NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A HYBRID SAMPLING METHOD FOR THREE-WAY CONTINGENCY TABLES

by

Aaron Stone

June 2018

Thesis Advisor:	Ruriko Yoshida
Second Reader:	Matthew Norton

Approved for public release. Distribution is unlimited.

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

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** June 2018	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE A HYBRID SAMPLING METHOD FOR THREE-WAY CONTINGENCY TABLES	**5. FUNDING NUMBERS**
6. AUTHOR(S) Aaron Stone	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**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 thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

We develop an algorithm blending Sequential Importance Sampling (SIS) and Markov Chain Monte Carlo (MCMC) to conduct goodness of fit testing on three-way contingency tables under the no-three-way interaction model. Unlike previous studies, we conduct SIS utilizing the hypergeometric distribution. Further, our hybrid method capitalizes on the positive aspects of SIS and MCMC while reducing their inefficiencies. We demonstrate the algorithm's performance on equal marginal data sets to highlight computational speed and accuracy. We then demonstrate the algorithm in accurately constructing the null distribution for dense tables that satisfy the asymptotic distribution assumptions. With this result in mind, we estimate the null distribution for sparse tables that violate these assumptions. Our hybrid scheme is shown, via simulation, to be more accurate than simply using the asymptotic distribution for sparse tables.

14. SUBJECT TERMS Sequential Importance Sampling, Markov Chain Monte Carlo, contingency table, exact inference	**15. NUMBER OF PAGES** 65
	16. PRICE CODE

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

THIS PAGE INTENTIONALLY LEFT BLANK

A HYBRID SAMPLING METHOD FOR THREE-WAY CONTINGENCY TABLES

Aaron Stone
Captain, United States Marine Corps
BS, Michigan State University, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2018

Approved by: Ruriko Yoshida
Advisor

Matthew Norton
Second Reader

Patricia A. Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

We develop an algorithm blending Sequential Importance Sampling (SIS) and Markov Chain Monte Carlo (MCMC) to conduct goodness of fit testing on three-way contingency tables under the no-three-way interaction model. Unlike previous studies, we conduct SIS utilizing the hypergeometric distribution. Further, our hybrid method capitalizes on the positive aspects of SIS and MCMC while reducing their inefficiencies. We demonstrate the algorithm's performance on equal marginal data sets to highlight computational speed and accuracy. We then demonstrate the algorithm in accurately constructing the null distribution for dense tables that satisfy the asymptotic distribution assumptions. With this result in mind, we estimate the null distribution for sparse tables that violate these assumptions. Our hybrid scheme is shown, via simulation, to be more accurate than simply using the asymptotic distribution for sparse tables.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

IPF	Iterative Proportional Fitting
MCMC	Markov Chain Monte Carlo
MLE	Maximum Likelihood Estimator
SIS	Sequential Importance Sampling

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

A *contingency table*, **X**, is one of the oldest objects which researchers use to analyze associations or interrelations between discrete random variables. In this research we assume that each random variable has a finite number of categories, which are often called “levels.” A “cell” is a particular event with these discrete random variables and a “cell count” is the number of the particular events observed in the given sample.

This research focuses on conducting a goodness of fit test under the no-three-way interaction model for $I \times J \times K$ contingency tables where $I,\ J,\ K \geq 2$. We consider “dense” contingency tables and “sparse” contingency tables. A “dense” contingency table is a table where the Maximum Likelihood Estimator (MLE) for each cell count under a particular model greater than five. In this instance, we can use the χ^2 distribution, which is an asymptotic distribution of the null distribution of test statistics for a goodness of fit test. A contingency table can be “sparse” and have a small conditional state space. By “sparse” we mean not all of the MLE for each cell count under the null model is greater than five. Therefore, we cannot use the asymptotic distribution as the null distribution for the goodness of fit test. In this case, Fisher’s exact test provides an accurate estimation of the p-value for the test. A contingency table can be “sparse” and have a large conditional state space. In this situation, we cannot use the asymptotic distribution as the null distribution and we cannot use Fisher’s exact test because Fisher’s exact test has to enumerate all contingency tables in the conditional state space. These types of tables require a sampling procedure in order to conduct a goodness of fit test. In this research we develop this sampling method.

Our sampling method is a hybrid of two popular sampling methods, Sequential Importance Sampling (SIS) and Markov Chain Monte Carlo (MCMC). We modify the SIS method developed by Chen et. al, in 2005. This method is a recursive algorithm that populates a contingency table cell by cell while conditioning on the marginal sums of the observed table. Our SIS method runs in a similar fashion except we conduct sampling for $I \times J \times K$ tables and we sample each cell count from the hypergeometric distribution instead of the uniform distribution. Our MCMC method is based on the work of Diaconis and Sturmfels in 1998. This method operates by conducting a series of basic moves and a metropolis algorithm to traverse the conditional state space. The addition of the metropolis algorithm ensures that the MCMC sampling is also from the hypergeometric distribution.

By conducting a hybrid method of SIS and MCMC we are able to leverage the positive aspects of both methods while balancing out the weaknesses. For instance, the MCMC method creates autocorrelation. This requires significant burn-in and thinning, which means that in order to reach a desired sample size, we have to sample many more tables and we have to discard most of the sampled tables for burn-in and thinning processes. The SIS method, on the other hand, samples tables which are independently and identically distributed. In addition, A Markov chain with basic moves is not guaranteed to be connected in the conditional state space. In order to make sure the chain is connected, we use the SIS procedure to sample initial contingency tables and then using such tables as initial states, we run multiple chains in the conditional state space. Doing this, we can sample contingency tables from the state space without sampling bias.

We demonstrate, via simulation, that our sampling method accurately estimates the asymptotic distribution for a “dense” table. With this result in mind, with “sparse” tables, we demonstrate that our novel method can estimate accurately the null distribution than using the asymptotic distribution.

Lastly, our hybrid sampling method has usability to any field interested in conducting goodness of fit testing. There are no restrictions to its applicability. Instead of answering a question, our hybrid methodology solves a problem.

Acknowledgments

I would like to thank my advisors for guiding me through one of the most difficult ventures I have undertaken in my life. It is in no way hyperbolic to say that without their guidance, mentoring, and assistance, I would still be staring at a blank page wondering how to begin. I felt like an equal partner in this process able to try new ideas and most importantly, make the mistakes that allowed me to grow intellectually.

I would like to thank my second reader for helping to develop a coherent story that the reader may follow throughout this work.

Finally, I would like to thank Bailey for always being free to take walks and listen while I tried to work through some of the coding issues I encountered.

THIS PAGE INTENTIONALLY LEFT BLANK

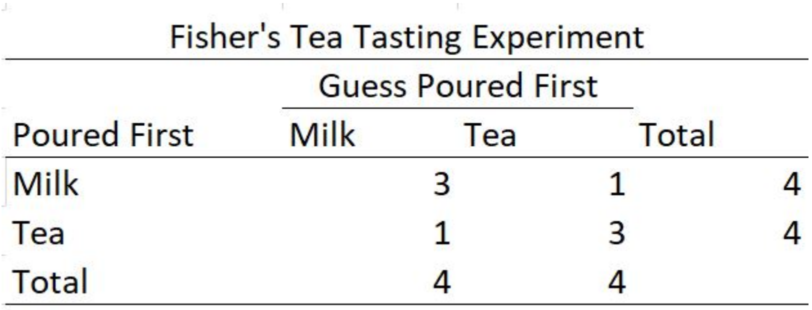
CHAPTER 1:
Introduction

In this chapter we provide background, motivation, and the novel aspects of our methodology.

1.1 Background

A *contingency table*, **X**, is one of the oldest objects researchers use to analyze associations or interrelations between discrete random variables [1]. In this research we assume that each random variable has a finite number of categories, which are often called “levels.” A “cell” is a particular event with these discrete random variables and a “cell count” is the number of times the particular event was observed in the given sample [2].

To illustrate a contingency table, we can consider a famous example problem from 1935. Muriel Bristol claimed, given a cup of tea with milk added, she had the ability to taste whether milk or tea was added to the cup first. Ronald Fisher developed an experiment to test her ability. He made eight cups of tea with milk, four in which milk was poured in first and four in which tea was poured in first. He then randomly gave these tea cups to Bristol and recorded her responses. Figure 1.1 is this 2×2 contingency table [2].



Fisher's Tea Tasting Experiment

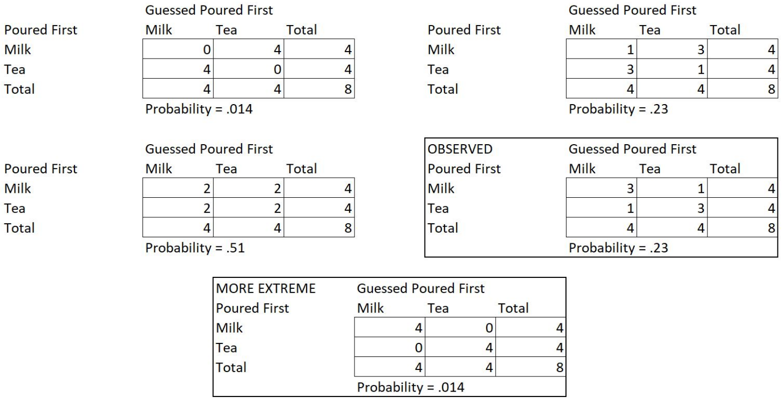
	Guess Poured First		
Poured First	Milk	Tea	Total
Milk	3	1	4
Tea	1	3	4
Total	4	4	

Figure 1.1. Fisher's Tea Test

In Figure 1.1, the "levels" are whether milk or tea was poured into the cup first. The “cell” Milk/Milk, or X_{11}, has the “cell count” of three. This means in three cases that milk was poured first, it was guessed as being poured first.

In this example, we would like to test the hypothesis that Bristol can truly distinguish the difference in how the cups of tea were made. If we were to conduct a χ^2 test of independence on Figure 1.1, the p-value would be .4795. But this table does not meet the assumption for the asymptotic distribution. For this reason, we would utilize Fisher's exact test.

