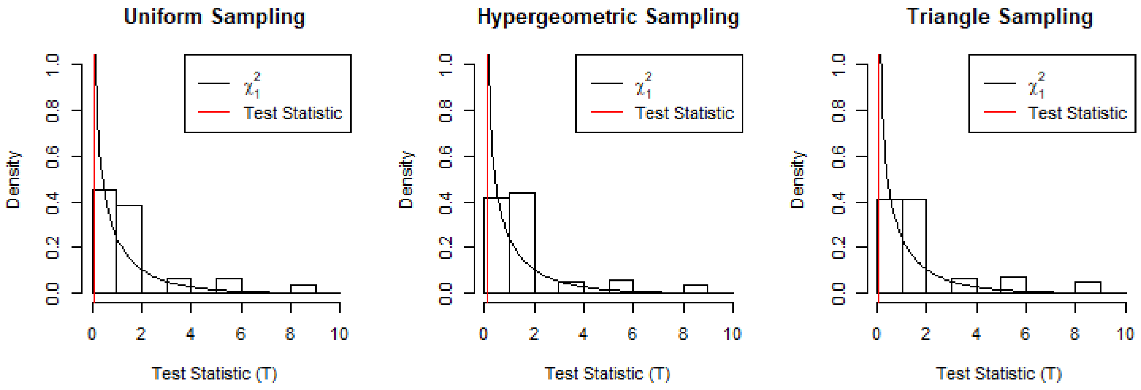


Figure B.9. Hybrid Sampling Test Statistic Distribution: $s = 1, v = 3, i = 3$.

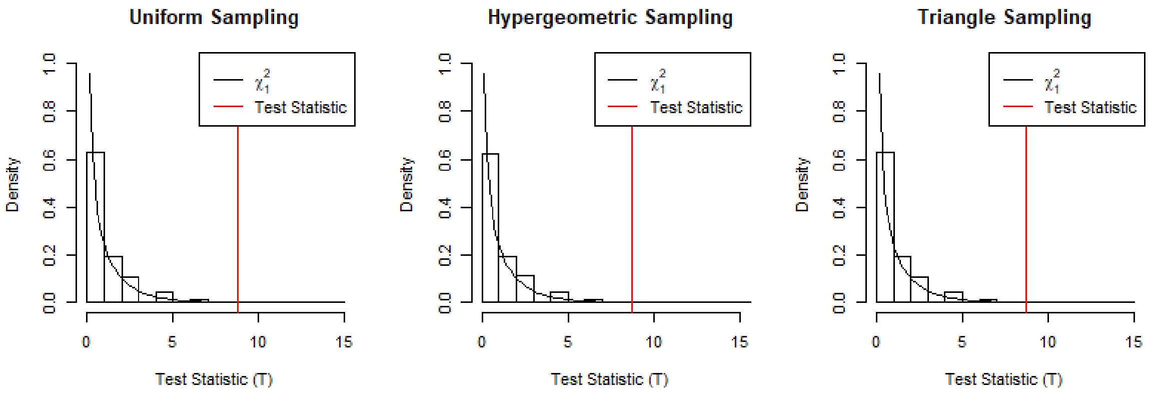


Figure B.10. Hybrid Sampling Test Statistic Distribution: $s = 2, v = 1, i = 1$.

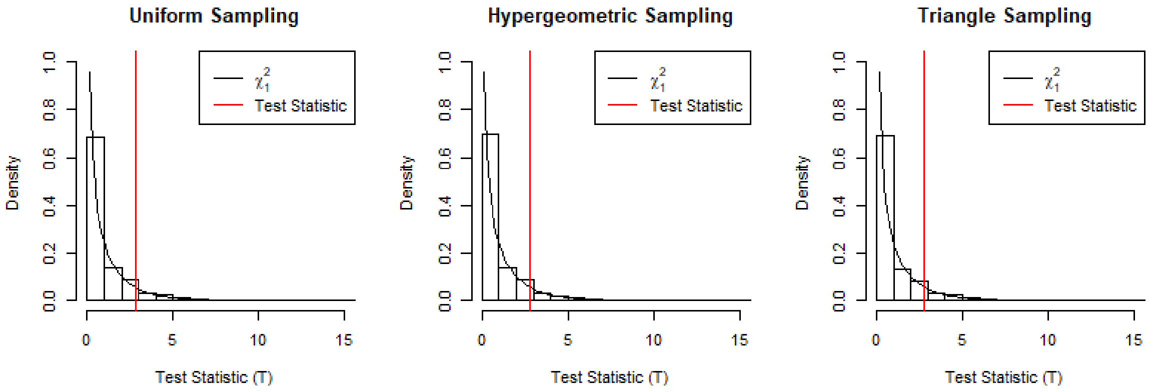


Figure B.11. Hybrid Sampling Test Statistic Distribution: $s = 2, v = 1, i = 2$.

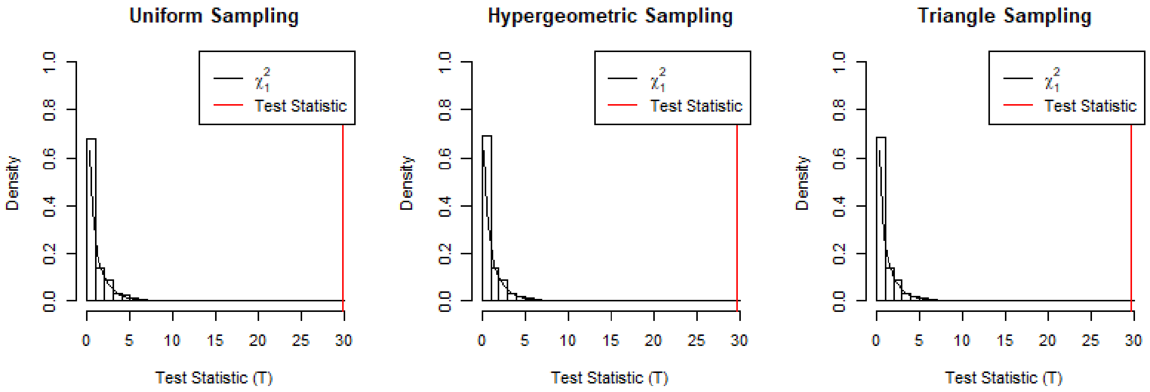


Figure B.12. Hybrid Sampling Test Statistic Distribution: $s = 2, v = 1, i = 3$.

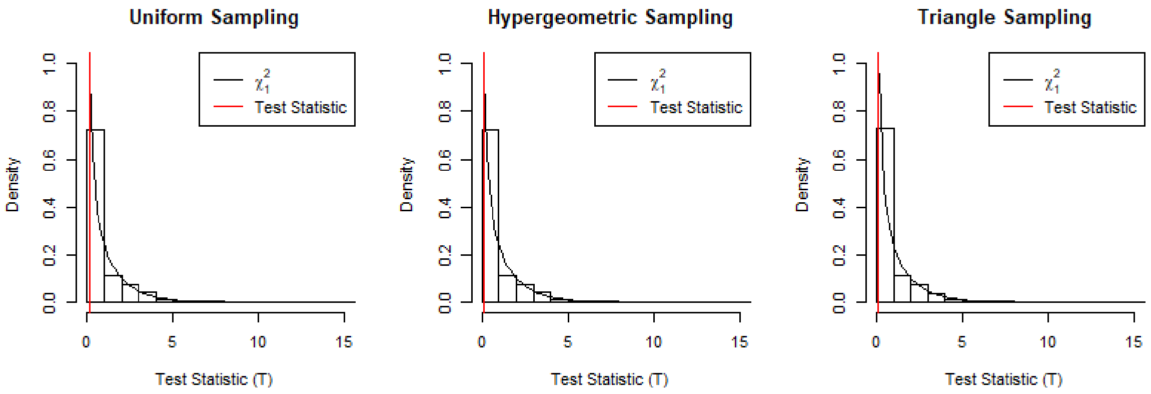


Figure B.13. Hybrid Sampling Test Statistic Distribution: $s = 2, v = 2, i = 1$.

