University of Kent

MA867: Project

Submitted in partial fulfillment of the requirements of the degree of
MSc Statistics, University of Kent, 2017

Bayesian Analysis of the Yule-Simon Distribution

Author:
Brandi Jess

Supervisor:
Dr. Fabrizio Leisen

August 25, 2017

Abstract

The Yule-Simon distribution has had little attention from the Bayesian perspective. In this paper, we will add to previous literature by adopting a Bayesian approach to the Yule-Simon distribution. This paper tests data-augmentation algorithms using simulated data and applies the Yule-Simon distribution to real data. We then extend the Yule-Simon to a two-parameter model, and test the proposed algorithm on simulated data.

Contents

Chapter 1

Introduction

The Yule-Simon distribution was first proposed by George Udny Yule in 1925. Yule [9] used the distribution to model the abundance of biological genera. Genera refers to the classification of animals, or more specifically, the classification of species of animals into groups. Species is the most descriptive classification for animals, whereas genera is the next most descriptive level of classification. Yule observed that monotypic genera are by far the most common, ditypic genera are less frequently seen, tritypic genera even less frequent, and so on. A monotypic genera would be the case where a species and genera are described simultaneously, ditypic genera would be the case where two species are described by the genera, and so forth. Yule then gives a very technical derivation of what we know of today as the Yule-Simon distribution.

In 1955, Herbert A. Simon [7] extended the applications of this distribution to model phenomena in the sociological and economical fields, as well as expanding on the applications in the biological field. Simon found it fitting to call this distribution the Yule distribution, but later it went on to become known as the Yule-Simon distribution. Simon observed that the negative-binomial and Fisher's logarithmic series are similar to the Yule-Simon distribution, but do not achieve all three of the following characteristics found in the data:

1. The distribution is J-shaped. The Yule-Simon distribution is known for being heavily skewed with a heavy upper tail, which can be estimated by

   $$f(i) = (a/i^k)b^i,$$

   where a and k are constants, and b is a constant close to one,

2. k is greater than one. In some cases, it is close to 2, and

3. $f(i)$ even describes the distribution for small values of i.

All three of these characteristics cannot be met by the negative-binomial or Fisher's logarithmic series, however, they can be met by the Yule-Simon distribution. Simon focused his paper on the application of the Yule-Simon distribution to model word frequencies in texts, specifically looking at James Joyce's *Ulysses* and the Eldrigde counts. Simon noted that there were two main assumptions for this model, written in terms of word frequencies:

1. the probability that the $k+1$ word has already appeared i times is proportional to $f(i,k)$, and

2. there is a constant probability that the $k+1$ word has not been seen before.

In addition to Simon's application to word frequencies, Simon also applied the Yule-Simon distribution to authors by the number of scientific articles published, cities by population size, and incomes by size, as well as reconsidered the application proposed by Yule to biological species.

In a text by Handcock et al. [3], it is noted that the Yule-Simon distribution is a specific case of the Waring distribution. The Waring distribution is a shifted form of the Beta-geometric distribution. The Waring distribution has the following probability mass function [4]:

$$f(x,p,a) = \frac{(p-a)(a+x-1)!p!}{p(a-1)!(p+x)!}, \qquad x = 0,1,2,...$$

where a and p are positive integers and p is greater than a. The Yule-Simon distribution is the case of the Waring distribution where $a=1$, as shown below:

$$\begin{aligned} f(x,p) &= \frac{(p-1)(1+x-1)!p!}{p(1-1)!(p+x)} \\ &= \frac{(p-1)x!(p!)}{p(p+x)!}. \qquad\qquad x = 0,1,2,\dots \end{aligned}$$

However, the Yule-Simon distribution is usually defined for positive values of x, so we get the following shifted version:

$$f(x,p) = \frac{p(x-1)!(p!)}{(x+p)!}. \qquad x = 1,2,\dots$$

Note that in the form of the equation above, p is only defined for integers. However, in practice, p can take on any positive real number, as we will see later. Handcock et al. [3] went on to define an MCMC algorithm to test on simulated data for social network models. The literature went on to use the MCMC methods to find maximum likelihood estimation of the parameters and to apply the Yule-Simon distribution to the biological network of protein-protein interactions in cells.

Garcia [2] also proposed an algorithm to estimate the parameter of the Yule-Simon using maximum likelihood estimation. In order to test this algorithm, they use simulated data. More specifically, they use a modified Polya urn process. The Polya urn process they define is one in which they have some number of bins each with only one ball in them. There is the probability α that a new bin is created for a new ball that comes in, and the probability $1 - \alpha$ that the new ball is placed in a bin that already exists. Then the probability that a bin has exactly i balls would be the probability mass function of the Yule-Simon distribution:

$$\rho \frac{\Gamma(k)\Gamma(\rho + 1)}{\Gamma(k + \rho + 1)},$$

where $\rho = 1/(1 - \alpha)$. Upon testing this algorithm, they found the results were pretty accurate.

In a text by Gallardo et al. [1], the Yule-Simon distribution was applied to cure rate models for cancer patients with malignant melanomas. Cure rate models focus on survival data where a proportion of patients are cured. In particular, they used the Yule-Simon distribution to model the number of concurrent cases. In this case, they shifted the Yule-Simon distribution to include the origin. They used the idea that an excessive number of cells producing metastasis could explain the presence of many short survival times. Metastasis is the growth of secondary malignant cells that appears away from the primary site. Malignant cells are cells that divide without control and invade tissues. Gallardo et al. [1] also used an extension of the Yule-Simon distribution which they called the Weibull-Yule-Simon distribution.

Leisen et al. [6] aimed to fill a gap in the current literature by pursuing a Bayesian analysis of the Yule-Simon distribution. They propose two objective priors for the Yule-Simon distribution: a Jeffreys rule prior and a loss-based approach prior. In the paper, they derive the priors and test them on simulated data, as well as some real data applications. The applications to real data include analyzing social media stock indexes, more specifically, looking at Facebook, Twitter, LinkedIn, and Google, analyzing the frequencies of some of the most common surnames from the U.S. Census, and analyzing the number of number one hits by an artist on the Billboard Hot 100 chart from 1955-2003. The proposed Bayesian analysis in this paper provided estimates of the parameters that were close to the true parameters, suggesting this may be a good model for the Yule-Simon distribution.