Fisher's exact test is accomplished by enumerating all possible contingency tables in the conditional state space. The conditional state space is all tables that have the same marginal sums as the observed table, in this case all marginal sums are four. We use the marginal sums as our condition because the marginal sums are the sufficient statistics to determine a contingency table's Maximum Likelihood Estimator (MLE). The complete state space for this table is shown in Figure 1.2.



	Guessed Poured First		
Poured First	Milk	Tea	Total
Milk	0	4	4
Tea	4	0	4
Total	4	4	8
	Probability = .014		

	Guessed Poured First		
Poured First	Milk	Tea	Total
Milk	1	3	4
Tea	3	1	4
Total	4	4	8
	Probability = .23		

	Guessed Poured First		
Poured First	Milk	Tea	Total
Milk	2	2	4
Tea	2	2	4
Total	4	4	8
	Probability = .51		

OBSERVED	Guessed Poured First		
Poured First	Milk	Tea	Total
Milk	3	1	4
Tea	1	3	4
Total	4	4	8
	Probability = .23		

MORE EXTREME	Guessed Poured First		
Poured First	Milk	Tea	Total
Milk	4	0	4
Tea	0	4	4
Total	4	4	8
	Probability = .014		

Figure 1.2. Fisher's Tea Test

To complete the exact test, we calculate the probability of seeing the tables that exist in the conditional state space. These probabilities are calculated using the hypergeometric distribution. The hypergeometric distribution can be described in a situation where we have green and red balls in an urn and we want to calculate the probability of drawing a certain number of green balls given that we draw some number of balls without replacement [1]. To calculate the p-value, we sum the probabilities of seeing the observed table and any table more extreme. In this case, more extreme would be arrangements of the table where Bristol could have identified more cups of tea correctly than she did. These values, which are listed below each table in Figure 1.2, sum to a p-value of .24 [2].

What this example has shown us is that even with a small contingency table, ignoring the asymptotic requirements have the potential to lead to incorrect conclusions. This example also shows a limitation of Fisher's exact test. This small table only has five tables in the conditional state space. As was shown by [3], even moderately sized tables can have trillions of possible arrangements and thus using Fisher's exact test is computationally prohibitive.

As we show in Chapter 4, an $I \times J \times K$ contingency table that has a large conditional state space and violates the cell count assumption has a different null distribution than the asymptotic distribution. Figure 1.3 shows the difference between these distributions and these differences are confirmed by usage of the Kolmogorov-Smirnov test.

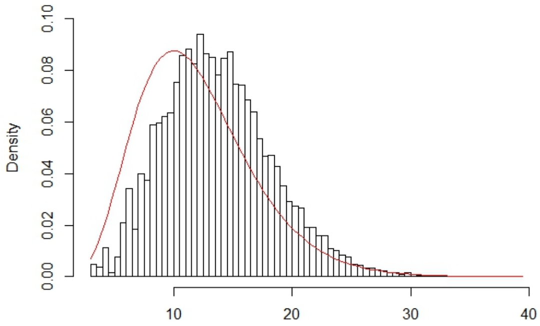


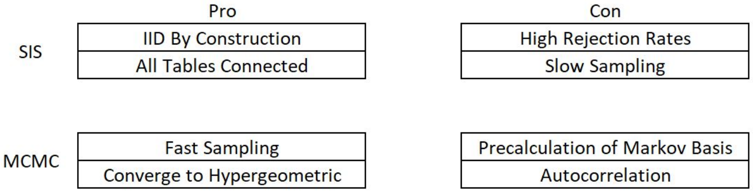
Figure 1.3. $I \times J \times K$ Null Distribution Result

1.2 Motivation

As large data sets become increasingly utilized by industry and governments, there is an ever increasing need for accurate model fitting procedures allowing for the extraction of information from these data sets. If a data set is sparse, the expected value of each cell is less than five, and enumerating all tables is computationally prohibitive then in order to extract information about the relationship between the variables a new methodology is required.

When confronted with a data set that violates the expected cell count assumption and has a computationally prohibitive conditional state space, the solution, in order to conduct accurate goodness of fit testing, is sampling. Sampling is done by randomly selecting tables from the conditional state space of the observed table. As the tables we sample are conditioned on the marginal sums, the sufficient statistics to infer the MLE, the sampled tables have a relationship with the observed table; they all have the same MLE. Therefore, we can utilize the sampled tables to construct the null distribution of test statistics and conduct hypothesis testing. There are two popular methods for sampling when utilizing contingency tables, Sequential Importance Sampling (SIS) and Markov Chain Monte Carlo (MCMC).

Kale et.al. [4], demonstrates the pros and cons of the SIS and MCMC methods for sampling contingency tables. These can be found in Figure 1.4.



	Pro	Con
SIS	IID By Construction	High Rejection Rates
	All Tables Connected	Slow Sampling
MCMC	Fast Sampling	Precalculation of Markov Basis
	Converge to Hypergeometric	Autocorrelation

Figure 1.4. SIS/MCMC Pros and Cons

Chen et. al. [5], developed and utilized a SIS method for estimating the total number of tables with given marginal sums utilizing the uniform distribution. They compared the efficiency of this SIS method with a MCMC method utilizing basic moves to sample tables from the uniform distribution. In this research, we develop a novel sampling method that blends the SIS method, sampling values from the hypergeometric distribution, with the MCMC method in order to conduct a goodness of fit test.

We propose a blend of the SIS method and the MCMC method, based on [4], in order to conduct model fitting on $I \times J \times K$ contingency tables under the no-three-way interaction model. Further, during SIS, we construct tables in the conditional state space by sampling from the hypergeometric distribution and not the uniform distribution. The algorithms we developed are fully described in Chapter 3.

1.3 Research Objectives

In the remainder of the work, we accomplish the following research objectives.

- Conduct SIS sampling from the hypergeometric distribution on $I \times J \times K$ data sets.
- Blend the SIS method with the MCMC method to create a more efficient sampler.

- Apply the blended method to $5 \times 4 \times 2$ sparse data set generated from the Poisson distribution to test the efficiency of the sampler.
- Apply the blended method to a real world data set.

1.4 Novelty

One of the novelties of our approach is applying the SIS method to three way contingency tables utilizing the hypergeometric distribution. Another novelty is the combination of the SIS method we developed in this research and MCMC methods for $I \times J \times K$ contingency tables. The blend of these two methods leads to gained efficiency's and reduction in the negative aspects of each method.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background and Literature Review

In this chapter we define the notation used throughout the paper and a brief overview of the theoretical background. This chapter also includes guidance to additional sources.

2.1 Notation

The following notation is used throughout the paper.

- Random variables are represented by capital letters; X
- Fixed values are represented by lower case letters; x
- Vectors, Arrays and Tables are represented by bold letters; **X**, **x**
- Initial tables are lower case and bold with a superscript; $\mathbf{x^0}$
- An individual cell in a table is x^0_{ijk}
- Summing over a dimension is $\mathbf{x^0_{+jk}}$. In this case we sum over the I dimension.
- Searching over a dimension is $\mathbf{x^0_{\cdot jk}}$. In this case we are looking at the values in the I dimension.

2.2 Definitions

Figure 2.1 is used to provide direction for the mathematical concepts we utilize in the construction of our SIS/MCMC blended algorithm.

2.2.1 Contingency Table

A *contingency table*, **X**, is used to analyze associations or interrelations between discrete random variables [1]. In our research we assume that each random variable has a finite number of categories, which are often called “levels.” A “cell” is a particular event with these discrete random variables and a “cell count” is the number of the particular event observed in the given sample.

In this research, we focus on three way contingency tables, $I \times J \times K$, a representation for three discrete random variables $W \in \{1, \dots, I\}$, $Y \in \{1, \dots, J\}$, and $Z \in \{1, \dots, K\}$ for

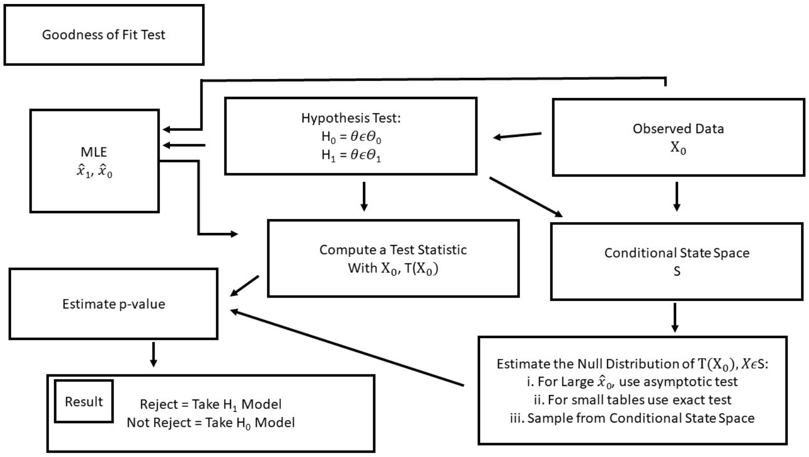

Goodness of Fit Test
MLE
$\hat{x}_1, \hat{x}_0$
Hypothesis Test:
$H_0 = \theta \epsilon \Theta_0$
$H_1 = \theta \epsilon \Theta_1$
Observed Data
X_0
Compute a Test Statistic
With X_0, $T(X_0)$
Conditional State Space
S
Estimate p-value
Estimate the Null Distribution of $T(X_0), X \epsilon S$:
i. For Large $\hat{x}_0$, use asymptotic test
ii. For small tables use exact test
iii. Sample from Conditional State Space
Result
Reject = Take H_1 Model
Not Reject = Take H_0 Model

Figure 2.1. Methodology Flow Diagram

I, J, $K \geq 2$.

When referencing **X**, individual cell counts are defined as X_{ijk}. For our analysis, we condition our data on the *marginal sums*. The row sums are defined as:

$$X_{+jk} = \sum_{i=1}^{I} X_{ijk} \text{ for } 1 \leq j \leq J,\ 1 \leq k \leq K.$$

Similarly, we can obtain the marginal sums over the second random variable Y, that is, the X_{i+k} and the marginals, X_{ij+} over the third random variable Z, which we refer to as the K way sums.