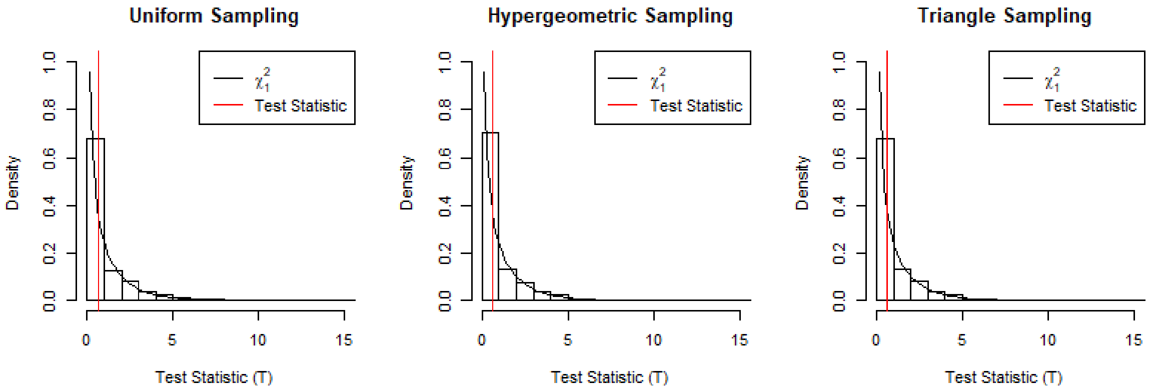


Figure B.14. Hybrid Sampling Test Statistic Distribution: $s = 2, v = 2, i = 2$.

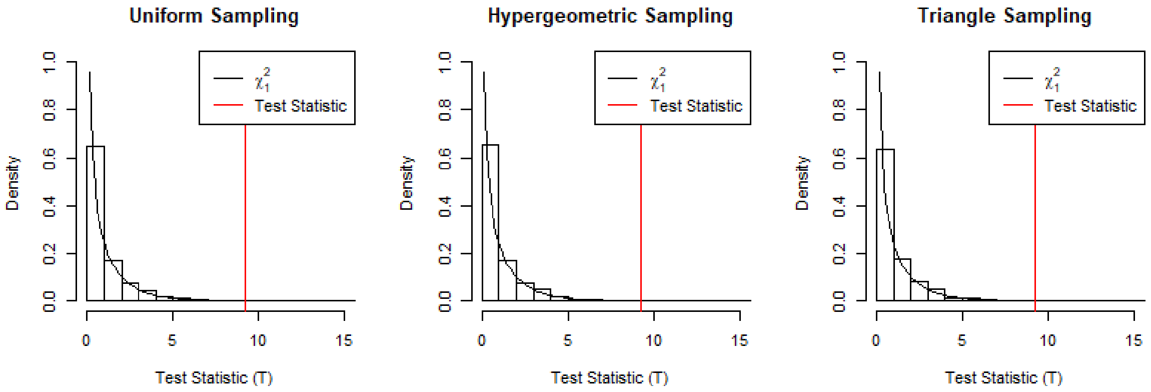


Figure B.15. Hybrid Sampling Test Statistic Distribution: $s = 2, v = 2, i = 3$.

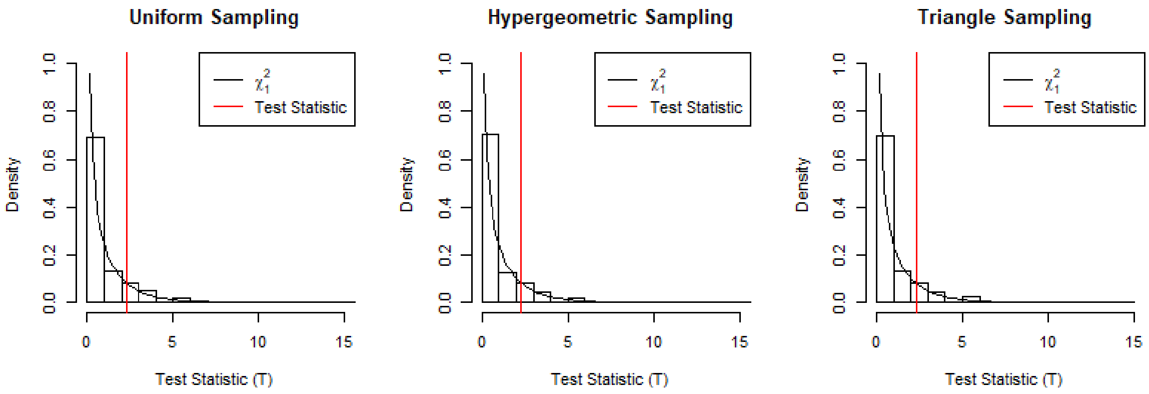


Figure B.16. Hybrid Sampling Test Statistic Distribution: $s = 2, v = 3, i = 1$.

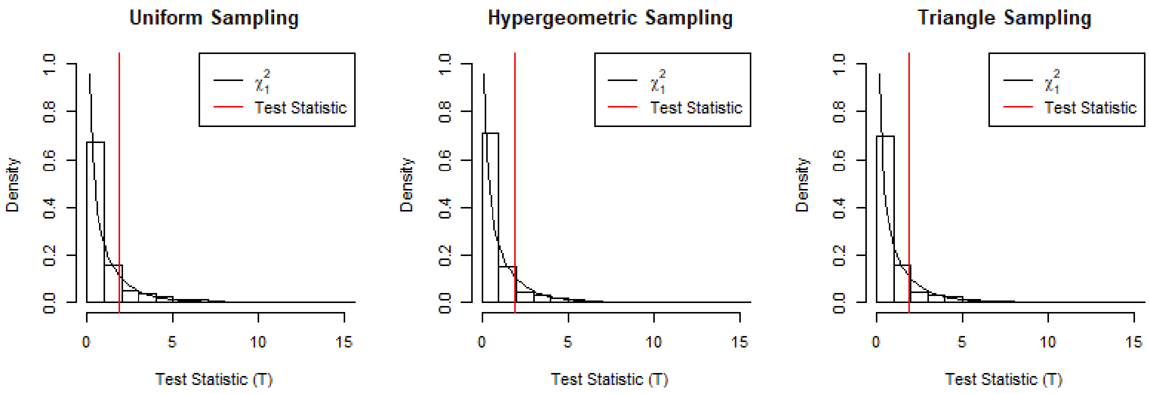


Figure B.17. Hybrid Sampling Test Statistic Distribution: $s = 2, v = 3, i = 2$.

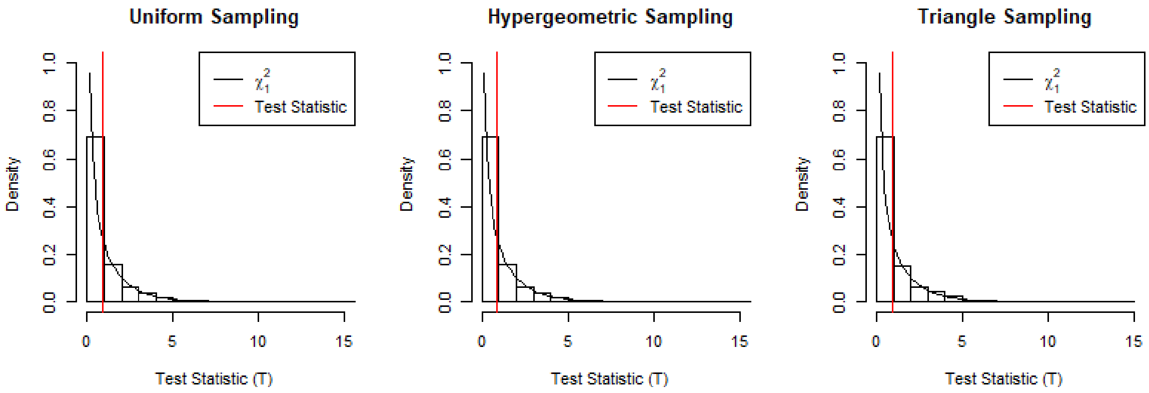


Figure B.18. Hybrid Sampling Test Statistic Distribution: $s = 2, v = 3, i = 3$.

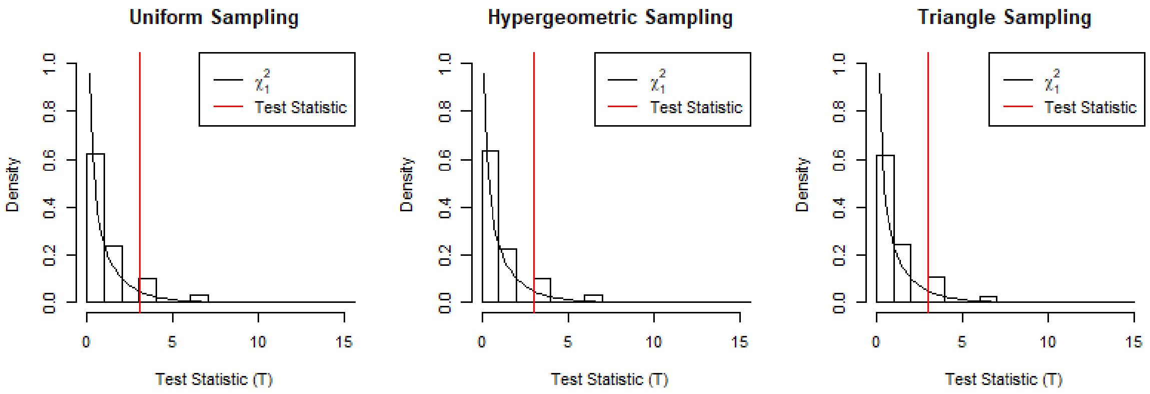


Figure B.19. Hybrid Sampling Test Statistic Distribution: $s = 3, v = 1, i = 1$.