The Yule-Simon distribution is a highly skewed, discrete distribution. This distribution is J-shaped and characteristically known for being heavy-tailed. It is also not uncommon to observe extreme values in the data, even when dealing with a small sample size. The Yule-Simon distribution is often used in cases where the data is in the form of frequencies or counts, as shown in the applications from current literature above.

As we can see from the current literature, there is a wide array of applications for the Yule-Simon distribution. However, new applications are still being found. In Chapter 2 of this paper, we will introduce the preliminary information for the Yule-Simon distribution and consider a Bayesian analysis approach to this model. We will then introduce algorithms for a simulation study and count data regression analysis, as well as apply the count data regression to real data examples. In Chapter 3, we will explore an algorithm for the two-parameter extension of the Yule-Simon distribution using simulated data. Lastly, Chapter 4 will cover points of discussions and conclusions.

Chapter 2

Bayesian Analysis of the Yule-Simon Distribution with Applications

2.1 The Yule-Simon Distribution

In Chapter 1, we mentioned that the Yule-Simon has the following form:

$$f(k;\rho) = \rho\frac{(k-1)!(\rho)!}{(k+\rho)!}$$

However, this restricts the values of ρ to be the positive integers, when we would like ρ to be able to take on any positive real number. Therefore, we use the following as the probability mass function of the Yule-Simon Distribution [4]:

$$f(k;\rho) = \rho\,\mathrm{B}(k,\rho+1), \qquad k = 1,2,\ldots \quad \text{and} \quad \rho > 0, \tag{2.1}$$

where B is the Beta function and ρ is the shape parameter. When working with factorials, we must use integers, so we write the Yule-Simon using the Beta function so that ρ is defined for any positive real number. This can alternatively be expressed as a mixture of geometric distributions, as follows:

$$\begin{aligned} K|W &\sim \mathrm{Geom}(e^{-W}) \\ W|\rho &\sim \mathrm{Exp}(\rho). \end{aligned} \tag{2.2}$$

Equation (2.2) is very useful for generating simulated data from the Yule-Simon distribution, as we will see in the following sections. From this, we get the following marginal distribution of the random vector (K, W):

$$f(k;\rho) = \int_0^\infty e^{-w}(1-e^{-w})^{k-1}\rho e^{-\rho w} dw. \tag{2.3}$$

The form in Equation (2.3) is especially useful for the data augmentation needed when we are working in the Bayesian setting. Data augmentation is the sampling algorithm used when we cannot directly compute our posterior distribution [8].

Next, we consider the following Bayesian model:

$$\begin{aligned} k_1, ..., k_n|\rho &\sim f(k;\rho) \\ \rho &\sim \text{Gamma}(a,b), \end{aligned} \tag{2.4}$$

where $f(k;\rho)$ is the function defined in Equation (2.1); the Yule-Simon distribution. From this, we get the following likelihood function:

$$\begin{aligned} L(k;\rho) &= \prod_{t=1}^{n} \int_0^\infty e^{-w_t}(1-e^{-w_t})^{k_t-1}\rho e^{-\rho w_t} dw_t \\ &= \int_{(0,\infty)^n} \prod_{t=1}^{n} e^{-w_t}(1-e^{-w_t})^{k_t-1}\rho e^{-\rho w_t} d\mathbf{w} \\ &= \int_{(0,\infty)^n} L(\mathbf{k},\mathbf{w},\rho) d\mathbf{w} \end{aligned} \tag{2.5}$$

where $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ is a vector of auxiliary variables and $\mathbf{k} = (k_1, k_2, \ldots, k_n)$ is a vector of observations.

Using a Gamma prior leads us to the following posterior distribution:

$$\pi(\rho, \mathbf{w}|\mathbf{k}) \propto L(\mathbf{k},\mathbf{w},\rho)\pi(\rho), \tag{2.6}$$

where $L(\mathbf{k},\mathbf{w},\rho) = \prod_{t=1}^{n} e^{-w_t}(1-e^{-w_t})^{k_t-1}\rho e^{-\rho w_t}$ and $\pi(\rho) \propto \rho^{a-1}e^{-b\rho}$.

We see that the posterior distribution is not an explicit known distribution, so we will use Monte Carlo methods to sample from the posterior distribution. Before we employ MCMC methods, we must derive the full conditional distributions for w_i and ρ. The full conditional for w_i is given as:

$$\pi(w_i|w_{-i}, \boldsymbol{k}, \rho) \propto e^{-\rho w_i}e^{-w_i}(1-e^{-w_i})^{k_i-1}.$$

Using the change in variable $t_i = e^{-w_i}$, it is easy to see that $\pi(w_i|w_{-i}, \boldsymbol{k}, \rho) \sim \text{Beta}(\rho + 1, k_i)$. We can then derive the full conditional distribution for parameter ρ, which is given by:

$$\pi(\rho|\boldsymbol{k}, \boldsymbol{w})) \propto \rho^{a+n-1}e^{-\rho(b+\sum_{i=1}^{n} w_i)} \sim \text{Gamma}\left(a+n, b+\sum_{i=1}^{n} w_i\right).$$

Because the distributions of the full conditional for both w_i and ρ are explicit, known distributions, we can use a Gibbs sampler to sample from the posterior distribution. We use the Gibbs sampler proposed by Leisen et al [5] for the data as follows:

- sample $t_i|\rho, k_i \sim \text{Beta}(\rho + 1, k_i)$, for $i = 1, \ldots, n$;
- compute $w_i = -\log(t_i)$, for $i = 1, \ldots, n$;
- sample $\rho|\mathbf{w}, \mathbf{k} \sim \text{Gamma}(a + n, b + \sum_{i=1}^{n} w_i)$.

In the following section, we will test the proposed algorithm using simulated data from the Yule-Simon distribution.