In our algorithms we represent the marginal sums as matrices. The row sum matrix is a $J \times K$ matrix where each row represents a "layer", we define "layer" as the different values of $k \in \{1, \ldots, K\}$, of $\mathbf{x^0}$ and each column represents the row number. For instance, if the dimensions of $\mathbf{x^0}$ are $3 \times 4 \times 5$ then the I-way sum matrix is a 4×5 matrix where cell 3,2 is the third row sum of $k = 2$.

2.2.2 Hypothesis Testing

A goodness of fit test is a variant of a hypothesis test which determines whether a null model or an alternative model better fits to an observed data set by measuring the difference between the observed data and the expectation under the null model [2]. If an observed data set is close to the expectation under the null model, we select the null model [2]. If it is significantly different, we select an alternative model. In order to measure this difference, we utilize a *test statistic*. In this thesis we use the χ^2 test statistic to measure the difference between the sampled table and the MLE.

In particular, we are interested in the *no-three-way interaction model*. With a given three way contingency table, $\mathbf{X}$, with categorical variables W, Y, and Z, the general log-linear model is defined as follows: Each cell count X_{ijk} is distributed according to the Poisson distribution with parameter θ such that

$$\log \theta_{ijk} = \lambda + \lambda_i^W + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{WY} + \lambda_{ik}^{WZ} + \lambda_{jk}^{YZ} + \lambda_{ijk}^{WYZ}$$

where λ_{ijk}^{WYZ} is the association parameter between the ith category in W, the jth category in Y, and the kth category in Z [2]. Under the no-three-way interaction model, $\lambda_{ijk}^{WYZ} = 0$. We are interested in conducting goodness-of-fit tests, that is, hypothesis tests with the null hypothesis

$$H_0 : \lambda_{ijk}^{WYZ} = 0.$$

In our project we consider the saturation model as the alternative hypothesis, i.e.,

$$H_1 : \lambda_{ijk}^{WYZ} \neq 0.$$

Under this model, the *sufficient statistics* are the statistics that contain enough information to infer the maximum likelihood estimator (MLE) for the parameters under this model.

For the no-three-way interaction model the sufficient statistics are, X_{+jk} for all $j \in \{1, \ldots, J\}$, $k \in \{1, \ldots, K\}$, X_{i+k} for all $i \in \{1, \ldots, I\}$, $k \in \{1, \ldots, K\}$, and X_{ij+} for all $i \in \{1, \ldots, I\}$, $j \in \{1, \ldots, J\}$ which are the *marginal sums* of $\mathbf{x^0}$.

Random variables W and Y are conditionally independent at level k of a random variable Z if:

$$P(Y = j|W = i,\ Z = k) = P(Y = j|Z = k),\ \text{for all } i = 1, \ldots I,\ \text{for all } j = 1, \ldots J.$$

Let

$$\pi_{ijk} = P(W = i,\ Y = j,\ Z = k),$$

and

$$\pi_{+jk} = P(Y = j,\ Z = k) = \sum_{i=i}^{I} P(W = i,\ Y = j,\ Z = k).$$

By the multiplication rule and Bayes rule, if W and Y are conditionally independent of Z, then

$$\begin{aligned} \pi_{ijk} &= P(W = i, Z = k)P(Y = j|W = i, Z = k) \\ &= \pi_{i+k}P(Y = j|Z = k) \\ &= \pi_{i+k}P(Y = j\ \ Z = k)/P(Z = k) \\ &= \pi_{i+k}\pi_{+jk}/\pi_{++k} \end{aligned} \tag{2.1}$$

where $\pi_{++k} = P(Z = k) = \sum_{i=1}^{I}\sum_{j=1}^{J}\pi_{ijk}$. Therefore, if W and Y are conditionally independent of Z,

$$\begin{aligned} \mu_{ijk} &= n\pi_{ijk} \\ \log(\mu_{ijk}) &= \log(n\pi_{ijk}) \\ &= \log(n) + \log(\pi_{i+k}) + \log(\pi_{+jk}) - \log(\pi_{++k}), \end{aligned} \tag{2.2}$$

where n is the sample size and μ_{ijk} is the expected cell count in cell X_{ijk}. Since we assume that each cell count is generated under the Poisson distribution, the expected cell count is a parameter for the cell count. Therefore, the sufficient statistics are the sums X_{+jk}, and X_{i+k} of the observed table $\mathbf{X}$ since one can estimate $\pi_{+jk} = X_{+jk}/n$, $\pi_{ij+} = X_{ij+}/n$, and $\pi_{i+k} = X_{i+k}/n$.

Similarly we can show that if W and Z are conditionally independent of Y, then

$$\pi_{ijk} = \pi_{ij+}\pi_{+jk}/\pi_{+j+}$$

and if Y and Z are conditionally independent of W, then

$$\pi_{ijk} \quad = \quad \pi_{ij+}\pi_{i+k}/\pi_{i++}.$$

Therefore, the sufficient statistics of $\mathbf{x^0}$ are the marginal sums since one can estimate the expected cell count $\mathbf{x^0}$ which is the MLE for the Poisson distribution for this particular cell.

2.2.3 Maximum Likelihood Estimation

Agresti, [2], defines Maximum Likelihood for parameter estimation as "given data for a chosen probability distribution the likelihood function is the probability of the data treated as a function of an unknown parameter. The maximum likelihood estimate is the parameter value that maximizes the function." In other words, it is the parameter that maximizes the probability of seeing the observed data. In our case we utilize the marginal sums in order to calculate the MLE of $\mathbf{x^0}$, which is the contingency table with the highest probability of being observed.

To determine the MLE for a given table, $\mathbf{x^0}$, we utilize an Iterative Proportional Fitting (IPF) technique as described in [6] via Algorithm 2.2.1.

Algorithm 2.2.1 Input: *The observed table* $x^0 = \left(x^0_{ijk}\right)_{1\le i\le I,\, 1\le j\le J,\, 1\le k\le K} \in Z^{I\times J\times K}$ *for* $I, J, K \in N$.

Output: *The MLE* $\mathbf{m} = \left(m_{ijk}\right)_{1\le i\le I,\, 1\le j\le J,\, 1\le k\le K}$ *under the no three-way interaction mode.*

Algorithm:

1. *Initialize* $m^1_{ijk} = 1,\ 1 \le i \le I,\ 1 \le j \le J,\ 1 \le k \le K.$
2. *Compute the marginals*

$$\begin{array}{lll} x_{ij+} = \sum_{k=1}^{K} x^0_{ijk} & \textit{for} & 1 \le i \le I,\ 1 \le j \le J. \\ x_{i+k} = \sum_{j=1}^{J} x^0_{ijk} & \textit{for} & 1 \le i \le I,\ 1 \le k \le K. \\ x_{+jk} = \sum_{i=1}^{I} x^0_{ijk} & \textit{for} & 1 \le j \le J,\ 1 \le k \le K. \end{array}$$

3. *Until convergence, iterate for* $l = 1, 2, \ldots$*:*

```
    3.1. $m_{ijk}^{3+l-1} = \frac{m_{ijk}^{3+l-2} x_{ij+}}{\sum_{k=1}^{K} m_{ijk}^{3+l-2}}$ for $1 \leq i \leq I$, $1 \leq j \leq J$,
    3.2. $m_{ijk}^{3+l} = \frac{m_{ijk}^{3+l-1} x_{i+k}}{\sum_{j=1}^{J} m_{ijk}^{3+l-1}}$ for $1 \leq i \leq I$, $1 \leq k \leq K$,
    3.3. $m_{ijk}^{3+l+1} = \frac{m_{ijk}^{3+l} x_{+jk}}{\sum_{i=1}^{I} m_{ijk}^{3+l}}$ for $1 \leq j \leq J$, $1 \leq k \leq K$,
4. Return $\mathbf{m}$.
```

2.2.4 χ^2 Test Statistic

The χ^2 test statistic is utilized to determine differences between the expected frequencies and the observed frequencies in one or more categories of a contingency table [1]. The calculation of the test statistic uses the sum of squared differences between the observed data set and the expected values of the table, or the maximum likelihood estimate, **m**. The test statistic is calculated as follows.

$$\chi^2 = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \frac{(x_{ijk}^0 - m_{ijk})^2}{m_{ijk}}.$$

2.2.5 Conditional State Space

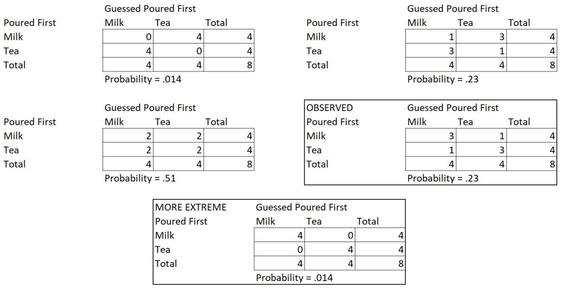
The conditional state space is the set of all tables that satisfy the marginal sums of a given data set [7]. As was described, the marginal sums are the sufficient statistics to determine the probabilities of seeing a particular X_{ijk} and constructing the MLE of the observed table. If we construct a new table, $\mathbf{x^1}$, with the same marginal sums as the observed data set, $\mathbf{x^0}$, this table exists in the conditional state space of $\mathbf{x^0}$. For this reason, we can use $\mathbf{x^1}$ and the MLE of $\mathbf{x^0}$, **m**, to calculate a test statistic. By repeating this process we can estimate the Null Distribution of $\mathbf{x^0}$. We use Fisher's Tea Test from Chapter One to illustrate an observed tables entire conditional state space. All marginal sums in Fisher's Tea Test are equal to four, Figure 2.2 is the complete conditional state space of Fisher's Tea Test. The data sets we use in our research have a conditional state space that is too large to enumerate all tables and there for we use our hybrid SIS/MCMC method to sample.

2.2.6 Null Distribution

In order to conduct conditional goodness-of-fit tests under log-linear models, usually we use asymptotic distributions. These distributions are utilized when the expected cell counts of each cell in the table are greater than five [1]. However, for sparse contingency tables, it is not appropriate to use asymptotic distributions [1] as this can lead to incorrect conclusions.

If the conditional state space is small, it is appropriate to utilize Fisher's Exact Test [1]. This was seen in the tea tasting example in Chapter 1 and again in Figure 2.2. All tables that are more extreme than the observed table are enumerated and their probability of occurring are used to construct the p-value.