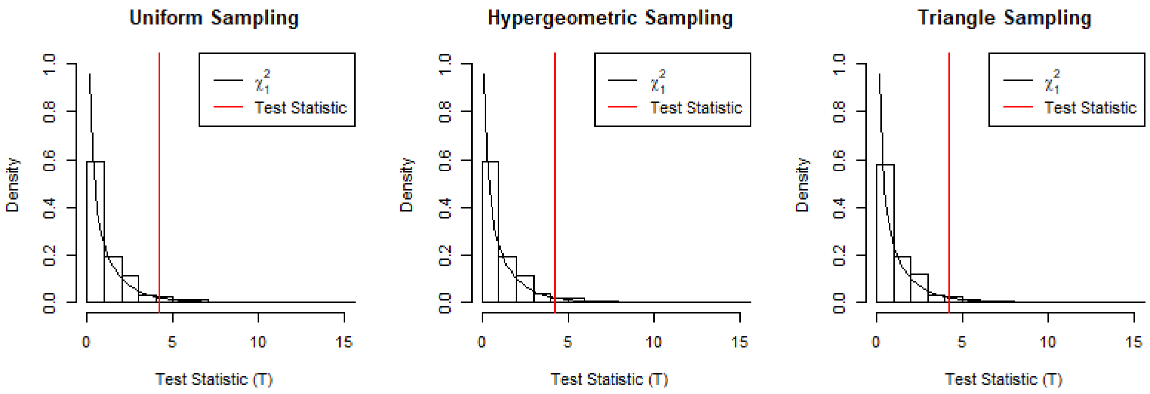


Figure B.20. Hybrid Sampling Test Statistic Distribution: $s = 3, v = 1, i = 2$.

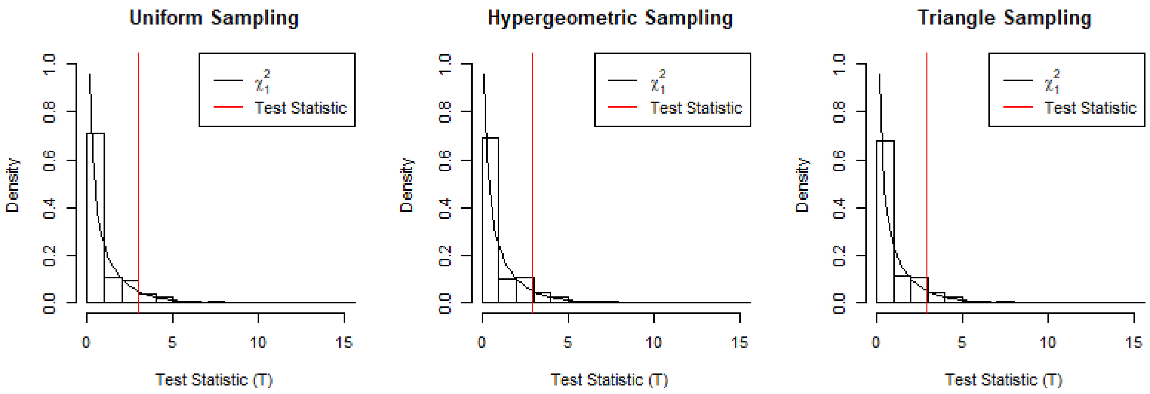


Figure B.21. Hybrid Sampling Test Statistic Distribution: $s = 3, v = 1, i = 3$.

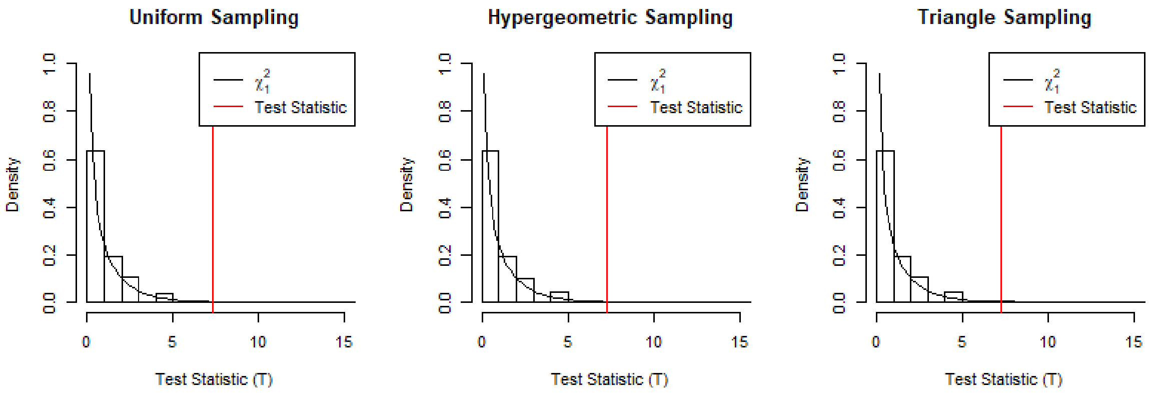


Figure B.22. Hybrid Sampling Test Statistic Distribution: $s = 3, v = 2, i = 1$.

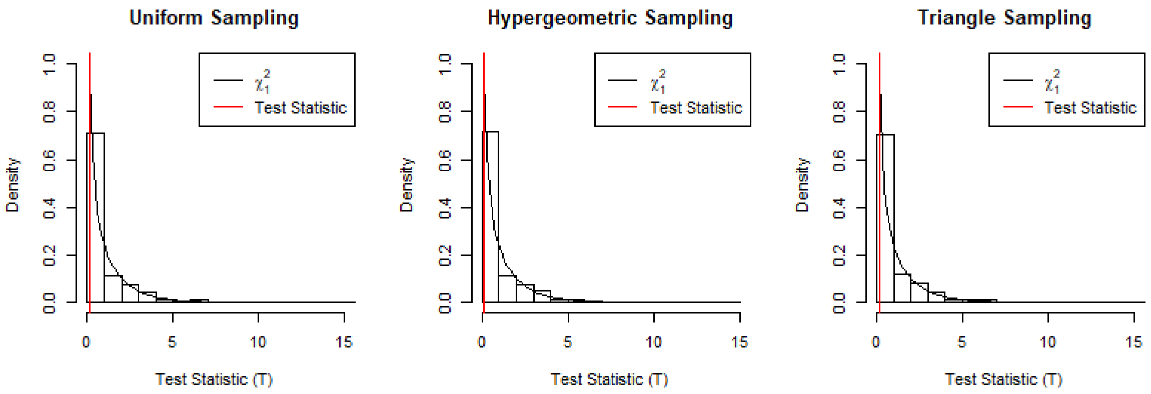


Figure B.23. Hybrid Sampling Test Statistic Distribution: $s = 3, v = 2, i = 2$.

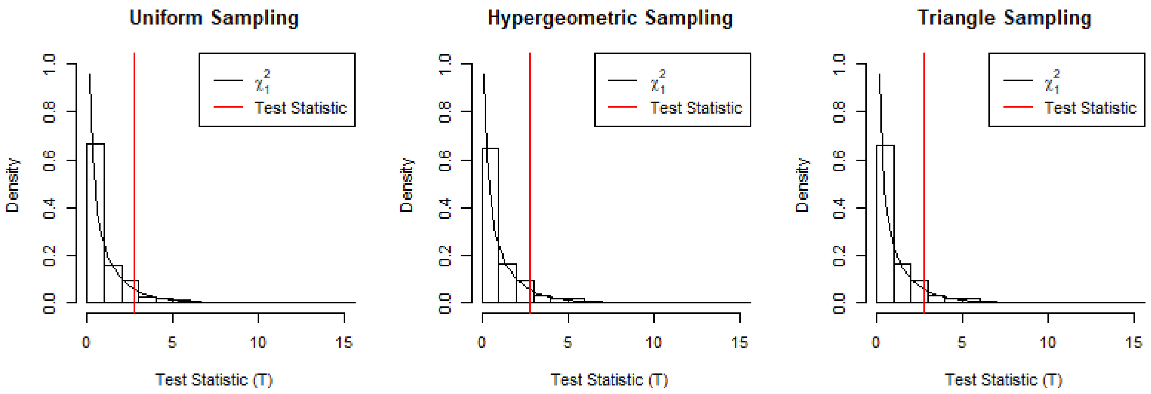


Figure B.24. Hybrid Sampling Test Statistic Distribution: $s = 3, v = 2, i = 3$.