2.2 Simulation Study

In this section, we will test the MCMC methods proposed by Leisen et al [5]. In order to test the algorithm, we will use simulated data from the Yule-Simon distribution. We will test a variety of sizes of samples, including $n = 50$, $n = 100$, and $n = 500$. The chosen values of the parameter ρ are 0.7 and 4.

As seen previously, for the Bayesian analysis we will use a Gamma prior. For these simulations, we chose the shape parameter to be $a = 0.25$ and the rate parameter to be $b = 0.05$. We ran 50,000 iterations of the Gibbs sampler with a burn-in period of 10,000 iterations. This was then replicated 20 times per sample. Table 2.1 shows the mean, median, relative mean squared error for the mean, and relative mean squared error for the median of the posteriors. The relative mean squared error is calculated by $\sqrt{MSE(\rho)}/\rho$. We use the relative mean squared error to give us an idea of the precision of our estimates.

ρ	n	Mean	Median	MSE Mean	MSE Median
0.7	50	0.7293	0.7214	0.1770	0.1141
0.7	100	0.7866	0.7821	0.1806	0.1241
0.7	500	0.6830	0.6823	0.0557	0.0388
4	50	5.8142	5.4447	0.6818	0.3681
4	100	4.1563	4.0526	0.2235	0.1391
4	500	4.2534	4.2318	0.1189	0.0764

Table 2.1: Summary statistics of the posterior distribution for the parameter ρ using simulated data from the Yule-Simon distribution with varying values of the parameter, ρ and sample sizes, n.

Table 2.1 shows us that our posterior sample means and medians are very close to the true parameters for ρ with small mean squared error values. This suggests that the proposed algorithm is performing well.

In Figure 2.1 and Figure 2.2, we see the graphical results of the posterior from the simulation study. These plots show one simulation of the Yule-Simon distribution for parameter $\rho = 0.7$ with sample sizes $n = 50$ and $n = 500$.

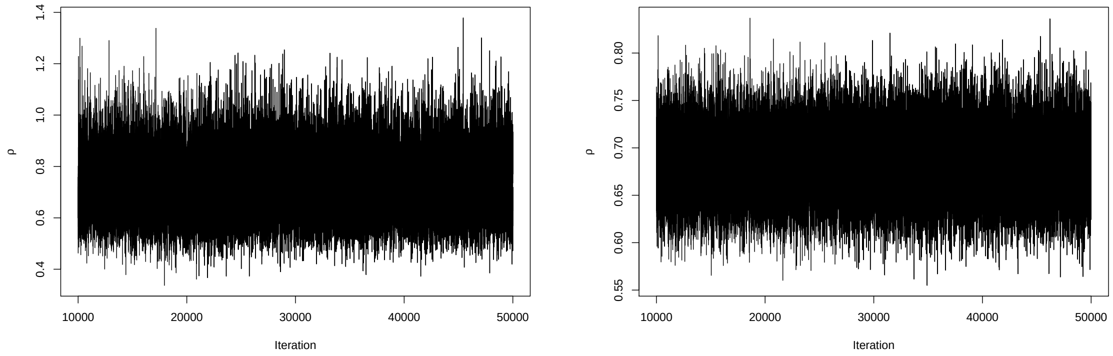

Figure 2.1: Posterior samples for simulated data from the Yule-Simon distribution with parameter $\rho = 0.7$ and sample size $n = 50$ (left) and $n = 500$ (right).

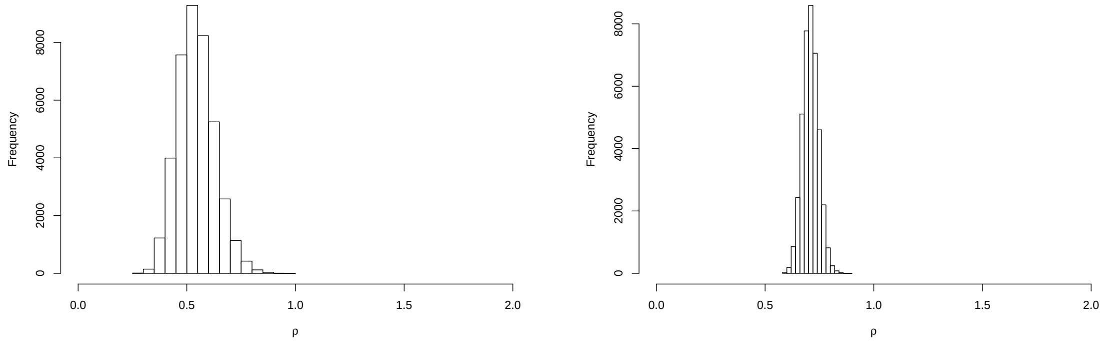

Figure 2.2: Posterior histograms for simulated data from the Yule-Simon distribution with parameter $\rho = 0.7$ and sample size $n = 50$ (left) and $n = 500$ (right).

From Figure 2.1, it appears as through the chains have good mixing and seem to be centered around the true parameters values for both sample sizes. The histograms in Figure 2.2 show that the variance is quite small for both $n = 50$ and $n = 500$. As expected, the variance is smaller for a larger sample size. The credible interval for $n = 50$ is $(0.6132, 0.6809)$ and for $n = 500$ is $(0.7493, 0.7563)$. The credible intervals correspond to that of the posterior histograms, with a larger interval for the smaller sample size. Thus, it seems as though the algorithm is accurate for both small and large sample sizes.

2.3 Count Data Regression

Count data regression refers to a data in which the response variable, say k_i, is in the form of counts. A count data regression model examines the relationship between a vector of independent variables, x_i, and the response, k_i. In order to use the count data regression model, we must make the following assumptions:

1. the data, k_i, follows a Yule-Simon distribution with parameter ρ_i,

$$f(k_i;\rho_i) = \rho_i \, \mathrm{B}(k_i, \rho_i + 1), \qquad k_i = 1, 2, \ldots, \rho_i > 0;$$

2. the parameter, ρ_i, is modeled by

$$\rho_i = \exp(\boldsymbol{x}_i'\boldsymbol{\beta}) \qquad i = 1, \ldots, n,$$

where $\boldsymbol{\beta}$ is a $(n_\beta \times 1)$ vector of parameters and $\boldsymbol{x}_i' = (1, x_{i2}, \ldots, x_{in_\beta})$ is a $(1 \times n_\beta)$ vector of regressors, including the constant;

3. the observation pairs (k_i, x_i) are independently distributed for $i = 1, \ldots, n$.