To demonstrate the conditional state space and its connection to the hypergeometric distribution we use Fisher's Tea Test from Chapter 1. In Figure 2.2 we have enumerated all possible tables with equal marginal sums to what Fisher observed. We use the hypergeometric distribution, described fully below, to calculate the probability of observing each tables.



Poured First	Guessed Poured First Milk	Tea	Total
Milk	0	4	4
Tea	4	0	4
Total	4	4	8

Probability = .014

Poured First	Guessed Poured First Milk	Tea	Total
Milk	1	3	4
Tea	3	1	4
Total	4	4	8

Probability = .23

Poured First	Guessed Poured First Milk	Tea	Total
Milk	2	2	4
Tea	2	2	4
Total	4	4	8

Probability = .51

OBSERVED Poured First	Guessed Poured First Milk	Tea	Total
Milk	3	1	4
Tea	1	3	4
Total	4	4	8

Probability = .23

MORE EXTREME Poured First	Guessed Poured First Milk	Tea	Total
Milk	4	0	4
Tea	0	4	4
Total	4	4	8

Probability = .014

Figure 2.2. Fisher's Tea Test State Space

As can be seen in Figure 2.2 the observed table has a probability of occurring of .23. The only table in the conditional state space that is more extreme is if Bristol had gotten every cup of tea correct. This occurs with a probability of .014. Our p-value for this test is the sum of these two events occurring which is approximately .24.

Fisher's exact test is inappropriate if the conditional state space is large. When we have sparse contingency tables and a large conditional state space the null distribution must be estimated by sampling. The hybrid SIS/MCMC method we developed can be utilized to construct the null distribution of the observed contingency table and more accurately estimate a p-value than by ignoring the asymptotic distribution violations.

2.2.7 P-value

In our test a p-value is used to determine whether to reject H_0 and utilize H_1 or determine there is insufficient evidence to reject H_0. To calculate the p-value we utilize the following algorithm. The outline of an exact conditional test is described in Algorithm 2.2.2:

Algorithm 2.2.2 *Exact conditional test*

Input*: The observed table* $\mathbf{x^0} \in \mathbf{Z^{I \times J \times K}}$ *for* $I, J, K \in N$.

Output*: The estimated p-value of* H_0.

Algorithm:

1. *Compute the sufficient statistics from* $\mathbf{x^0}$ *for the MLE under the null model.*
2. *Compute MLE.*
3. *Compute the test statistic* $\chi^2(\mathbf{x^0})$.
4. *Sample tables* $\mathbf{x^1}, \ldots, \mathbf{x^n}$ *from the conditional state space given the sufficient statistics.*
5. *Calculate the test statistics* $\chi^2(\mathbf{x^1}), \ldots, \boldsymbol{\chi^2}(\mathbf{x^n})$ *for* $\mathbf{x^1}, \ldots, \mathbf{x^n}$*, respectively.*
6. *Estimate p-value by computing*

$$\frac{\sum_{i=1}^{n} \mathcal{I}_{\chi^2(\mathbf{x^0}) \geq \boldsymbol{\chi^2}(\mathbf{x^i})}}{n}.$$

The focus of our work is Step 4 of the ’Exact Conditional Test’ algorithm. We demonstrate how we successfully conduct a hybrid approach leveraging the positive aspects of SIS and MCMC while minimizing the drawbacks [4]. This algorithm is fully described in Chapter 3.

2.2.8 Sequential Importance Sampling

Chen et. al [5], developed an SIS method for an exact conditional test on the discrete exponential family. Chen et. al [8], focused on the independence model for two-way contingency tables and applied the SIS procedure to the *volume test* which is developed by [3] instead of a classical conditional goodness of fit test on the independence model. One of the issues of their method is only sampling contingency tables from the uniform distribution. In order to conduct a goodness of fit test, one must sample tables from the hypergeometric distribution.

Another issue with the sampling method developed in [8] is computational time. In their method, one has to compute lower and upper bounds for each cell count in a table by solving integer programming problems. Since solving an integer programming problem is NP-hard [9] and one has to solve $2 \cdot I \cdot J$ many integer programming problems for sampling a $I \times J$ contingency table, their method is computationally expensive.

Therefore, in this thesis, we propose a novel method to sample an $I \times J \times K$ table contingency from the hypergeometric distribution efficiently by blending the SIS and MCMC methods.

2.2.9 Hypergeometric Distribution

The reason we sample from the hypergeometric distribution is the relationship between contingency tables and the hypergeometric distribution [2]. As can be seen below, a contingency table is a representation of the hypergeometric distribution. In this table, A is the total number of balls in the urn, G is the number of green balls in the urn, and $A - G$ is the number of red balls in the urn. $a + x$ is the total number of balls drawn from the urn, x is the number of green balls drawn from the urn, $a - x$ is the total number of red balls drawn.

	Drawn	Not Drawn	Total
Green	x	$G - x$	G
Red	a	$A - a - G$	$A - G$
Total	$a + x$	$A - a - x$	A

The probability that we draw some value of green balls, x, given we draw $a + x$ total balls, is calculated using the hypergeometric distribution:

$$P(X = x) = \frac{\binom{G}{x}\binom{A-G}{a}}{\binom{A}{a+x}}.$$

This relationship is not limited to 2×2 tables. For three-way contingency tables the hypergeometric distribution is defined as

$$\frac{\left(\prod_i X_{i++}!\right)\left(\prod_j X_{+j+}!\right)\left(\prod_k X_{++k}!\right)}{(n!)^2 \prod_i \prod_j \prod_k X_{ijk}!}$$

where $n = X_{+++}$.

As can be seen by the 2×2 example above, the marginal sums are the sufficient statistics that can determine the probability of drawing a certain number of green balls, x, [2]. This relationship is true in $I \times J \times K$ tables as well. In our method for sampling, we utilize the marginal sums to create a $2 \times 2 \times 2$ matrix that calculates the required probabilities for sampling.

Before that can occur, we manipulate the marginal sum matrices introduced in Section 2.2.1. Algorithm 2.2.3 runs on the $\mathbf{r}$, $\mathbf{c}$, and $\mathbf{k}$ marginal sum matrices prior to sampling from the hypergeometric distribution. The sum matrix, $\mathbf{S}$, will be used to describe the algorithm because the algorithm operates on $\mathbf{r}$, $\mathbf{c}$, and $\mathbf{k}$ in the same way.

Algorithm 2.2.3 *Create 2×2 table for hypergeometric sampling*

- *Input: Marginal Sum matrix* $\mathbf{S}$.
- *Output:* 2×2 *matrix,* $\bar{\mathbf{S}}$.

  Algorithm:

  1. *Set* $\bar{S}_{11} = S_{11}$
  2. *Set* $\bar{S}_{12} = \sum_{j=1}^{J} S_{1j} - S_{11}$.
  3. *Set* $\bar{S}_{21} = \sum_{i=1}^{I} S_{i1} - S_{11}$.
  4. *Set* $\bar{S}_{22} = \sum_{j=1}^{J} \sum_{i=1}^{I} S_{ij} - \bar{S}_{21} - \bar{S}_{12} - S_{11}$.

We input each marginal sum matrix and utilize the returned $\bar{\mathbf{r}}$, $\bar{\mathbf{c}}$, $\bar{\mathbf{k}}$ to determine the hypergeometric probabilities for sampling. We populate the $2 \times 2 \times 2$, which we define as ζ, in the following way:

K = 1	C1	C2
R1	x	$\bar{r}_{11} - x$
R2	$\bar{c}_{11} - x$	$\bar{r}_{12} - \bar{c}_{11} + x$

K = 2	C1	C2
R1	$\bar{k}_{11} - x$	$\bar{r}_{21} + x - \bar{k}_{11}$
R2	$\bar{c}_{21} + x - \bar{k}_{11}$	$\bar{r}_{22} - \bar{c}_{21} - x + \bar{k}_{11}$

The value of x in the above table takes values between a lower and upper bound calculated from the below formulas.

The lower bound is computed by:

$$L = \max(0, \bar{c}_{11} - \bar{r}_{12}, \bar{c}_{11} - \bar{c}_{21} - \bar{k}_{11}).$$

The upper bound is calculated by:

$$U = \min(\bar{r}_{11}, \bar{c}_{11}, \bar{k}_{11}).$$

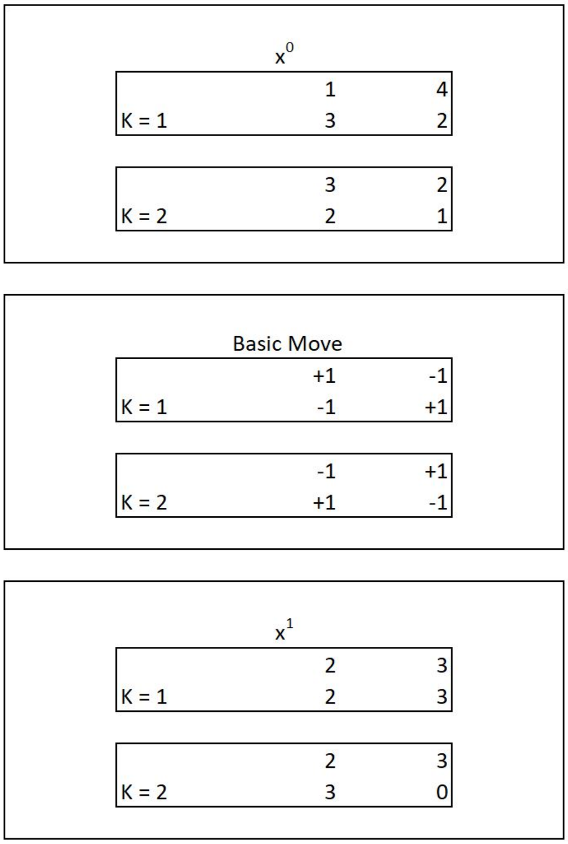
In our method, the hypergeometric probabilities are then calculated with the following formula:

$$P(X = x) = \frac{\left(\prod_{l=1}^{2} \prod_{s=1}^{2} \bar{r}_{ls}!\bar{c}_{ls}!\bar{k}_{ls}!\right)}{(n!)^2 \zeta_{111}!\zeta_{121}!\zeta_{211}!\zeta_{221}!\zeta_{112}!\zeta_{122}!\zeta_{212}!\zeta_{222}!}.$$

2.2.10 Markov Chain Monte Carlo

The Monte Carlo Method simulation technique utilizes repeated random sampling in order to find an approximate solution to a numerical problem which would be difficult to solve by other methods [7]. A Markov Chain is a sequence of random variables with the same conditional state space which satisfy the Markov property. The combination of these tools is the MCMC method. We use the MCMC method as described by [10]. This method is a way to traverse the conditional state space while conditioning upon fixed marginal sums from $\mathbf{x^0}$ [10].