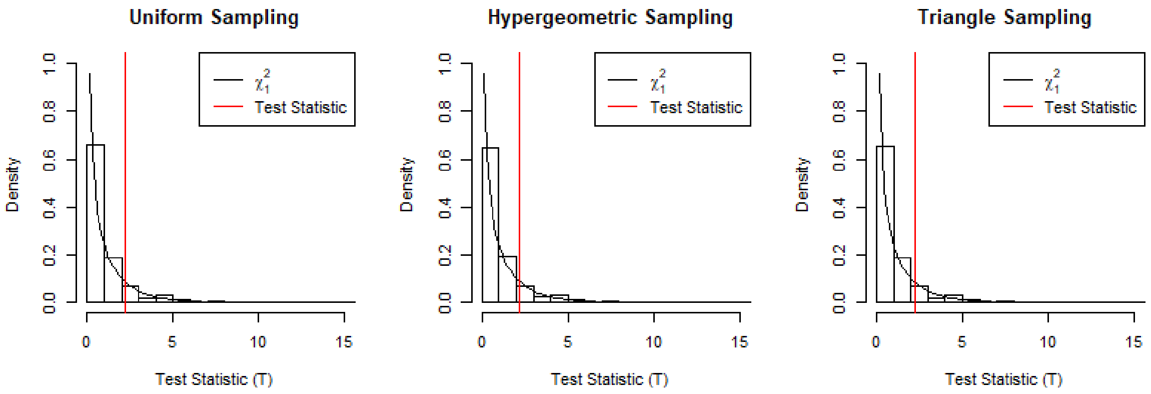


Figure B.25. Hybrid Sampling Test Statistic Distribution: $s = 3, v = 3, i = 1$.

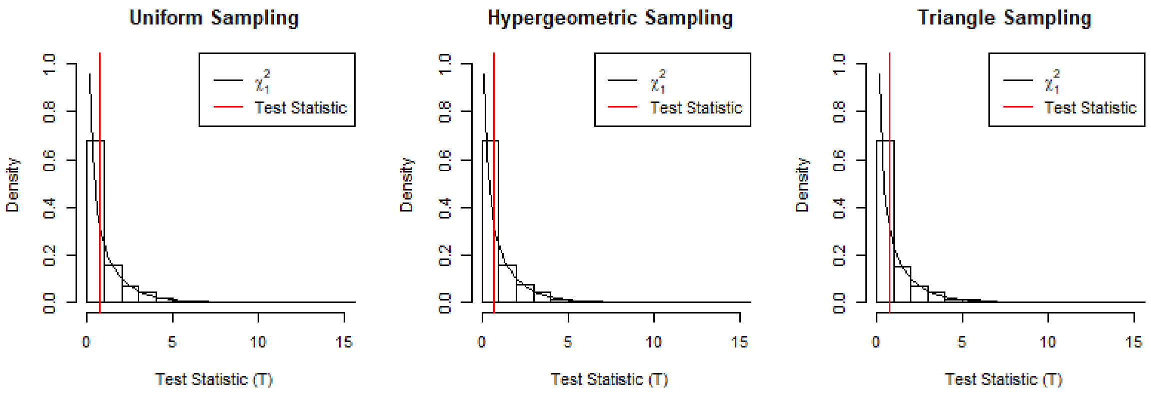


Figure B.26. Hybrid Sampling Test Statistic Distribution: $s = 3, v = 3, i = 2$.