Here we will focus on the case where we have only one regressor. This gives us $\boldsymbol{\beta}' = (\beta_0, \beta_1)$ and $\boldsymbol{x}_i' = (1, x_{i2})$, which implies that $\rho_i = \exp\{\beta_0 + \beta_1 x_{i2}\}$. We assume a standard bivariate normal prior for β. We then arrive at the following augmented version of the posterior distribution:

$$\pi(\boldsymbol{\beta}, \boldsymbol{w}, \boldsymbol{x} | \boldsymbol{k}) \propto \left[\prod_{i=1}^{n} e^{-w_i}(1 - e^{-w_i})^{k_i - 1}\right] exp\left\{\sum_{i=1}^{n} \boldsymbol{x}_i'\boldsymbol{\beta}\right\} \left[\prod_{i=1}^{n} \exp\{-e^{\boldsymbol{x}_i'\boldsymbol{\beta}} w_i\}\right] e^{-(1/2)\boldsymbol{\beta}'\boldsymbol{\beta}}.$$

From the posterior we arrive at the full conditional distribution for parameter β which is given as:

$$\pi(\boldsymbol{\beta} | \boldsymbol{w}, \boldsymbol{x}, \boldsymbol{y}) \propto \left[\prod_{i=1}^{n} \exp\{-e^{\boldsymbol{x}_i'\boldsymbol{\beta}} w_i\}\right] \exp\left\{-(1/2)\boldsymbol{\beta}'\boldsymbol{\beta} + \sum_{i=1}^{n} \boldsymbol{x}_i'\boldsymbol{\beta}\right\}. \qquad (2.7)$$

As the full conditional in equation (2.7) is not an explicit, known distribution, we will use Monte Carlo methods to sample from the posterior distribution as proposed by Leisen et al. [6]. Here we will adopt a Metropolis within Gibbs sampling method to obtain samples from the posterior distribution. The Gibbs sampler and random walk Metropolis Hastings used for the count data regression are as follows:

- sample $t_i | \beta_0, \beta_1, x_i, k_i \sim \mathrm{Beta}(\exp\{\beta_0 + \beta_1 x_{i2}\} + 1, k_i)$, for $i = 1, \ldots, n$;

- compute $w_i = -\log t_i$, for $i = 1, \ldots, n$;

- sample $\boldsymbol{\beta}|\mathbf{w,k,x}$ from the random walk Metropolis-Hastings algorithm.

For the random walk Metropolis-Hastings algorithm, we use the following acceptance ratio:

$$\alpha = min\left\{1, \frac{\pi(\beta_{new})}{\pi(\beta_{old})}\right\}.$$

In this case, we will use the log of the acceptance ratio because the logs are easier to work with. Taking the log, we get the following for the ratio between the full conditional distributions for β_{new} and β_{old}:

$$\log\left(\frac{\pi(\beta_{new})}{\pi(\beta_{old})}\right) = -\sum_{i=1}^{n} e^{\boldsymbol{x}_i'\beta_{new}} w_i - \frac{1}{2}\beta_{new}'\beta_{new} + \sum_{i=1}^{n} \boldsymbol{x}_i'\beta_{new}$$
$$+ \sum_{i=1}^{n} e^{\boldsymbol{x}_i'\beta_{old}} w_i - \frac{1}{2}\beta_{old}'\beta_{old} + \sum_{i=1}^{n} \boldsymbol{x}_i'\beta_{old}$$

where $\beta_{old} = \beta_{k-1}$ and $\beta_{new} = \beta_{k-1} + \epsilon_k$. Here, we chose ϵ to be normally distributed with mean 0 and variance 1.

For our simulations, we chose to sample the regressor values from a Uniform(0, 1). We simulated data using the parameters $\beta_0 = 1.5$ and $\beta_1 = -1$, as well as $\beta_0 = 3.5$ and $\beta_1 = -0.5$, with sample sizes $n = 50$, $n = 100$, and $n = 500$. We ran 20 simulations of each with 50,000 iterations and a burn-in period of 10,000 iterations. Table 2.2 shows the summary statistics of the posterior distribution for the parameter β. The summary statistics include the mean, median, mean squared error of the mean, mean squared error of the median, and a 95% credible interval.

n	β	Mean	Median	MSE Mean	MSE Median	95% CI
500	$\beta_0 = 1.5$	1.4602	1.4999	0.0706	0.0004	(1.4550, 1.4652)
	$\beta_1 = -1$	-0.9996	-1	0.0706	0.0005	(-1.0058, -0.9934)
100	$\beta_0 = 1.5$	1.4601	1.4998	0.0707	0.0020	(1.4486, 1.4716)
	$\beta_1 = -1$	-0.9995	-1	0.0714	0.0031	(-1.0135, -0.9854)
50	$\beta_0 = 1.5$	1.4603	1.4999	0.0707	0.0039	(1.4441, 1.4765)
	$\beta_1 = -1$	-0.9994	-1	0.0724	0.0060	(-1.0194, -0.9793)
500	$\beta_0 = 3.5$	3.4998	3.4999	0.0706	0.0003	(3.4561, 3.4738)
	$\beta_1 = -0.5$	-0.4997	-0.5	0.0707	0.0004	(-0.5052, -0.4945)
100	$\beta_0 = 3.5$	3.4601	3.4996	0.0707	0.0005	(3.4486, 3.4715)
	$\beta_1 = -0.5$	-0.5002	-0.5	0.0707	0.0006	(-0.5140, -0.4863)
50	$\beta_0 = 3.5$	3.4602	3.4997	0.0727	0.0032	(3.4440, 3.4766)
	$\beta_1 = -0.5$	-0.5012	-0.5	0.0732	0.0058	(-0.5204, -0.4994)

Table 2.2: Summary statistics of the posterior distribution for the parameter β using simulated data from the Yule-Simon distribution.