This method utilizes a series of moves in a Markov basis, described in the following subsection, in order to traverse the conditional state space. The MCMC method utilizes a set of 'basic moves' in order to traverse the conditional state space. This concept is best described by the example in Figure 2.3. In Figure 2.3, $\mathbf{x^0}$ is our observed data set. The basic move is the process of adding and subtracting 1 to cells in $\mathbf{x^0}$ in a way the transitions us from $\mathbf{x^0}$ to $\mathbf{x^1}$ while maintaining the marginal sums. This procedure moves us to a new table in the conditional state space. This process is repeated until a large enough sample size has been achieved, taking into account the requirement for burn-in and thinning. In general, basic moves cannot guarantee that a chain is connected which can cause sampling bias [11]. An example of a table where basic move cannot traverse the conditional state space is provided in Figure 2.4.



x^0

	1	4
K = 1	3	2

	3	2
K = 2	2	1

Basic Move

	+1	-1
K = 1	-1	+1

	-1	+1
K = 2	+1	-1

x^1

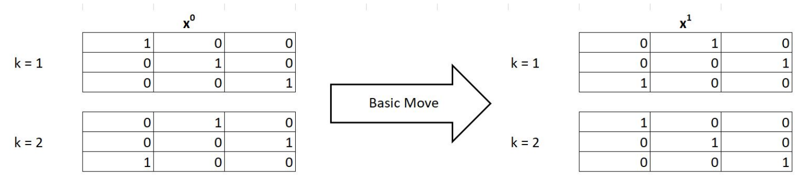
	2	3
K = 1	2	3

	2	3
K = 2	3	0

Figure 2.3. Example of Basic Move

2.2.11 Markov Basis

A drawback of a purely MCMC sampling method is the computation of a Markov Basis. A Markov Basis is the set of all moves such that any table in the conditional state space are connected to any other table via a finite number of moves without sampling from outside of the conditional state space [7]. Without computing a Markov Basis, one cannot know if all tables are connected which can possibly introduce sampling bias. Figure 2.4 is an example of a $3 \times 3 \times 2$ table that is not connected by a set of basic moves and therefore MCMC alone would be unable to sample from this state space.



x^0, k = 1

1	0	0
0	1	0
0	0	1

x^0, k = 2

0	1	0
0	0	1
1	0	0

x^1, k = 1

0	1	0
0	0	1
1	0	0

x^1, k = 2

1	0	0
0	1	0
0	0	1

Figure 2.4. Non-Ergodic State Space

As can be seen from Figure 2.4, a basic move cannot connect $\mathbf{x^0}$ to $\mathbf{x^1}$. This issue is solved by utilizing SIS as the sampling method.

De Loera and Onn, [11], showed that for tables $I \times J \times K$, in general, there are unbounded many moves in a Markov Basis for the no-three-way interaction model. By conducting hybrid sampling with the SIS and MCMC we bypass the need for computing a Markov Basis as the SIS samples depend on the marginal sums and not the previous tables values.

2.2.12 Metropolis Algorithm

We utilize the Metropolis step described in [10], this additional step guarantees that the Markov Chain is aperiodic and reversible. These properties are important because a Markov Chain that is aperiodic does not have cycles in the chain and reversibility means the Markov Chain has the same probability mass function both forwards and backwards. When these three properties exists, we are sampling from the hypergeometric distribution [7].

We calculate a statistic, a log of ratios, g, defined as:

$$g = \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \log \mathbf{x^n_{ijk}}! - \log \mathbf{x^{n+1}_{ijk}}!.$$

Where $\mathbf{x^n}$ is the current table in the MCMC chain and $\mathbf{x^{n+1}}$ is the proposed move. Once g is calculated, a random uniform $[0, 1]$, U, is generated. If $\min(e^g, 1) \leq U$ the move is accepted and the move occurs. If $\min(e^g, 1) > U$ the move is not accepted and we remain at table $\mathbf{x^n}$.

2.2.13 Kolmogorov-Smirnov Two-Sample Test

We utilize the Kolmogorv-Smirnov Two Sample test to validate whether our estimated null distribution and the asymptotic distribution are in fact equal. In the test, the null hypothesis is:

$$H_0 : F_Y(x) = F_X(x) \; \forall \; x$$

If the null hypothesis is true, we are unable to conclusively state the the distributions are from different populations [12].

2.3 Literature Review

SIS and MCMC methods for contingency tables have gained increasing attention in recent years. The research in this thesis utilizes findings from the following papers. A full review of these works is beyond the scope of this thesis and thus we refer readers to the original articles.

Chen, Diaconis, Holmes, Liu [5] describe methods for efficient SIS on two way 0-1 contingency tables and tables without a 0-1 constraint. The authors describe their method of SIS and its usage in determining the total number of tables in the conditional state space with given marginal sums by sampling from the uniform distribution.

The authors describe a SIS method for constructing contingency tables based on the marginal sums by recursively sampling each cell utilizing the uniform distribution. Their algorithm is described in Algorithm 2.3.1.

Algorithm 2.3.1 *(SIS for two way tables)*

- *Input the number of rows I, the number of columns J, the observed table* $\mathbf{x^0}$.
- *Output A table* $\mathbf{x}$ *with the same row and column sums with* $\mathbf{x^0}$ *sampled via SIS. Algorithm*

```
1. Compute the row sums $X_{i+}$ for $i = 1, \ldots, I$ and column sums $X_{+j}$ for
   $j = 1, \ldots J$.
2. For $i = 1, \ldots, (I-1)$ do:
   2.1. For $j = 1, \ldots, (J-1)$ do:
        i. Pick an integer $x$ uniformly from $[0, \min\{X_{i+} - (\sum_{k=1}^{j-1} X_{ik}), X_{+j} - (\sum_{k=1}^{i-1} X_{kj})\}]$, where we define $\sum_{k=1}^{0} X_{ik} = \sum_{k=1}^{0} X_{kj} = 0$.
3. For $i = 1, \ldots, I$ do:
   3.1. Set $X_{iJ} = X_{i+} - \sum_{k=1}^{J-1} X_{ik}$.
4. For $j = 1, \ldots, J$ do:
   4.1. Set $X_{Ij} = X_{+j} - \sum_{k=1}^{I-1} X_{kj}$.
5. Return $\mathbf{x}$.
```

Chen, Dinwoodie, and Sullivant [8] generalized the SIS procedures for all models in the discrete exponential family. They consider a system of linear equations and inequalities to define the sufficient statistics under a log-linear model. The sampling scheme they developed utilizes linear integer programming to recursively sample from the uniform distribution. Currently the SIS procedure uses uniform conditional distributions because the proposed distribution from the SIS procedure is close to the uniform distribution. Sampling from the hypergeometric distribution currently does not perform well for sparse tables.

Diaconis and Sturmfels [10] developed a MCMC approach to the goodness of fit test on log-linear models using the notion of *Markov bases*. The algorithms allows for sampling from the conditional distribution, given the marginals of a contingency tables, for discrete exponential families. They describe the implementation of the MCMC algorithm with a Markov basis to ensure an irreducible, aperiodic, and reversible Markov chain. See [10] for details.

Kahle, Yoshida, and Garcia-Puente [4] consider hybrid schemes to conduct exact conditional inference in discrete exponential families. The authors describe various methods of both MCMC and SIS algorithms as well as the pros and cons of each individual method. The MCMC method faces two major problems, (1) converging to the hypergeometric distribution on the conditional state space, called mixing, and (2) the efficiency of the chain. These problems create samples that are identically distributed but because the moves are Markovian, the *n*th sampled table, t_n, is dependent on the t_{n-1},th sampled table which creates the requirement for burn-in and thinning. In general it is unknown how many samples are required to thin in order to create iid samples. Benefits of the MCMC method are the speed of computation and once convergence on the conditional state space has occurred, the algorithm generates samples from the hypergeometric distribution allowing for exact inference. Two of the issues facing the SIS method is the ability to use the hypergeometric distribution as the conditional distribution and because of the recursive nature of the method, is is computationally expensive. The advantages of the SIS method is that it samples independent identically distributed tables by construction and does not require thinning and burn-in or any precomputation such as the construction of a Markov basis. The authors propose and explain the costs and benefits of multiple hybrid schemes that combine the SIS and MCMC method. In our paper we utilize one such method, SIS Initializations, in an attempt to leverage the iid sampling of the SIS method, with the convergence via the MCMC method.

CHAPTER 3:
Algorithm Description

In this chapter we describe our algorithm.

3.1 Algorithms

For describing our algorithm, the observed data set is defined as $\mathbf{x^0}$ and the output of the algorithm is defined as $\mathbf{x^1}$. The implementation of this algorithm in R takes marginal sums computed from the observed data $\mathbf{x^0}$ as its input. Our marginal sums are stored as matrices and manipulated through the execution of the algorithm. The matrices for the marginal sums are defined as follows.

The Row sums matrix, $\mathbf{r}$ is a $J \times K$ matrix where

$$r_{jk} := \sum_{j=1}^{J} X^0_{ijk}, \text{ for } i = 1, \ldots I \text{ and } k = 1, \ldots K.$$

The Column sums matrix, $\mathbf{c}$ is a $I \times K$ matrix where

$$c_{ik} := \sum_{i=1}^{I} X^0_{ijk}, \text{ for } j = 1, \ldots J \text{ and } k = 1, \ldots K.$$

The K-way sums matrix, $\mathbf{k}$ is a $J \times I$ matrix where

$$k_{ji} := \sum_{k=1}^{K} X^0_{ijk}, \text{ for } j = 1, \ldots J \text{ and } i = 1, \ldots I.$$

For accounting purposes during the construction of $\mathbf{x^1}$ we conduct row, column, and k-directional swaps of $\mathbf{x^1}$ in order to only operate on X^1_{111}. Because we only operate on X^1_{111}, we also only utilize cell count on $(1, 1)$ of $\mathbf{r}$, $\mathbf{c}$, $\mathbf{k}$. The swap is conducted on $\mathbf{x^1}$, $\mathbf{r}$, $\mathbf{c}$, and $\mathbf{k}$. Example 3.1.1 demonstrates this procedure.