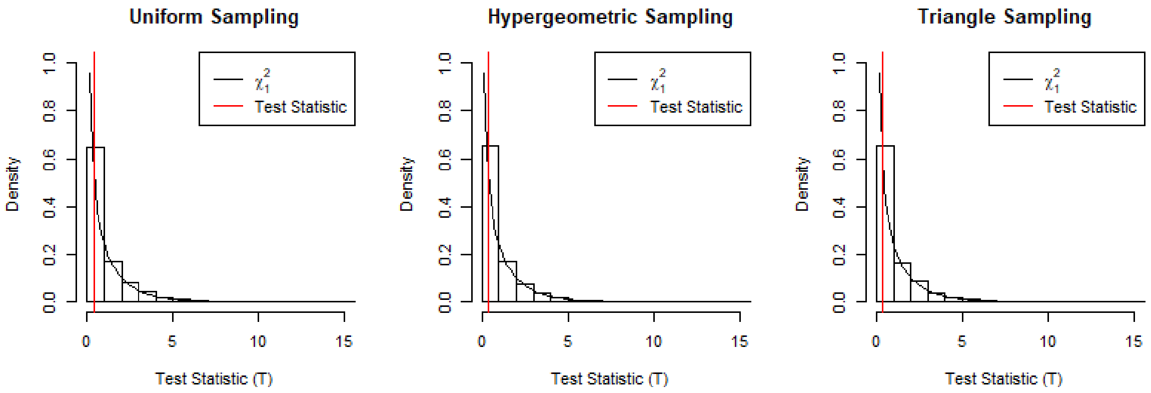


Figure B.27. Hybrid Sampling Test Statistic Distribution: $s = 3, v = 3, i = 3$.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C: R Code

```
res0 <- glm(L ~ Y1+Y2, family = binomial(link="logit"), data=tab)
res1 <- glm(L ~ Y1+Y2+Y3, family = binomial(link="logit"), data=tab)

#Load the relevant libraries
library(Rcplex)      #We use cplex as the solver for the MIP
library(triangle)    #This package allows for each random number from triangle distribution


#Bring the data into R
#Set the appropriate working directory
setwd('C:/Users/Pat/Desktop/Thesis/thesis_data')

#Read in the csv file
nun <- read.csv(file="Goeman2.csv", header=TRUE, sep=",")
dim(nun) #there should be 2480 observations

#Reorganize the observations into a contingency table
T <- xtabs(~apoe+EDCAT+age2+priorstate+currentstate, data=nun)
sum(T)  #total cell counts should again be equal to 2480

###Building a contingency table with relevant cell counts
###For this hypothesis test, we look at transitions to dementia (state 4)
build.array <- function(s,v,T) {
  data1 <- T[,,,s,4]
  data2 <- T[,,,s,v]

  I <- dim(data1)[1]
  J <- dim(data1)[2]
  K <- dim(data1)[3]

  x <- array(NA, dim=c(2,I,J,K))
  x[1,,,] <- data1
  x[2,,,] <- data2

  return(x)
}



##############################################################################
#Build the Lawrence Lifting matrix A, and solve for sufficient statistic b. Ax=b.
#Lawrence lifting matrix (A) gives the relation between the observation vector and the sufficient statistic

#Function create.configuration produces the levels for each factor
create.configuration <- function(n){
  #this function creates the configuration (sufficient stat) matrix for a given J (K)
  top.row <- rep(1,n)
  bottom.row <- seq(1,n)
  A <- rbind(top.row,bottom.row)
  return(A)
}


lawrence.lift <- function(...){
  #input to this function is arbitrary number of configuration matrices in a 2xJ(K) format
  #output is the Lawrence lifting matrix with Segre product in upper left, zeros in upper right,
  # and two identical Identity(I) matrices on bottom
  #how many different configurations (variates) are there?
  configurations <- list(...)
  #print(configurations)
  n <- length(configurations)

  #calculate number of rows and columns in Segre product
  #by setup of logistic regression, each variate has "0" (failure) row and "1" (success) row
  num.rows <- n + 1
  cols <- c()
  for (i in 1:n){
    cols[i] <- dim(configurations[[i]])[2]
    #print(cols)
  }
  num.cols <- prod(cols)
  #print(num.cols)
  segre <- matrix(NA,num.rows,num.cols)

  #create the Segre product matrix
  #fill in the row of 1’s
  segre[1,] <- 1

  #fill in the "iteration" row
  for (j in 2:num.rows) {
    repeats <- num.cols/prod(cols[1:j-1])
    segre[j,] <- rep(unique(configurations[[j-1]][2,]),each=repeats)
  }

  #create the Lawrence lifting matrix
  zeros <- matrix(0,num.rows,num.cols)
  eye <- diag(num.cols)          #identity matrix of proper dimension
  eye <- cbind(eye,eye)
  lawrence <- cbind(segre,zeros)
  lawrence <- rbind(lawrence, eye)

  return(lawrence)
}


sufficient.statistic <- function(A,x,extra.var){
  A <- A[-(5-extra.var),]
  b <- A %*% aperm(x,c(2,3,4,1))
  output <- list(A,b)
  return(output)
}
```

```
#################################################################################
###Hara san code modified by Yoshida san for conducting MCMC sampling
#  As written for the calculation of a "move," this code assumes:
#    H_0 : logit with 2 covariates
#    H_1 : logit with 3 covariates
#

#Function choose.cells determines which levels (1:d) will be permuted for each dimension
#2 or 4 levels could be permuted

choose.cells <- function(d){
# choose cells for (1, -1, -1, 1) or (-1, 1, 1, -1)
# input : number of levels
# output : cells for (1, -1, -1, 1) or (-1, 1, 1, -1)

        a <- sort(sample(1:d,2,replace=TRUE))

        if( a[1]==a[2] ){
                cell <- rep(a[1],4)
        }else{
                diff <- a[2] - a[1] - 1
                if(diff %% 2 == 0){
                        b <- diff %/% 2
                }else{
                        b <- diff %/% 2 + 1
                }
                c <- sample(0:b,1)
                cell <- c(a[1],a[1]+c,a[2]-c,a[2])
                }

return(cell)
}


#Function deg4move applies +1/-1 moves to the contingency table
#There will be 4 +1/-1 moves generated that preserve sufficient statistic

deg4move <- function(d){

        # generating deg 4 moves
        # input : dimension of a table
        # output : a deg 4 move

        repeat{
                z <- array(rep(0,48),dim=d) #depends on total number of cells

            #d[1] is the Response Dimension
                i2 <- choose.cells(d[2])
                i3 <- choose.cells(d[3])
#                i4 <- choose.cells(d[4])
#                i5 <- choose.cells(d[5])
            #because d[4] is not part of H0, it is not part of sufficient statistic
                i4 <- c(sample(d[4],1),sample(d[4],1),sample(d[4],1),sample(d[4],1))

                imat <- rbind(i2,i3,i4)

                z[1,i2[1],i3[1],i4[1]] <- 1
                z[1,i2[2],i3[2],i4[2]] <- -1
                z[1,i2[3],i3[3],i4[3]] <- z[1,i2[3],i3[3],i4[3]] -1
                z[1,i2[4],i3[4],i4[4]] <- 1

                z[2,i2[1],i3[1],i4[1]] <- -z[1,i2[1],i3[1],i4[1]]
                z[2,i2[2],i3[2],i4[2]] <- -z[1,i2[2],i3[2],i4[2]]
                z[2,i2[3],i3[3],i4[3]] <- -z[1,i2[3],i3[3],i4[3]]
                z[2,i2[4],i3[4],i4[4]] <- -z[1,i2[4],i3[4],i4[4]]

                deg <- sum(abs(z))
```

```
                 if(deg == 8){  #-1 should cancel with +1
                         break
                 }
         }

         if(runif(1) < 0.5){              #coin flip for +/-
                 return(list(z,imat))
         }else{
                 return(list(-z,imat))
        }

}


#Function "idata" simply transforms the contingency table cell counts into individual observations

idata <- function(x3, d3, extra.var){
# transformation from contingency table to individual data
# to use glm() for estimating logit model
# input : a table
# output : an individual data

        tab <- as.vector(NULL)

        for ( i1 in 1:d3[1] ){
                for ( i2 in 1:d3[2] ){
                        for ( i3 in 1:d3[3]){
                                for ( i4 in 1:d3[4] ){
                                    if ( (i1 == 1) && (x3[i1,i2,i3,i4] !=0 ) ) {
                                        for ( j in 1:x3[i1,i2,i3,i4] ){
                                            tab <- rbind(tab,c(1,i2,i3,i4))
                                        }
                                    }else if ( (i1 == 2) && (x3[i1,i2,i3,i4] !=0 ) ){
                                        for ( j in 1:x3[i1,i2,i3,i4] ){
                                            tab <- rbind(tab,c(0,i2,i3,i4))
                                        }
                                    }

                                }
                        }
                }
        }

        tab <- data.frame(tab)

      if (extra.var == 3)
        colnames(tab) = c('L','Y1','Y2','Y3')
      else if (extra.var == 2)
        colnames(tab) = c('L','Y1','Y3','Y2')
      else
        colnames(tab) = c('L','Y2','Y3','Y1')

      return(tab)
}


#Determine acceptance ratio via Metropolis-Hastings algorithm

accept.ratio <- function(w,z,x){
# computaion of acceptance ratio in MH procedure
# input : a move, nonzero cells in the move, current contingency table
# output : acceptance ratio

        if( z[1,w[1,1],w[2,1],w[3,1]]==1 ){

                num <- x[2,w[1,1],w[2,1],w[3,1]] * x[1,w[1,2],w[2,2],w[3,2]] *
                x[1,w[1,3],w[2,3],w[3,3]] * x[2,w[1,4],w[2,4],w[3,4]]
                den <- ( x[1,w[1,1],w[2,1],w[3,1]] + 1 ) *
                       ( x[2,w[1,2],w[2,2],w[3,2]] + 1 ) *
                       ( x[2,w[1,3],w[2,3],w[3,3]] + 1 ) *
                       ( x[1,w[1,4],w[2,4],w[3,4]] + 1 )

                return(min(1,num/den))

        }else{

                num <- x[1,w[1,1],w[2,1],w[3,1]] * x[2,w[1,2],w[2,2],w[3,2]] *
                x[2,w[1,3],w[2,3],w[3,3]] *  x[1,w[1,4],w[2,4],w[3,4]]

                den <- ( x[2,w[1,1],w[2,1],w[3,1]] + 1 ) *
                       ( x[1,w[1,2],w[2,2],w[3,2]] + 1 ) *
                       ( x[1,w[1,3],w[2,3],w[3,3]] + 1 ) *
                       ( x[2,w[1,4],w[2,4],w[3,4]] + 1 )

                return(min(1,num/den))
        }
}




#############################################################