The results in Table 2.2 show the parameter estimates for β_0 and β_1 for varying sample sizes; $n = 50$, $n = 100$, and $n = 500$. We can see from these results that this algorithm is effective for both small and large sample sizes, as the mean and median are very close to the true parameter values. The mean squared error are also small suggesting the estimates are quite precise. Note that the credible intervals are also smaller for larger sample sizes, as expected. Next, we plot the posterior means and histograms to give a visualization of the results.

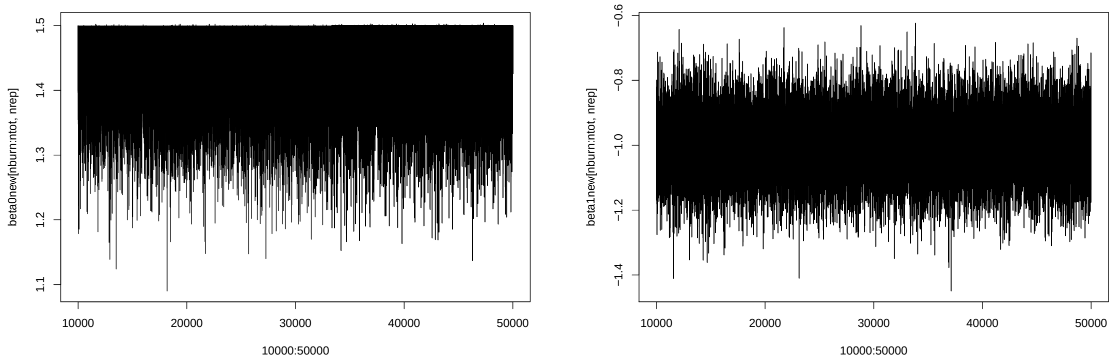

Figure 2.3: Posterior samples for simulated count data regression with $\beta_0 = 1.5$ (left) and $\beta_1 = -1$ (right) for a sample size of $n = 500$.

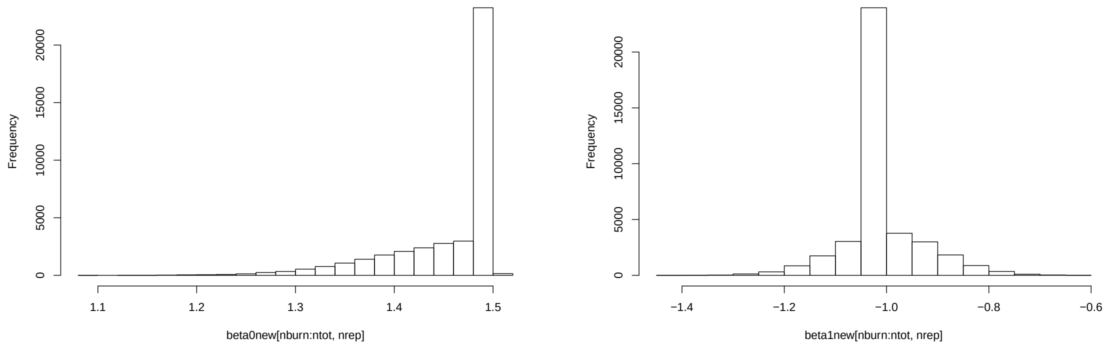

Figure 2.4: Posterior histograms for simulated count data regression with $\beta_0 = 1.5$ (left) and $\beta_1 = -1$ (right) for a sample size of $n = 500$.

In Figures 2.3 and 2.4, we show the posterior samples and histograms for β_0 and β_1 with sample size $n = 500$. The posterior samples show a good mixing of the chains centered around the true parameters, $\beta_0 = 1.5$ and $\beta_1 = -1$. Furthermore, the histograms show that the confidence intervals are relatively small.

2.4 Real Data Application

2.4.1 National Park Visits

Next, we look to apply some real data to the count data regression model from section 2.3. In the COUNT package in R, there are several data sets which are given with counts as the variable as interest. Upon examining these data sets, I found the data *loomis*, which is data taken from Loomis(2003). This data reports the frequency of national park visits in a year for individuals as reported in a survey. The data set is intended to use number of visits as the output and gives three categorical variables as inputs. These variables include gender, how long it took to travel to the park, and annual income. Figure 2.5 shows the distribution of the data.

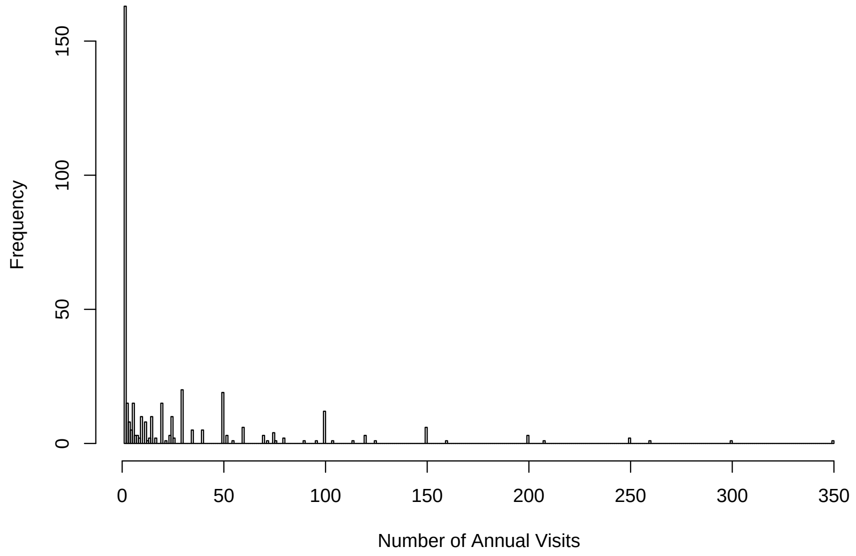

Figure 2.5: Counts of the number of times an individual has visited a National Park in a year from R's *loomis* data set.

From Figure 2.5, we can see that there are heavy tails, with many individuals reporting only visiting one national park in a year and a few individuals having visited a very large number of national parks in a year. Due the the heavy tails, we will fit a Yule-Simon to this data.