Example 3.1.1 *To describe the method we will use a numeric example on r, which for this example will be a* 3×3 *Matrix.*

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}.$$

The column swap occurs after X^1_{111} *has been sampled and the marginal sums have been updated. After the swap r is displayed below.*

$$\begin{bmatrix} 2 & 3 & 1 \\ 5 & 6 & 4 \\ 8 & 9 & 7 \end{bmatrix}.$$

Algorithm 3.1.2 *Swap of Columns*

- *Input:* $\mathbf{x^1}$, $\mathbf{r}$, $\mathbf{c}$, $\mathbf{k}$.
- *Output: Updated* $\mathbf{x^1}$, $\mathbf{r}$, $\mathbf{c}$, $\mathbf{k}$.

Algorithm

```
1. Set $X^1{}_{1,(1,\ldots,J),1} = X^1{}_{1,(2,\ldots,J,1),1}$.
2. Set $r_{1,(1,\ldots,J)} = r_{1,(2,\ldots,J,1)}$.
3. Set $c_{1,(1,\ldots,I)} = c_{1,(2,\ldots,I,1)}$.
4. Set $k_{(1,\ldots,J),1} = k_{(2,\ldots,J,1),1}$.
```

Algorithm 3.1.2 does not only occur on the columns but the rows and layers as well. In our algorithm rows, columns, or layers are swapped based on which $\mathbf{x^1}$ cell is being operated on. It is easy to work through which row, column, or pillar of $\mathbf{x^1}$, $\mathbf{r}$, $\mathbf{c}$, or $\mathbf{k}$ is to be swapped.

Our SIS algorithm under the no-three-way interaction model is completed by utilizing a compilation of several smaller algorithms, the largest of which is the SIS step, which generates a new table.

3.1.1 $\mathbf{X^1} < 0$ Correction Algorithms

Algorithms 3.1.3, 3.1.4, and 3.1.5 described below, are utilized to prevent unnecessary rejections of sampled tables. During the SIS procedure, it is possible that the hypergeometric sampling creates new tables with different marginal sums than $\mathbf{x^0}$. This table cannot be utilized for calculating a p-value because the sufficient statistics are different.

We are interested in reducing the rejection rate of the SIS method because our algorithm utilizes a 'While' loop to achieve the desired quantity of SIS samples. For this reason, high rejection rates leads to longer computational time.

If we were sampling from $I \times J$ tables, none of these additional algorithms would be required. It is the addition of the K direction to our arrays that the correction algorithms become a necessity. The algorithms are described below and simple examples of the algorithms are shown in Appendix A.

Algorithm 3.1.3 is used to reduce the rejection rate by utilizing computed cell counts in $\mathbf{x^1}$ to correct a potential negative value from being forced into the table during the final step. This algorithm runs when $j = J$ in the SIS procedure.

Algorithm 3.1.3 *$j = J$ Corrective Action*

- *Input:* $\mathbf{x^1}$.
- *Output: Corrected* $\mathbf{x^1}$.

Algorithm

```
1. If $k_{11}$ or $c_{11} < r_{11}$,
    1.1. Then $X^1_{111} = \min(k_{11}, c_{11})$.
    1.2. Update $k_{11}$, $c_{11}$, $r_{11}$.
    1.3. Locate $k_{\cdot 1}$ or $c_{1\cdot} \geq r_{11} = w$.
    1.4. Set $X^1_{1w1} = X^1_{1w1} + r_{11}$.
    1.5. Set $c_{1w} = c_{1w} - r_{11}$.
    1.6. Set $k_{w1} = k_{w1} - r_{11}$.
    1.7. Set $r_{11} = 0$.
2. Else $X^1_{111} = r_{11}$.
    2.1. Set $k_{11} = k_{11} - r_{11}$.
    2.2. Set $c_{11} = c_{11} - r_{11}$.
    2.3. Set $r_{11} = 0$.
```

Algorithm 3.1.4 is used in the same way as 3.1.3. The algorithm runs when $i = I$ during the SIS procedure. An example of this procedure is given in Appendix A.

Algorithm 3.1.4 *$i = I$ Corrective Action*

- *Input:* $\mathbf{x^1}$.
- *Output: Corrected* $\mathbf{x^1}$.

Algorithm

```
1. If $k_{11} < c_{11}$,
    1.1. Then $X^1_{111} = k_{11}$.
    1.2. Set $r_{11} = r_{11} - k_{11}$; $k_{11} = 0$.
    1.3. Locate $b$ in $X^1_{\cdot 21} = c_{11}$.
            i. If $b = \emptyset$ Reject $X^1$.
    1.4. Set $X^1_{b11} = X^1_{b21} + X^1_{b11}$.
    1.5. Set $k_{1b} = k_{1b} + X^1_{b21}$.
    1.6. Set $r_{11} = r_{11} - X^1_{b21}$.
    1.7. Set $r_{12} = r_{12} + X^1_{b21}$.
    1.8. Set $X^1_{b21} = 0$.
2. Else $X^1_{111} = c_{11}$.
    2.1. Set $k_{11} = k_{11} - c_{11}$.
    2.2. Set $r_{11} = r_{11} - c_{11}$.
    2.3. Set $c_{11} = 0$.
```

Algorithm 3.1.5 is used to reduce rejection rates by distributing potential negative value into neighboring cells. There is an example of 3.1.5 in Appendix A. This algorithm is used to eliminate the negative value and provide an acceptable table, i.e. $\mathbf{x^1} \geq 0$. However, it has the potential to move the negative cell count to a different place in $\mathbf{x^1}$, leading to a rejection.

Algorithm 3.1.5 *$i = I$ and $j = J$ corrective action*

- *Input:* $\mathbf{x^1}$.
- *Output: Corrected* $\mathbf{x^1}$.

Algorithm

```
1. If $X^1_{ij1} < 0$,
    1.1. Set $X^1_{ij1} = X^1_{ij1} - X^1_{111}$.
    1.2. Set $X^1_{i11} = X^1_{i11} - X^1_{111}$.
    1.3. Set $X^1_{1j1} = X^1_{1j1} - X^1_{111}$.
    1.4. Set $X^1_{i11} = 0$.
    1.5. Update $\mathbf{r}, \mathbf{c}, \mathbf{k}$.
    1.6. If any $\mathbf{x^1} < 0$, reject $X^1$.
2. Else $X^1_{ij1} = c_{11}$.
    2.1. Set $c_{11} = 0$.
    2.2. Set $r_{11} = 0$.
    2.3. Set $k_{11} = k_{11} - c_{11}$.
```

3.1.2 SIS Algorithms

Algorithm 3.1.6 is our SIS method.

Algorithm 3.1.6 *SIS Algorithm*

- *Input:* $I, J, K, \mathbf{r}, \mathbf{c}, \mathbf{k}$.
- *Output:* $\mathbf{x^1}$.

Algorithm

```
1. For $k = 1, \ldots, K - 1$,
    1.1. For $i = 1, \ldots, I$,
            i. For $j = 1, \ldots J$,
                A. While $j \neq J$ and $i \neq I$,
                    • Algorithm 2.2.3.
                    • $x^1_{111}$ = Hypergeometric Sample Result.
                    • Algorithm 3.1.2.
                B. If $j = J$,
                    • Algorithm 3.1.3.
                    • Algorithm 3.1.2.
                C. If $i = I$,
                    • Algorithm 3.1.4.
                    • Algorithm 3.1.2.
                D. If $i = I$ and $j = J$,
                    • Algorithm 3.1.5.
                    • Algorithm 3.1.2.
    2. When $k = K$,
        2.1. $x^1_{ijK} = \mathbf{k}$.
```

Return $\mathbf{x^1}$.

Algorithm 3.1.7 is our MCMC algorithm. To calculate a p-value on a given data set, Algorithm 3.1.7 is executed a given number of times, *n*, for each SIS sample constructed. For our simulation we generate 10,000 MCMC samples per each SIS table. We set the burn-in value at 25% and thinning at 25 tables.

Algorithm 3.1.7 *MCMC Algorithm*

- *Input:* $\mathbf{x^1}$.
- *Output:* $\mathbf{x^2}$ *via MCMC.*

Algorithm

```
1. For $n = 1, \ldots N$
    1.1. Draw 2 Random Rows, 2 Random Columns, and 2 Random layers.
    1.2. $\mathbf{x^2}$ = Execute "Basic Move" on $\mathbf{x^1}$.
    1.3. $\mathbf{x^1} = \mathbf{x^2}$.
    1.4. Next n.
```

Algorithm 3.1.8 is the full algorithm which takes an initial data set $\mathbf{x^0}$ and estimates a p-value for the no-three-way interaction model.