mcmc.sampling <- function(x, sim, extra.var, a=A, B=b) {
# computaion of likelihood ratio (LR) test statistics from a single Markov chain
# input : a starting contingency table, number of simulations/moves to run,
#         the factor not part of H0
# output : a vector of LR test statistics which converges to true distribution under certain assumptions

# First, permute the matrix to maintain the sufficient statistic when performing an MCMC move
  if (extra.var == 3)
    x3 <- aperm(x,c(1,2,3,4))
  else if (extra.var == 2)
    x3 <- aperm(x,c(1,2,4,3))
  else
    x3 <- aperm(x,c(1,3,4,2))

  d3 <- dim(x3)

  tab <- idata(x3, d3, extra.var)
  names = colnames(tab)

  markov.chain <- rep(NA,sim)      #initialize a vector to store the test statistics

  #Run the MCMC simulation
  for ( i in 1:sim ){

        # generating a move
         w <- deg4move(d3)

         # z : move
      z <- w[[1]]
         y <- x3 + z

         # cell : nonzero cells
         cell <- w[[2]]

         #MH procedure
         if(min(y) >= 0){
                 r <- accept.ratio(cell,z,x3)
                 if(runif(1)<r){
                         x3 <- y
                 }
      }

      #optional check of sufficient statistic maintained
```

```
#      if (extra.var == 3)
#        print( sum(B - a %*% aperm(x3,c(2,3,4,1))) ) #check the sufficient statistic statisfied
#      else if (extra.var == 2)
#        print( sum(B - a %*% aperm(x3,c(2,4,3,1))) ) #check the sufficient statistic statisfied
#      else
#        print( sum(B - a %*% aperm(x3,c(4,2,3,1))) ) #check the sufficient statistic statisfied

        # Estimating logit models of H_1 and H_0 for accepted table
        tab <- idata(x3, d3, extra.var)
        # H_1 model
      res1 <- glm(as.formula(paste("L ~ ", paste(names[2:4], collapse= "+"))),
              family=binomial(link="logit"), data=tab)
      # H_0 model
      res0 <- glm(as.formula(paste("L ~ ", paste(names[2:3], collapse= "+"))),
              family=binomial(link="logit"), data=tab)

        markov.chain[i] <- res0$deviance - res1$deviance
  }
  return(markov.chain)
}




###########       SIS         ############
#We need the estimated MODE (based on MLE) for Triangle Sampling

modes <- function(res0, x){
  vector.x = as.vector(aperm(x,c(2,3,4,1)))
  fixed_observations = rowSums(matrix(vector.x, ncol=2))
  n = length(vector.x)
  row1 = rep(1,n/2)
  row2 = rep(c(1,2),n/4)
  row3 = rep(c(1,1,2,2,3,3),n/12)
  row4 = rep(c(1,2,3,4), each=6)

  newdata2 = data.frame( t(rbind(row1,row2,row3,row4)) )
  colnames(newdata2) = c('L','Y1','Y2','Y3')
  p.hat = predict(res0, newdata2, type="response")
  mode1 = p.hat*fixed_observations
  mode2 = fixed_observations - mode1
  mle = as.vector(c(mode1,mode2))

  return(mle)
}


#Sampling Step.  Algorithm 3.7.4, Step 3c.
sampling <- function(Aprime,bprime,rand_col,method,mode=0){
  #Using Rcplex LIBRARY for this algoritm
  #Solve a problem of the form: min/max x_j s.t. Ax=b, x>=0

  #create the objective function (min/max x1)
  f.obj <- rep(0, dim(Aprime)[2])
  f.obj[rand_col] <- 1

  #the 'sense' of the constraints are equality constraints
  #the 'type' of variables allowed are integers
  #the value for the single variable in the objective function is the objective function
  lower_bound <- Rcplex(cvec=f.obj, Amat=Aprime, bvec=bprime, Qmat=NULL,
    lb=rep(0, dim(Aprime)[2]), ub=rep(Inf,dim(Aprime)[2]), objsense=c("min"), sense="E", vtype="I", n=1,
    control=list(trace=0))$obj
  lower_bound <- round(lower_bound,0)   #simply ensuring bound is integer

  upper_bound <- Rcplex(cvec=f.obj, Amat=Aprime, bvec=bprime, Qmat=NULL,
    lb=rep(0,dim(Aprime)[2]), ub=rep(Inf,dim(Aprime)[2]), objsense=c("max"), sense="E", vtype="I", n=1,
    control=list(trace=0))$obj
  upper_bound <- round(upper_bound,0)  #simply ensuring bound is integer
  if (method == ’uniform’){                              #UNIFORM
      if (is.na(lower_bound) | is.na(upper_bound)){
        x_star <- 0                                             #hole
      }else if (upper_bound < lower_bound){
        x_star <- 0                                             #something went wrong
      }else if (upper_bound == lower_bound){
            x_star <- lower_bound
      }else{
            x_star <- sample(seq(lower_bound,upper_bound,1),1)
      }
 }else if (method == ’hyper’){                         #HYPERGEOMETRIC
      if (is.na(lower_bound) | is.na(upper_bound)){
        x_star <- 0                                             #hole
      }else if (upper_bound < lower_bound){
        x_star <- 0                                             #something went wrong
      }else if (upper_bound == lower_bound){
            x_star <- lower_bound
      }else{
        x_star <- rhyper(1,m=upper_bound,n=upper_bound,k=lower_bound+upper_bound)
      }
  }else{
      if (is.na(lower_bound) | is.na(upper_bound)){     #TRIANGLE
          x_star <- 0                                             #hole
      }else if (upper_bound < lower_bound){
          x_star <- 0                                             #something went wrong
      }else if (upper_bound == lower_bound){
              x_star <- lower_bound
      }else if (mode < lower_bound){
              lower <- round(mode,0)                               #round mode to nearest integer
          x_star <- round(rtriangle(1, lower-.5, upper_bound+.49, lower-.5), 0)
            x_star <- max(x_star,lower_bound)                    #x_star cannot be less than lower bound
      }else if (mode > upper_bound){
          upper <- round(mode,0)
          x_star <- round( rtriangle(1, lower_bound-.5,upper+.49, upper+.49), 0)
          x_star <- min(upper_bound, x_star)                   #x_star cannot be more than upper bound
      } else {
        x_star <- round( rtriangle(1,lower_bound-.5,upper_bound+.49, mode), 0)
      }
  }
  return(x_star)
}


#Matrix reduction step.  Algorithm 3.7.3.  Step 3e of Algorithm 3.7.4.
matrix_reduction <- function(Aprime,bprime,x_star,rand_col){
        A1 <- Aprime[,rand_col]          #A1 is the randomly chosen column A
        Aprime <- Aprime[,-rand_col]     #Aprime are the remaining columns
        bprime <- bprime - A1*x_star
      #print(bprime)
        output <- list(Aprime, bprime)
        return(output)                   #return Aprime and bprime
}


#SIS generation.  Algorithm 3.7.4.
#Input: Lawrence Lifting Matrix A, sufficient statistic b, sampling method,
#       randomly (or sequentially) select columns, the mode for triangle sampling
SIS <- function(A, b, method, random=TRUE, mle=0) {
  mode = 0  #initialize mode to 0

  m <- nrow(A)
  n <- ncol(A)    #dim(A)[2]  #the number of xs we solve for

  #create a vector to store the solution (x)
  y <- rep(NA, n)
  used_columns <- rep(0,n)
  Aprime <- A           #initialize Aprime
  bprime <- b           #initialize brime

  for (i in 1:(n-1)){
    if(random==TRUE)
      rand_col = sample(dim(Aprime)[2],1) #used for random column selection
    else
      rand_col = 1      #always choose the first column for sequential sampling

    true_col = which(used_columns == 0)[rand_col]
    used_columns[true_col] = 1

    if (method=='triangle')
      mode = mle[rand_col]

    x_star <- sampling(Aprime,bprime,rand_col,method,mode)
    y[true_col] <- x_star

    #Update new A matrix and b
    output <- matrix_reduction(Aprime,bprime,x_star,rand_col)
          Aprime <- output[[1]]
          bprime <- output[[2]]
  }

  #for i = n (the last cell)
  Aprime <- matrix(Aprime)
  rand_col = sample(dim(Aprime)[2],1)
  true_col = which(used_columns == 0)[rand_col]
  used_columns[true_col] = 1

  if (method=='triangle')
    mode = mle[rand_col]

  x_star <- sampling(Aprime,bprime,rand_col,method,mode)
  y[true_col] <- x_star

  return(y)
}




#######################################
######   SIS and MCMC Hybrid   #####
#Thinning and burn-in
#Step 2c-d of Algorithm 3.9.1
thinning = function(LR.matrix,B,sim,Q,k){
  m = B + 1
  LR2 = matrix(NA, nrow=(sim-B)/Q, ncol=k)

  i = 1
  while (m <= sim){
    LR2[i,] = LR.matrix[m,]
    m = m + Q
    i = i + 1
  }
  return(LR2)
}


#Algorithm 3.9.1
sis.mcmc.hybrid = function(s, v, extra.var, method, random, sim=100, k=10, B=0, Q=20, T){
  start_time <- Sys.time()                      #Used for timing runtime
  rejections = 0                                #start the rejection counter

  LR.matrix <- matrix(NA, nrow=sim, ncol=k)   #set up a matrix to store likelihood ratios, each column is Markov chain

  x <- build.array(s,v,T)
  d <- dim(x)

  if (extra.var == 3){