When analyzing the data set *loomis*, the variables of interest are the sample size, $n = 342$, and $\mathbf{k}$, the counts of national parks visited in a year. For this dataset, 5 simulations were run with 10,000 iterations and a burn-in period of 1,000 iterations.

Table 2.3 shows the summary statistics of the posterior distribution including the mean, median, mean squared error of the mean, mean squared error of the median, and 95% credible intervals. We can see from these results that the posterior means are very

n	β	Mean	Median	MSE Mean	MSE Median	95% CI
342	β_0=3.5	3.4608	3.5	0.0698	0	(3.4547, 3.4669)
	β_1=0.5	0.5001	0.5	0.0701	0	(0.4926, 0.5075)

Table 2.3: Summary statistics of the posterior distribution for the parameter β using R's loomis data set.

close to the true parameters and the median is equal to the true parameters. This implies that the algorithm is working correctly for the `loomis` data set. Next, we examine the posterior samples and posterior histograms shown below.

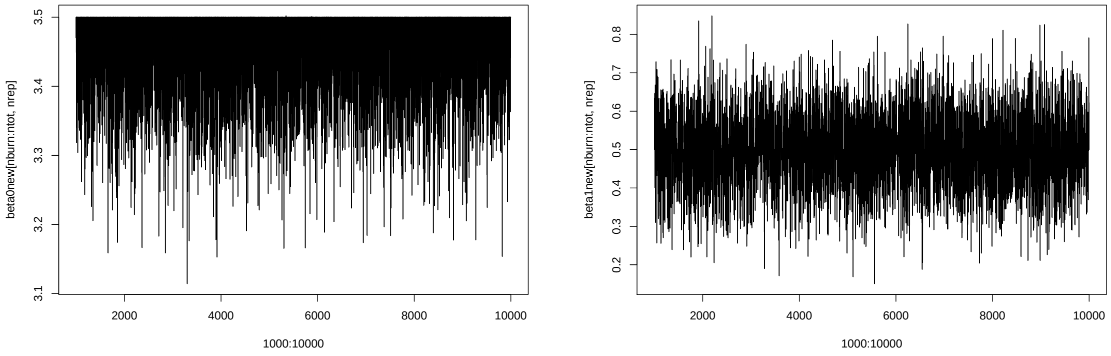

Figure 2.6: Posterior samples for the data set *loomis*.

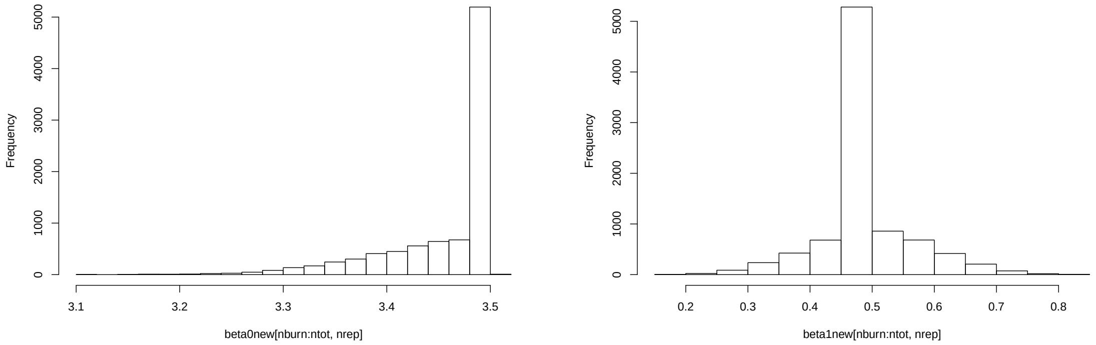

Figure 2.7: Posterior histograms for the data set *loomis*.

Figures 2.6 and 2.7 show the posterior samples and histograms for the *loomis* data set. The sample size is $n = 342$ using the variable `gender` as the regressor and the variable `anvisits` as the variable of interest. The initial value for $\beta_0 = 3.5$ and for $\beta_1 = 0.5$. We can see from the posterior sample and histogram that the algorithm has centered around the true parameters. There is a good mixing of the chain and the histogram has most values concentrated right at the true parameters. We have tested a few different starting values with similar results, suggesting that the algorithm is functioning appropriately and the Yule-Simon may potentially be a good fit for this data. To check if the Yule-Simon is a good fit, we could produce a sample from the posterior predictive distribution and compare this graphically to the distribution of our data.

2.4.2 Length of Stay in the Hospital

Another data set from R's COUNT package is called `azdrg112`. This data set gives the length of stay in a hospital for patients after having a CABG or PTCA heart procedure. The data comes from the 1995 Arizona Medicare data for DRG (Diagnostic Related Group) 112. The variable of interest is length of stay in the hospital, so this is represented as **k**. Figure 2.8 shows the distribution of the data. Again, we can see that the distribution of the data is J-shaped with many people only having to stay one night after their procedures, and some people having to stay a long time, up to 53 nights. The sample size of this data set is quite large, so the histogram bar above 50 can be difficult to see from Figure 2.8.

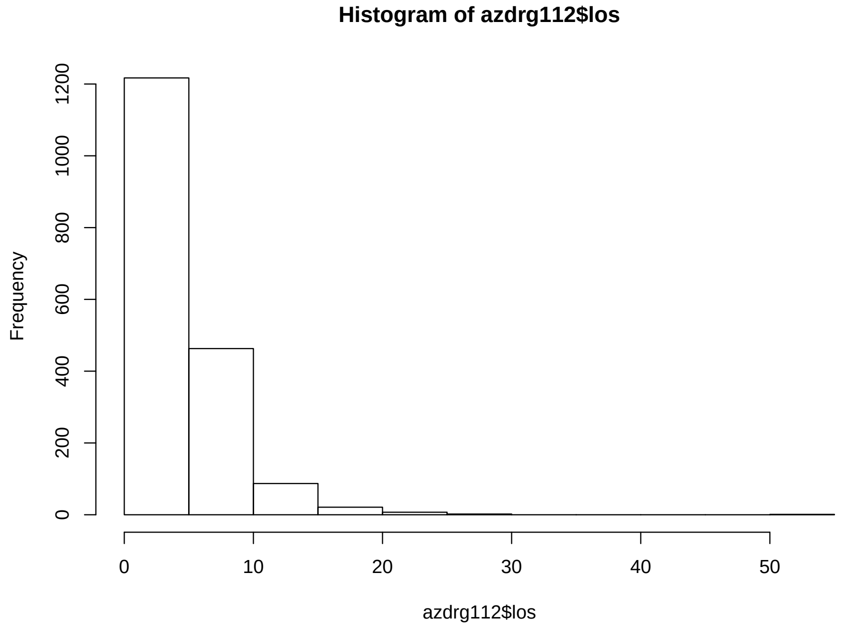

Figure 2.8: Length of stay in the hospital by individuals after having a heart procedure from R's azdrg112 data set.