Algorithm 3.1.8 *P-Value Algorithm ; n is our required SIS sample size and m is our required MCMC sample size*

- *Input:* $\mathbf{x^0}$, *n, m.*
- *Output: Estimated p-value.*

Algorithm

```
1. Collect Dimensional data of $\mathbf{x^0}$.
2. Calculate $\mathbf{r}$, $\mathbf{c}$, and $\mathbf{k}$ of $\mathbf{x^0}$.
3. Calculate the MLE of $\mathbf{x^0}$.
4. Calculate the initial $\chi^2$ of $\mathbf{x^0}$ and MLE.
5. While $t < n$,
    5.1. Calculate $\mathbf{x^1}$ using Algorithm 3.1.6.
    5.2. Accept or Reject $\mathbf{x^1}$.
    5.3. Calculate $\chi^2$ of $\mathbf{x^1}$ and the MLE.
        • For $m = 1 \ldots M$
            – $\mathbf{x^2}$ = Execution of 3.1.7 on $\mathbf{x^1}$
            – Metropolis Score calculated and acceptance determined
            – Calculate $\chi^2$ of $\mathbf{x^2}$ and the MLE.
            – next m.
        • Burn-in and Thinning of MCMC $\chi^2$ values
    5.4. Next t.
6. Calculate p-value by:
        $\frac{\sum_{t=1}^{n} \mathcal{I}_{\chi^2(\mathbf{x^0}) \geq \chi^2(\mathbf{x^i})}}{n}$, where $\mathcal{I}$ is the indicator function.
7. Return p-value.
```

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Data Sets, Analysis, and Results

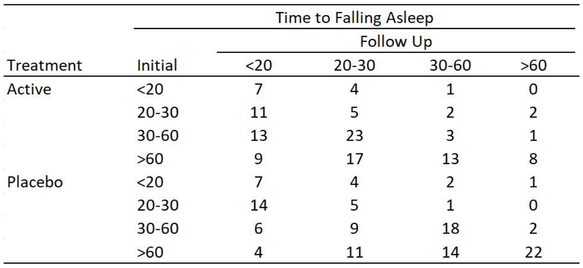
In the chapter we describe our data sets and examine the results of our simulations.

4.1 Data Sets

We utilize the following simulated data sets:

1. 3×3×3 Data Table, All Marginals = 10.
2. 4×4×4 Data Table, All Marginals = 4.
3. 5×5×5 Data Table, All Marginals = 5.
4. Randomly Generated Dense 5×4×2 table.
5. Randomly Generated Sparse 5×4×2 Table.
6. Sleep Data from [2], 3×4×2 Data Table.

Data sets 1, 2, and 3 are non random data sets. These data sets are only used to test the performance of the SIS algorithm not to conduct goodness of fit testing. Data set 4 is a dense table. This table satisfies the asymptotic distribution assumptions. For this reason, we expect the estimated null distribution to match the asymptotic distribution. Data set 5 is a sparse data set that does not match the asymptotic distribution assumptions. We expect the null distribution from data set 5 to be different than the asymptotic distribution. Data set 6 is Time to Falling Asleep, by Treatment and Occasion from [2], which is also a sparse table. This data set is shows in Figure 4.1.



	Time to Falling Asleep				
		Follow Up			
Treatment	Initial	<20	20-30	30-60	>60
Active	<20	7	4	1	0
	20-30	11	5	2	2
	30-60	13	23	3	1
	>60	9	17	13	8
Placebo	<20	7	4	2	1
	20-30	14	5	1	0
	30-60	6	9	18	2
	>60	4	11	14	22

Figure 4.1. Time to Falling Asleep, by Treatment and Occasion

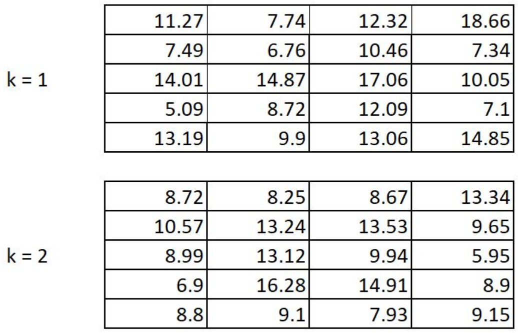
4.2 Equal Marginals Tables

The first tables were utilized simply to test our SIS algorithms performance. The following table summarizes computational time and rejection rates for the $3 \times 3 \times 3$, $4 \times 4 \times 4$, and $5 \times 5 \times 5$. We created 100 tables with the SIS algorithm and determine the rejection rates. The results can be seen in the below table.

Array	Computation Time	Rejection Rate
$3 \times 3 \times 3$	14.27 Seconds	2%
$4 \times 4 \times 4$	28.63 Seconds	22%
$5 \times 5 \times 5$	61.76 Seconds	45%

4.3 Dense $5 \times 4 \times 2$ Table

This data set meets the requirements of the asymptotic distribution. Therefore, if our sampler works, our estimated null distribution will match the χ^2 distribution with 12 degrees of freedom. The MLE for this table is shown in Figure 4.2.



	11.27	7.74	12.32	18.66
	7.49	6.76	10.46	7.34
k = 1	14.01	14.87	17.06	10.05
	5.09	8.72	12.09	7.1
	13.19	9.9	13.06	14.85

	8.72	8.25	8.67	13.34
	10.57	13.24	13.53	9.65
k = 2	8.99	13.12	9.94	5.95
	6.9	16.28	14.91	8.9
	8.8	9.1	7.93	9.15

Figure 4.2. MLE of Dense $5 \times 4 \times 2$ Table

Figure 4.3 is the result of executing 1,000 SIS samples with 10,000 MCMC tables for each SIS table.

The distribution of test statistics seems to match the χ^2 distribution. Further, we conducted a Kolmogorov-Smirnov test for fit between our sampled tables and the χ^2_{12} distribution and failed to reject H_0. This result demonstrates that our hybrid sampling scheme is accurate. Further, since the data does satisfy the χ^2 assumptions the p-value calculated from a χ^2

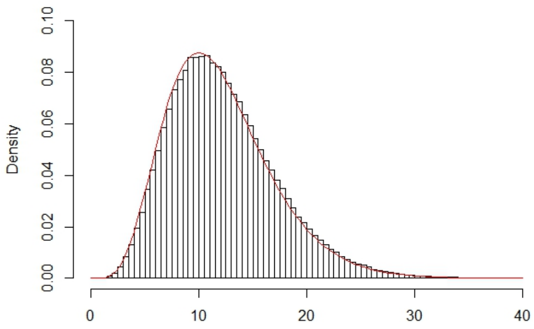
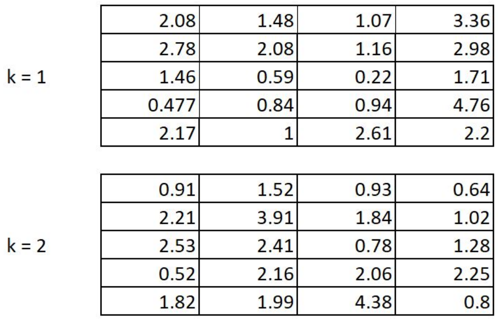


Figure 4.3. Results of Dense $5 \times 4 \times 2$ Table with Hybrid Sampling

table should match the p-value estimated by our algorithm. The χ^2 p-value is .549 and the p-value calculated by our algorithm is .522.

4.4 Sparse $5 \times 4 \times 2$ Table

This data set does not meet the requirements of the asymptotic distribution. As can be seen from the MLE of this table in Figure 4.4, none of the cells in the MLE are greater than five.



	2.08	1.48	1.07	3.36
	2.78	2.08	1.16	2.98
k = 1	1.46	0.59	0.22	1.71
	0.477	0.84	0.94	4.76
	2.17	1	2.61	2.2
	0.91	1.52	0.93	0.64
	2.21	3.91	1.84	1.02
k = 2	2.53	2.41	0.78	1.28
	0.52	2.16	2.06	2.25
	1.82	1.99	4.38	0.8

Figure 4.4. MLE of Sparse $5 \times 4 \times 2$ Table

The histogram of 1,000 SIS tables with 10,000 MCMC samples are found in Figure 4.5. As can be seen, the distribution does not match the χ^2_{12}, which can lead to incorrect conclusions when conducting goodness-of-fit testing. However, determining a p-value utilizing our methodology prevents these errors.

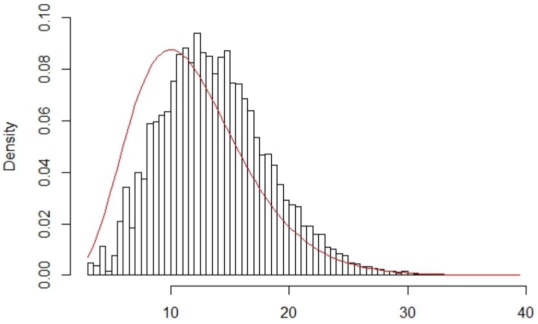


Figure 4.5. Results of Sparse 5 × 4 × 2 Table with Hybrid Sampling

As can be seen, the distribution does not appear to match the χ^2_{12}, which we expected. After conducting Kolmogorov-Smirnov test we reject H_0 and have determined the two distributions are not the same. This result shows us that simply ignoring the violation of cell count requirements from the asymptotic distribution can lead to incorrect conclusions and model selection.

4.5 Time to Falling Asleep by Treatment and Occasion

Figure 4.6 is the result of the simulation of 1,000 SIS and 10,000 MCMC samples on the Time to Falling Asleep by Treatment and Occasion data set from [2]. As you can see from the histogram, the data does not follow a χ^2_6 distribution; it is skewed to the left. A Kolmogorov-Smirnov test conducted on the estimated null distribution and a χ^2_6 led to a rejection of H_0. The p-value calculated when utilizing a χ^2_6 is .002 which rejects the null hypothesis that $H_0 : \lambda^{XYZ}_{ijk} = 0$. The p-value we calculate is .035, which fails to reject at the $\alpha = .01$ level.

These results show the value of our new methodology. The potential for incorrect model fitting exists when using the asymptotic distribution on sparse $I \times J \times K$ contingency tables. This error is not present when utilizing our SIS MCMC hybrid sampling scheme.

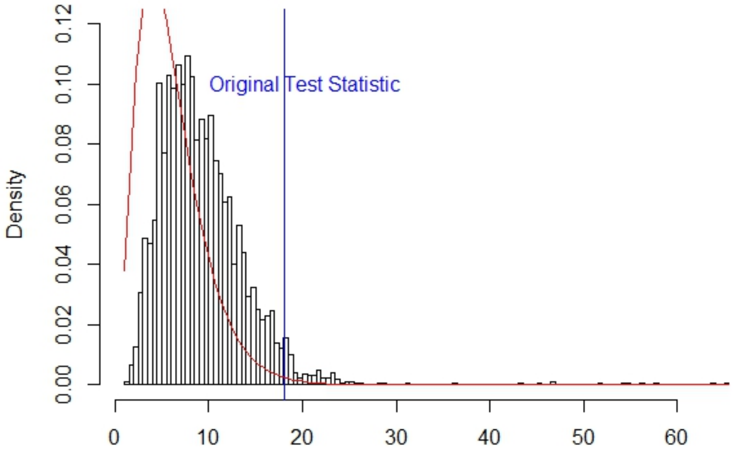


Figure 4.6. Results of Sleep Data

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5: Conclusion

In this chapter we summarize our main results and provide recommendations for follow on work.