    x2 <- aperm(x,c(1,2,3,4))
    d2 <- dim(x2)
  }else if (extra.var == 2){
    x2 <- aperm(x,c(1,2,4,3))
    d2 <- dim(x2)
  }else{
    x2 <- aperm(x,c(1,3,4,2))
    d2 <- dim(x2)
  }

  tab <- idata(x2, d2, extra.var)
  names = colnames(tab)

  # H_1 model
  res1 <- glm(as.formula(paste("L ~ ", paste(names[2:4], collapse= "+"))),
              family=binomial(link="logit"), data=tab)
  # H_0 model
  res0 <- glm(as.formula(paste("L ~ ", paste(names[2:3], collapse= "+"))),
              family=binomial(link="logit"), data=tab)
  test.statistic <- res0$deviance - res1$deviance
  print(paste("Test Statistic: ", test.statistic))

  #create Lawrence Lifting matrix
  A.2 = create.configuration(d[2])
  A.3 = create.configuration(d[3])
  A.4 = create.configuration(d[4])
  A <- lawrence.lift(A.4,A.3,A.2)

  output <- sufficient.statistic(A,x,extra.var)
  A <- output[[1]]
  b <- output [[2]]    #sufficient statistic
  print(paste("First 3 elements of sufficient statistic: ", b[1:3]))

  if (method==’triangle’)
    mle = modes(res0,x)
  else
    mle = 0

  #Generate k independent SIS starting points (trials)
  for (trial in 1:k) {
    success <- FALSE         #Start with not satisfying sufficient statistic
    while(!success) {
      y <- SIS(A,b,method,random,mle)
      check_sum <- sum( abs(b - A %*% y) )
      if (check_sum < 1) {
        success <- TRUE              #Independent SIS starting point generated!
        print(c(’SIS trial’,trial))     #Status print message
        xx <- array(y, c(2,3,4,2))      #Put into correct dimensions
        xx <- aperm(xx,c(4,1,2,3))       #and permute
      } else {                       #ELSE you found a HOLE
        rejections = rejections + 1     #increment rejection counter
      }
    }
    LR.matrix[,trial] <- mcmc.sampling(xx, sim, extra.var,A,b)
  }

  thinned.matrix <- thinning(LR.matrix,B,sim,Q,k)

  end_time <- Sys.time()
  run_time <- end_time - start_time

  output = list(test.statistic, thinned.matrix, rejections, run_time)
  return(output)
}
```

```
###########       MAIN        ############
#INPUT PARAMETERS
s = 2
v = 2
extra.var = 2
method = ’uniform’
random = TRUE
sim = 4400         #number of MCMC runs
k = 100            #number of independent SIS tables
B <- 200           #burn-in
Q <- 20            #thinning interval


trial.1.1.1.u = sis.mcmc.hybrid(1, 1, 1, ’uniform’, random, sim, k, B, Q, T)
trial.1.1.1.h = sis.mcmc.hybrid(1, 1, 1, ’hyper’, random, sim, k, B, Q, T)
trial.1.1.1.t = sis.mcmc.hybrid(1, 1, 1, ’triangle’, random, sim, k, B, Q, T)

trial.1.1.2.u = sis.mcmc.hybrid(1, 1, 2, ’uniform’, random, sim, k, B, Q, T)
trial.1.1.2.h = sis.mcmc.hybrid(1, 1, 2, ’hyper’, random, sim, k, B, Q, T)
trial.1.1.2.t = sis.mcmc.hybrid(1, 1, 2, ’triangle’, random, sim, k, B, Q, T)

trial.1.1.3.u = sis.mcmc.hybrid(1, 1, 3, ’uniform’, random, sim, k, B, Q, T)
trial.1.1.3.h = sis.mcmc.hybrid(1, 1, 3, ’hyper’, random, sim, k, B, Q, T)
trial.1.1.3.t = sis.mcmc.hybrid(1, 1, 3, ’triangle’, random, sim, k, B, Q, T)

trial.1.2.1.u = sis.mcmc.hybrid(1, 2, 1, ’uniform’, random, sim, k, B, Q, T)
trial.1.2.1.h = sis.mcmc.hybrid(1, 2, 1, ’hyper’, random, sim, k, B, Q, T)
trial.1.2.1.t = sis.mcmc.hybrid(1, 2, 1, ’triangle’, random, sim, k, B, Q, T)

trial.1.2.2.u = sis.mcmc.hybrid(1, 2, 2, ’uniform’, random, sim, k, B, Q, T)
trial.1.2.2.h = sis.mcmc.hybrid(1, 2, 2, ’hyper’, random, sim, k, B, Q, T)
trial.1.2.2.t = sis.mcmc.hybrid(1, 2, 2, ’triangle’, random, sim, k, B, Q, T)

trial.1.2.3.u = sis.mcmc.hybrid(1, 2, 3, ’uniform’, random, sim, k, B, Q, T)
trial.1.2.3.h = sis.mcmc.hybrid(1, 2, 3, ’hyper’, random, sim, k, B, Q, T)
trial.1.2.3.t = sis.mcmc.hybrid(1, 2, 3, ’triangle’, random, sim, k, B, Q, T)

trial.1.3.1.u = sis.mcmc.hybrid(1, 3, 1, ’uniform’, random, sim, k, B, Q, T)
trial.1.3.1.h = sis.mcmc.hybrid(1, 3, 1, ’hyper’, random, sim, k, B, Q, T)
trial.1.3.1.t = sis.mcmc.hybrid(1, 3, 1, ’triangle’, random, sim, k, B, Q, T)

trial.1.3.2.u = sis.mcmc.hybrid(1, 3, 2, ’uniform’, random, sim, k, B, Q, T)
trial.1.3.2.h = sis.mcmc.hybrid(1, 3, 2, ’hyper’, random, sim, k, B, Q, T)
trial.1.3.2.t = sis.mcmc.hybrid(1, 3, 2, ’triangle’, random, sim, k, B, Q, T)

trial.1.3.3.u = sis.mcmc.hybrid(1, 3, 3, ’uniform’, random, sim, k, B, Q, T)
trial.1.3.3.h = sis.mcmc.hybrid(1, 3, 3, ’hyper’, random, sim, k, B, Q, T)
trial.1.3.3.t = sis.mcmc.hybrid(1, 3, 3, ’triangle’, random, sim, k, B, Q, T)

trial.2.1.1.u = sis.mcmc.hybrid(2, 1, 1, ’uniform’, random, sim, k, B, Q, T)
trial.2.1.1.h = sis.mcmc.hybrid(2, 1, 1, ’hyper’, random, sim, k, B, Q, T)
trial.2.1.1.t = sis.mcmc.hybrid(2, 1, 1, ’triangle’, random, sim, k, B, Q, T)

trial.2.1.2.u = sis.mcmc.hybrid(2, 1, 2, ’uniform’, random, sim, k, B, Q, T)
trial.2.1.2.h = sis.mcmc.hybrid(2, 1, 2, ’hyper’, random, sim, k, B, Q, T)
trial.2.1.2.t = sis.mcmc.hybrid(2, 1, 2, ’triangle’, random, sim, k, B, Q, T)

trial.2.1.3.u = sis.mcmc.hybrid(2, 1, 3, ’uniform’, random, sim, k, B, Q, T)
trial.2.1.3.h = sis.mcmc.hybrid(2, 1, 3, ’hyper’, random, sim, k, B, Q, T)
trial.2.1.3.t = sis.mcmc.hybrid(2, 1, 3, ’triangle’, random, sim, k, B, Q, T)
trial.2.2.1.u = sis.mcmc.hybrid(2, 2, 1, ’uniform’, random, sim, k, B, Q, T)
trial.2.2.1.h = sis.mcmc.hybrid(2, 2, 1, ’hyper’, random, sim, k, B, Q, T)
trial.2.2.1.t = sis.mcmc.hybrid(2, 2, 1, ’triangle’, random, sim, k, B, Q, T)

trial.2.2.2.u = sis.mcmc.hybrid(2, 2, 2, ’uniform’, random, sim, k, B, Q, T)
trial.2.2.2.h = sis.mcmc.hybrid(2, 2, 2, ’hyper’, random, sim, k, B, Q, T)
trial.2.2.2.t = sis.mcmc.hybrid(2, 2, 2, ’triangle’, random, sim, k, B, Q, T)

trial.2.2.3.u = sis.mcmc.hybrid(2, 2, 3, ’uniform’, random, sim, k, B, Q, T)
trial.2.2.3.h = sis.mcmc.hybrid(2, 2, 3, ’hyper’, random, sim, k, B, Q, T)
trial.2.2.3.t = sis.mcmc.hybrid(2, 2, 3, ’triangle’, random, sim, k, B, Q, T)

trial.2.3.1.u = sis.mcmc.hybrid(2, 3, 1, ’uniform’, random, sim, k, B, Q, T)
trial.2.3.1.h = sis.mcmc.hybrid(2, 3, 1, ’hyper’, random, sim, k, B, Q, T)
trial.2.3.1.t = sis.mcmc.hybrid(2, 3, 1, ’triangle’, random, sim, k, B, Q, T)

trial.2.3.2.u = sis.mcmc.hybrid(2, 3, 2, ’uniform’, random, sim, k, B, Q, T)
trial.2.3.2.h = sis.mcmc.hybrid(2, 3, 2, ’hyper’, random, sim, k, B, Q, T)
trial.2.3.2.t = sis.mcmc.hybrid(2, 3, 2, ’triangle’, random, sim, k, B, Q, T)

trial.2.3.3.u = sis.mcmc.hybrid(2, 3, 3, ’uniform’, random, sim, k, B, Q, T)
trial.2.3.3.h = sis.mcmc.hybrid(2, 3, 3, ’hyper’, random, sim, k, B, Q, T)
trial.2.3.3.t = sis.mcmc.hybrid(2, 3, 3, ’triangle’, random, sim, k, B, Q, T)

trial.3.1.1.u = sis.mcmc.hybrid(3, 1, 1, ’uniform’, random, sim, k, B, Q, T)
trial.3.1.1.h = sis.mcmc.hybrid(3, 1, 1, ’hyper’, random, sim, k, B, Q, T)
trial.3.1.1.t = sis.mcmc.hybrid(3, 1, 1, ’triangle’, random, sim, k, B, Q, T)

trial.3.1.2.u = sis.mcmc.hybrid(3, 1, 2, ’uniform’, random, sim, k, B, Q, T)
trial.3.1.2.h = sis.mcmc.hybrid(3, 1, 2, ’hyper’, random, sim, k, B, Q, T)
trial.3.1.2.t = sis.mcmc.hybrid(3, 1, 2, ’triangle’, random, sim, k, B, Q, T)

trial.3.1.3.u = sis.mcmc.hybrid(3, 1, 3, ’uniform’, random, sim, k, B, Q, T)
trial.3.1.3.h = sis.mcmc.hybrid(3, 1, 3, ’hyper’, random, sim, k, B, Q, T)
trial.3.1.3.t = sis.mcmc.hybrid(3, 1, 3, ’triangle’, random, sim, k, B, Q, T)

trial.3.2.1.u = sis.mcmc.hybrid(3, 2, 1, ’uniform’, random, sim, k, B, Q, T)
trial.3.2.1.h = sis.mcmc.hybrid(3, 2, 1, ’hyper’, random, sim, k, B, Q, T)
trial.3.2.1.t = sis.mcmc.hybrid(3, 2, 1, ’triangle’, random, sim, k, B, Q, T)

trial.3.2.2.u = sis.mcmc.hybrid(3, 2, 2, ’uniform’, random, sim, k, B, Q, T)
trial.3.2.2.h = sis.mcmc.hybrid(3, 2, 2, ’hyper’, random, sim, k, B, Q, T)
trial.3.2.2.t = sis.mcmc.hybrid(3, 2, 2, ’triangle’, random, sim, k, B, Q, T)

trial.3.2.3.u = sis.mcmc.hybrid(3, 2, 3, ’uniform’, random, sim, k, B, Q, T)
trial.3.2.3.h = sis.mcmc.hybrid(3, 2, 3, ’hyper’, random, sim, k, B, Q, T)
trial.3.2.3.t = sis.mcmc.hybrid(3, 2, 3, ’triangle’, random, sim, k, B, Q, T)

trial.3.3.1.u = sis.mcmc.hybrid(3, 3, 1, ’uniform’, random, sim, k, B, Q, T)
trial.3.3.1.h = sis.mcmc.hybrid(3, 3, 1, ’hyper’, random, sim, k, B, Q, T)
trial.3.3.1.t = sis.mcmc.hybrid(3, 3, 1, ’triangle’, random, sim, k, B, Q, T)

trial.3.3.2.u = sis.mcmc.hybrid(3, 3, 2, ’uniform’, random, sim, k, B, Q, T)
trial.3.3.2.h = sis.mcmc.hybrid(3, 3, 2, ’hyper’, random, sim, k, B, Q, T)
trial.3.3.2.t = sis.mcmc.hybrid(3, 3, 2, ’triangle’, random, sim, k, B, Q, T)

trial.3.3.3.u = sis.mcmc.hybrid(3, 3, 3, ’uniform’, random, sim, k, B, Q, T)
trial.3.3.3.h = sis.mcmc.hybrid(3, 3, 3, ’hyper’, random, sim, k, B, Q, T)
trial.3.3.3.t = sis.mcmc.hybrid(3, 3, 3, ’triangle’, random, sim, k, B, Q, T)
```