The sample size for this data set is $n = 1798$. Here we used the variable `type1` as the regressor. The variable `type1` classifies the procedure as either emergency/urgent admission or elective admission. The following table show the results for 5 simulations of 10,000 iterations with a burn-in period of 1,000 iterations.

n	β	Mean	Median	MSE Mean	MSE Median	95% CI
342	β_0=1.5	1.4633	1.5	0.0705	0.0005	(1.4605, 1.4661)
	β_1=3	2.9843	3	0.0706	0.0005	(2.9811, 2.9875)

Table 2.4: Summary statistics of the posterior distribution for the parameter β using R's azdrg112 data set.

Table 2.4 shows the summary statistics of the posterior distribution including the mean, median, mean squared error of the mean, mean squared error of the median, and

95% credible intervals. The means and medians prove to be close to our true parameters, which is a sign the algorithm is doing what it is supposed to. We also ran other starting values for the parameters which also provided posterior means and medians close to the true parameters. Again, we plot the posterior samples and posterior histograms to check the chains have good mixing and are centered around our true parameters.

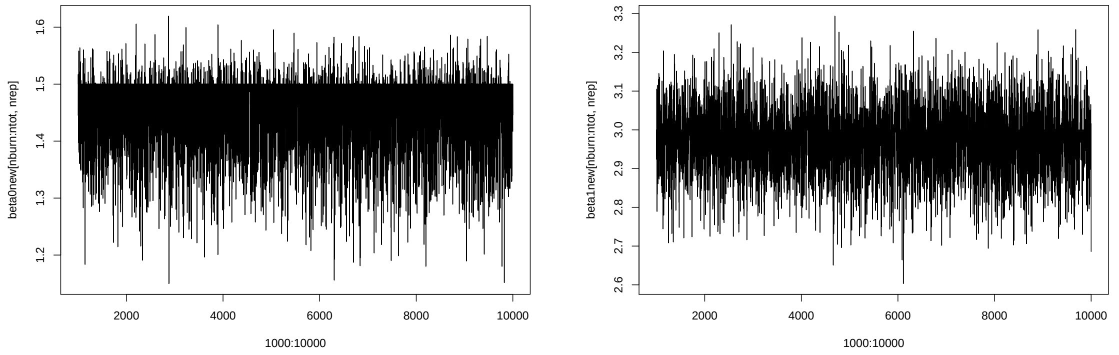

Figure 2.9: Posterior samples for the data set *azdrg112*.

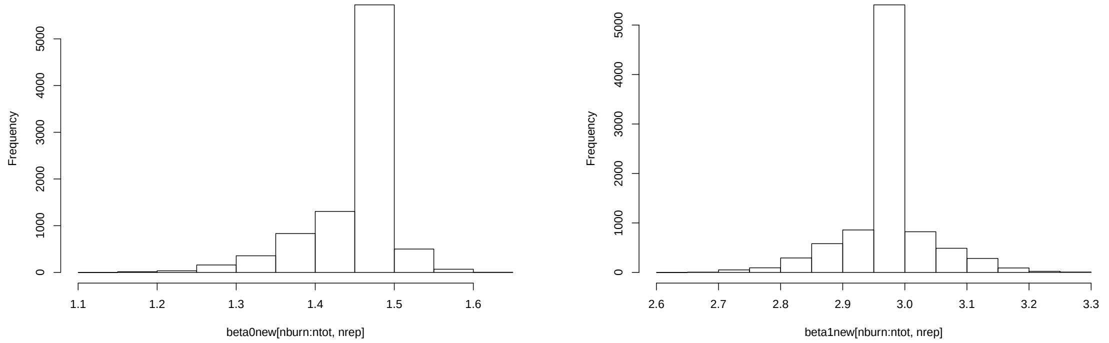

Figure 2.10: Posterior histograms for the data set *azdrg112*.

Figures 2.9 and 2.10 show the posterior samples and histograms for the `azdrg112` data set. The sample size is $n = 342$ using the variable `type1` as the regressor and the variable `los` as the variable of interest. For Figures 2.9 and 2.10 we chose the initial value as $\beta_0 = 1.5$ and as $\beta_1 = 3$. Figure 2.9 shows a good mixing of the chains centered around our true parameters. Since alternative starting values for β_0 and β_1 also provided close estimates, this shows that the algorithm is functioning correctly for the `azdrg112` data set. The posterior histogram shows that the values around $\beta_0 = 1.5$ and $\beta_1 = 3$ have the most frequencies and the variance seems to be relatively small. This suggests that the Yule-Simon distribution could also be a good fit for this data set. As previously explained, we could plot the samples from the posterior predictive distribution against the treu distribution of the data in order to check if the Yule-Simon distribution is actually a good fit for this data.

These application to real data could be new expansions into the scope of what the Yule-Simon distribution can model. As we have seen from previous literature, the applications come from a wide variety of topics. Throughout the review of current literature, we have not seen the length of stay in a hospital after a procedure and the number of National Parks visited in a year mentioned.

Chapter 3

A Two-Parameter Yule-Simon Distribution Extension

Next, we are interested in exploring a two-parameter Yule-Simon distribution. Recall that the Yule-Simon distribution's probability mass function can be written as follows:

$$f(k; \rho) = \int_0^\infty e^{-w}(1 - e^{-w})^{k-1} \rho e^{-\rho w} dw. \tag{3.1}$$

For equation (3.1), we let W be an exponentially distributed random variable with parameter ρ and let K be a Geometric distribution with probability of success equal to e^{-W}.