5.1 Conclusions

Contingency tables will continue to be a widely used method for analyzing data sets. As large data sets become more and more common, sparse data sets will also be more common. As we have shown, the χ^2 distribution is not a good approximation for the null distribution of test statistics for sparse data sets. Therefore, using the asymptotic distribution, i.e., the χ^2 distribution for sparse $I \times J \times K$ contingency tables, can lead to incorrect conclusion when conducting goodness of fit testing.

Via simulation we demonstrated that our SIS/MCMC methodology accurately estimates the null distribution for dense $I \times J \times K$ contingency tables under the no-three-way interaction model. Therefore, we are confident that our method works accurately for estimating the null distribution for sparse $I \times J \times K$ contingency tables. As was demonstrated, our method provides a more accurate p-value estimate for a given data set than the result obtained by simply utilizing the χ^2 distribution and ignoring the MLE assumptions.

Finally, our SIS/MCMC hybrid scheme can be utilized on $I \times J \times K$ contingency tables that are sparse or dense from any academic field. Our new methodology is not limited to the conclusion we derived for Time to Falling Asleep by Treatment and Occasion from [2]. This new sampling scheme can be utilized on any $I \times J \times K$, for $I,\ J,\ K \geq 2$, data set in any field of research. It provides a goodness-of-fit test under the no-three-way interaction model regardless of the source of the data.

5.2 Follow-on Work

Although this method is successful, the SIS scheme has limitations in terms of its ability to handle high dimensional $I \times J \times K$ contingency tables. As the dimensional size of the table increases, rejection rates also increase, leading to increased computational time for a given simulation. Improvements in the SIS method would reduce this issue.

The SIS/MCMC has general applicability for $I \times J \times K$ contingency tables. For this reason, the method would benefit many different researchers if it was modified and developed into an R package.

APPENDIX: Algorithm Examples

A.1 Example 1

This example demonstrates Algorithm 3.1.5 from Chapter 3 using a $3 \times 3 \times 3$ table with all marginal sums, **r**, **c**, and **k** equal to 10. Algorithm 3.1.5 only occurs when i = I and j = J, the last cell to be sampled for each layer k ∈ K. For this example, we look at $k = 1$. We choose $k = 1$ to demonstrate the concept of the algorithm. Algorithm 3.1.5 operates the same regardless of which layer we are currently sampling for.

$$k = 1 \left| \begin{matrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{matrix} \right|$$

$$k = 2 \left| \begin{matrix} 0 & 10 & 0 \\ 0 & 0 & 10 \\ 10 & 0 & 0 \end{matrix} \right|$$

$$k = 3 \left| \begin{matrix} 0 & 0 & 10 \\ 10 & 0 & 0 \\ 0 & 10 & 0 \end{matrix} \right|$$

The below is the current value of $\mathbf{x^1}$. As you can see, the required value to complete this table is to place -1 into X^1_{331}. This would lead to a rejection. To complete the algorithm, we place -1 into X^1_{331} and can now execute Algorithm 3.1.5.

$$k = 1 \left| \begin{matrix} 3 & 2 & 5 \\ 3 & 1 & 6 \\ 4 & 7 & \enclose{circle}{-1} \end{matrix} \right|$$

The algorithm takes the -1 from X^1_{331} and subtracts it from X^1_{331} and X^1_{221}. It then adds the -1 to X^1_{321} and X^1_{231}. The values for k = 1 are now valid and sampling can continue for the remainder of $\mathbf{x^1}$.

$$k = 1 \begin{vmatrix} 3 & 2 & 5 \\ 3 & 2 & 5 \\ 4 & 6 & 0 \end{vmatrix}$$

A.2 Example 2

This example demonstrates Algorithm 3.1.4 from Chapter 3 using a 3 × 3 × 3 table with all marginal sums, **r**, **c**, and **k** equal to 10. This can be seen below.

$$k = 1 \begin{vmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{vmatrix}$$

$$k = 2 \begin{vmatrix} 0 & 10 & 0 \\ 0 & 0 & 10 \\ 10 & 0 & 0 \end{vmatrix}$$

$$k = 3 \begin{vmatrix} 0 & 0 & 10 \\ 10 & 0 & 0 \\ 0 & 10 & 0 \end{vmatrix}$$

Let's suppose we have already sampled k = 1 and are currently sampling cells in k = 2. For this example, we are currently on $\mathbf{x^1}_{\mathbf{322}}$, which is where our error could occur.

$$k = 1 \begin{vmatrix} 3 & 2 & 5 \\ 3 & 2 & 5 \\ 4 & 6 & 0 \end{vmatrix}$$

$$k = 2 \begin{vmatrix} 5 & 3 & 2 \\ 4 & 1 & 5 \\ 1 & \textcircled{0} & 0 \end{vmatrix}$$

$$k = 3 \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

The current values of **c**. The bold value is the current required value to make $\sum X^1_{+22} = 10$.

The current values of **k**. The bold value is the current required value to make $\sum X^1_{32+} = 10$.

$$\begin{vmatrix} 0 & 0 & 0 \\ 0 & \textcircled{6} & 3 \\ 10 & 10 & 10 \end{vmatrix}$$

If we place the value of 6, from **c**, into X^1_{322} then k_{23} will become -2. Forcing a negative value into the k = 3 layer of $\mathbf{x^1}$ causing a rejection.

$$\begin{vmatrix} 2 & 3 & 5 \\ 5 & 7 & \textcircled{4} \\ 3 & 0 & 10 \end{vmatrix}$$

We set X^1_{322} = 4, which is min(k_{32}, c_{22}). Then we set k_{32} = 0, C_{22} = 2. X^1 now holds these values. Currently, X^1_{+22} does not sum to 10. We now scan $X^1_{.32}$ for the leftover value of 2 from c_{22}. We find this in X^1_{132}. We add this value to X^1_{122} thus c_{22} = 0. All marginals are now updated and X^1_{132} is set = 0.

$$k = 1 \begin{vmatrix} 3 & 2 & 5 \\ 3 & 2 & 5 \\ 4 & 6 & 0 \end{vmatrix}$$

$$k = 2 \begin{vmatrix} 5 & 3 & 2 \\ 4 & 1 & 5 \\ 1 & \textcircled{4} & 0 \end{vmatrix}$$

$$k = 3 \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

After executing Algorithm 3.1.4 $\mathbf{x^1}$ where k = 2 satisfies all marginal sums.

$$k = 1 \begin{vmatrix} 3 & 2 & 5 \\ 3 & 2 & 5 \\ 4 & 6 & 0 \end{vmatrix}$$

$$k = 2 \begin{vmatrix} 5 & 5 & 0 \\ 4 & 1 & 5 \\ 1 & 4 & 0 \end{vmatrix}$$

$$k = 3 \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

Values of **c** after Algorithm 3.1.4.

$$\begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 10 & 10 & 10 \end{vmatrix}$$

Values of **k** after Algorithm 3.1.4.

$$\begin{vmatrix} 2 & 3 & 5 \\ 3 & 7 & 0 \\ 5 & 0 & 5 \end{vmatrix}$$

Below is the completed Table. With the inclusion of Algorithm 3.1.4 we were able to salvage a $\mathbf{x^1}$ that would have been rejected.

$$k = 1 \left| \begin{matrix} 3 & 2 & 5 \\ 3 & 2 & 5 \\ 4 & 6 & 0 \end{matrix} \right|$$

$$k = 2 \left| \begin{matrix} 5 & 5 & 0 \\ 4 & 1 & 5 \\ 1 & 4 & 5 \end{matrix} \right|$$

$$k = 3 \left| \begin{matrix} 2 & 3 & 5 \\ 3 & 7 & 0 \\ 5 & 0 & 5 \end{matrix} \right|$$

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] J. Devore, *Probability and Statistics for Engineering and the Sciences*, 9th ed. Pacific Grove, California: Brooks Cole Publishing Co., 2015.

[2] A. Agresti, *Categorical Data Analysis*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2002, vol. 2.

[3] P. Diaconis and B. Efron, "Testing for independence in a two-way table: New interpretations of the chi-square statistic," *In The Annals of Statistics*, vol. 13, no. 3, pp. 845–874, 1985.

[4] D. Kahle, R. Yoshida, and L. Garcia-Puente, "Hybrid schemes for exact conditional inference in discrete exponential families," *Annals of the Institute od Statistical Mathematics*, 2017. Available: https://doi.org/10.1007/s10463-017-0615-z

[5] Y. Chen, P. Diaconis, S. Holmes, and C. Liu, "Sequential monte carlo methods for statistical analysis of tables," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 109–120, 2005.

[6] E. Deming and F. Stephan, "On a least squares adjustment of a sampled frequency table when the expected marginal totals are known," *The Annals of Mathematical Statistics*, vol. 11, no. 4, pp. 427–444, 1940.

[7] S. Ross, *Introduction to Probability Models*. Waltham, Massachusetts: Academic Press., 2014, vol. 11.

[8] Y. Chen, I. Dinwoodie, and S. Sullivant, "Sequential importance sampling for multiway tables," *Annals of Statistics*, vol. 34, no. 1, pp. 523–545, 2006.

[9] J. K. Lenstra and A. H. G. Rinnooy Kan, "Complexity of scheduling under precedence constraints," *Operations Research*, vol. 26, no. 1, pp. 22–35, 1978.

[10] P. Diaconis and B. Sturmfels, "Algebraic algorithms for sampling from conditional distributions," *The Annals of Statistics*, vol. 26, no. 1, pp. 363–397, 1998.

[11] J. De Leora and S. Onn, "Markov bases of three-way tables are arbitrarily complicated," *Journal of Symbolic Computation*, vol. 41, no. 2, pp. 173–181, 2006.

[12] J. Gibbons and C. S., *Nonparametric Statistical Inference*, 5th ed. Boca Rotan, Florida: CRC Press, 2011.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

2. Defense Technical Information Center
   Ft. Belvoir, Virginia