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

Agresti A (2002) *Categorical Data Analysis* (New York, NY: Wiley), 2nd ed. edition.

Besag J, Clifford P (1989) Generalized Monte Carlo significance tests. *Biometrika* 76(25):633–642.

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

Chen Y, Dinwoodie I, Sullivant S (2006) Sequential importance sampling for multiway tables. *Annals of Statistics* 34(1):523–545.

Conover W (1999) *Practical Nonparametric Statistics*, 201–202 (New York, NY: Wiley), 3rd ed. edition.

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

Diaconis P, Efron B (1985) Testing for independence in a two-way table: New interpretations of the chi-square statistic (with discussion). *Annals of Statistics* 13:845–913.

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

Dobrow R (2016) *Introduction to Stochastic Processes with R* (Hoboken, NJ: Wiley).

Drton M, Sturmfels B, Sullivant S (2009) Markov bases. *Lectures on Algebraic Statistics*, volume 39: Oberwolfach Seminars, 1–28 (Basel, Switzerland: Birkhauser).

Fam PS (2012) Analysis of two-way contingency tables. *Applied Mathematical Sciences* 6(79):3917–3925.

Fisher RA (1922) On the interpretation of χ^2 from contingency tables, and the calculation of P. *Journal of the Royal Statistical Society* 85(1):87–94, https://doi.org/10.2307/2340521.

Grayson D, Stillman M (2017) Macaulay2, a software system for research in algebraic geometry. Available at https://faculty.math.illinois.edu/Macaulay2/.

Guo S, Thompson E (1992) Performing the exact test of Hardy–Weinberg proportion for multiple alleles. *Biometrics* 48:361–372.

Haberman SJ (1988) A warning on the use of chi-squared statistics with frequency tables with small expected cell counts. *Journal of the American Statistical Association* 83(402):555–560, URL http://www.jstor.org/stable/2288877, DOI: 10.2307/2288877.

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

Hastings WK (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1):97–109.

Kahle D, O'Neill C, Sommars J (2017a) A computer algebra system for r: Macaulay2 and the m2r package. Available at https://arxiv.org/pdf/1706.07797.pdf.

Kahle D, Yoshida R, Garcia-Puente L (2017b) Hybrid schemes for exact conditional inference in discrete exponential families. *Annals of Institute of Statistical Mathematics* 1–29, DOI: 10.1007/s10463-017-0615-z.

Kateri M (2014) Contingency table analysis: Methods and implementation using R. Balakrishnan N, ed., *Statistics for Industry and Technology*, 1 (New York, NY: Springer), DOI: 10.1007/978-0-8176-4811-4_1.

Lenstra AK, Lenstra HW, Lovász L (1982) Factoring polynomials with rational coefficients. *Mathematische Annalen* 261(4):515–534, DOI: 10.1007/BF01457454.

Link W, Eaton M (2012) On thinning of chains in MCMC. *Methods in Ecology and Evolution* 3:112–115, DOI: 10.1111/j.2041-210X.2011.00131.x.

Monaghan E (2016) *Estimating the depth of the Navy recruiting market*. M.s. thesis, Naval Postgraduate School, Monterey, CA.

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

National Institute on Aging (2015) *Alzheimer's disease genetics fact sheet*. Bethesda, MD, https://www.nia.nih.gov/health/alzheimers-disease-genetics-fact-sheet.

Salazar J, Schmitt F, Yu L, Mendiondo M (2007) Shared random effects analysis of multistate Markov models: Application to a longitudinal study of transitions to dementia. *Statistics in Medicine* 26:568–580.

Schnorr CK (1987) A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science* 53:201–224.

Schrijver A (1986) *Theory of Linear and Integer Programming* (Chichester, West Sussex, England: Wiley), 1st ed. edition.

Steorts R (2016) Intro to Markov Chain Monte Carlo. Class notes, Duke University course STA 360/601, Durham, NC. http://www2.stat.duke.edu/~rcs46/lecturesModernBayes/601-module6-markov/markov-chain-monte-carlo.pdf.

Tyas S, Salazar J, Snowdon D, Desrosiers M, Riley K, Mendiondo M, Kryscio R (2007) Transitions to mild cognitive impairments, dementia, and death: Findings from the nun study. *American Journal of Epidemiology* 165(11):1231–1238.

Wei S, Kryscio R (2016) Semi-Markov models for interval censored transient cognitive states with back transitions and a competing risk. *Statistical Methods in Medical Research* 25(6):2909–2924, DOI 10.1177/0962280214534412.

White L (2009) *Predicting hospital admissions with Poisson regression analysis*. M.s. thesis, Naval Postgraduate School, Monterey, CA.

Xi J, Yoshida R, Haws D (2013) Estimating the number of zero-one multi-way tables via sequential importance sampling. *Annals of Institute of Statistical Mathematics* 1–29, DOI: 10.1007/s10463-017-0615-z.

Xie Z (2016) *Topics in Logistic Regression Analysis*. Ph.D. thesis, Dept. Statistics, Univ. Kentucky, Lexington, KY.

Yoshida R (2010) Connectivity of fibers with positive margins in multi-dimensional contingency tables. Presentation, The Second CREST-SBM International Conference Harmony of Groebner bases and the modern industrial society, Osaka, Japan. http://polytopes.net/pdf/crest.pdf.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California