For the two-parameter representation, we propose the Negative Binomial distribution for the parameter K, rather than the Geometric distribution. The Yule-Simon is defined for positive integers rather than the natural numbers, so we use a version of the Negative Binomial which assumes positive integers and has probability of success equal to $1 - e^{-W}$. This form is crucial to define a data augmentation scheme for Bayesian analysis of the two-parameter Yule-Simon distribution.

$$P(K = k|W = w) = \binom{k + r - 2}{k - 1} e^{-rw}(1 - e^{-w})^{k-1} \qquad k = 1, 2, \ldots \tag{3.2}$$

In equation (3.2), if we set $r = 1$, we simply have the Geometric distribution, and therefore we would have the one-parameter Yule-Simon distribution as before. The two-parameter Yule-Simon distribution would then have the following probability mass function:

$$g(k; \rho) = \int_0^\infty \binom{k + r - 2}{k - 1} e^{-rw}(1 - e^{-w})^{k-1} \rho e^{-\rho w} dw. \tag{3.3}$$

If we make a change of variable $t = e^{-w} \Rightarrow w = -\ln(t) \Rightarrow dw = -dt/t$, we get the following:

$$g(k;\rho) = \binom{k+r-2}{k-1} \rho \, \mathrm{B}(k, \rho + r). \tag{3.4}$$

The likelihood of this model is then given by:

$$\begin{aligned} L(\mathbf{k}, \mathbf{w}, \rho, r) &= \prod_{i=1}^{n} \binom{k_i+r-2}{k_i-1} e^{-rw_i}(1-e^{-w_i})^{k_i-1} \rho e^{-\rho w_i} \\ &= \rho^n e^{-(\rho+r)\sum_{i=1}^{n} w_i} \prod_{i=1}^{n} \binom{k_i+r-2}{k_i-1} (1-e^{-w_i})^{k_i-1}. \end{aligned} \tag{3.5}$$

Using the priors $\pi(p) \sim \text{Gamma}(a, b)$, and $\pi(r) \sim \text{Geom}(\theta)$, we then get the following posterior distribution:

$$\begin{aligned} \pi(\rho, r, \mathbf{w}|\mathbf{k}) &\propto L(\mathbf{k}, \mathbf{w}, \rho, r)\pi(\rho)\pi(r) \\ &\propto \rho^n e^{-(\rho+r)\sum_{i=1}^{n} w_i} \left[\prod_{i=1}^{n} \binom{k_i+r-2}{k_i-1} (1-e^{-w_i})^{k_i-1} \right] \rho^{a-1} e^{-b\rho} \theta (1-\theta)^{r-1} \end{aligned} \tag{3.6}$$

Because the posterior distribution is not an explicit known distribution, we will need to employ MCMC methods. We must next find the full conditional distributions of our parameters. From the posterior distribution, we can get the full conditional distribution for w_i, which is given by:

$$\pi(w_i|w_{-i}, \rho, r, \boldsymbol{k}) \propto e^{-(\rho+r)w_i}(1-e^{-w_i})^{k_i-1}$$

Substituting $t_i = e^{-w_i}$, we get:

$$\begin{aligned} \pi(t_i|w_{-i}, \rho, r, \boldsymbol{k}) &\propto t^{\rho+r-1}(1-t)^{k_i-1} \\ &\sim \text{Beta}(\rho + r, k_i). \end{aligned}$$

We get the following for the full conditional distribution of ρ:

$$\begin{aligned} \pi(\rho|r, \boldsymbol{w}, \boldsymbol{k}) &\propto \rho^{a+n-1} e^{-\rho(b+\sum_{i=1}^{n} w_i)} \\ &\sim \text{Gamma}(a+n, b + \sum_{i=1}^{n} w_i). \end{aligned}$$

Finally, we get the following full conditional distribution for the parameter r:

$$\pi(r|\rho, \boldsymbol{w}, \boldsymbol{k}) \propto (1-\theta)^{r-1} e^{-r\sum_{i=1}^{n} w_i} \prod_{i=1}^{n} \binom{k_i + r - 2}{k_i - 1}.$$

We can now use a Metropolis within Gibbs in order to sample from the posterior distribution. Here, we use a random walk Metropolis Hastings and a Gibbs sampler as follows:

- sample $t_i \sim \text{Beta}(\rho + r - 1, k_i)$, for $i = 1, \ldots, n$;
- compute $w_i = -log(t_i)$, for $i = 1, \ldots, n$;
- sample $\rho | r, \boldsymbol{w}, \boldsymbol{k} \sim \text{Gamma}(a + n, b + \sum_{i=1}^{n} w_i)$;
- sample $r|\rho, \boldsymbol{w}, \boldsymbol{k}$ from the Metropolis Hastings algorithm.

Sampling ρ is pretty straightforward, however, for r we need to sample using the Metropolis Hastings algorithm. The Metropolis Hastings algorithm uses an acceptance ratio. Here we will use the log of the acceptance ratio as follows:

$$\begin{aligned}\log(\alpha) = &\left[(r_{new} - 1)\log(1-\theta) - r_{new}\sum_{i=1}^{n} w_i + \sum_{i=1}^{n} \log\binom{k_i + r_{new} - 2}{k_i - 1}\right] \\ &- \left[(r-1)\log(1-\theta) - r\sum_{i=1}^{n} w_i + \sum_{i=1}^{n} \log\binom{k_i + r - 2}{k_i - 1}\right] + \log(4/3),\end{aligned}$$

where r represents r_{j-1}. We know r_{new} must be discrete because we use r_{new} in the combinatoric term. We chose r_{new} to be a Geometric random variable with probability of success equal to $3/4$.

Now that we have derived an algorithm for sampling from the posterior distribution, we can use simulated data to test the algorithm. The next section of this paper will generate simulated data to test the sampling methods from the two-parameter Yule-Simon distribution.

3.1 Simulation Study

For the following simulation study, we ran one simulation of 50,000 iterations with a burn-in period of 10,000 iterations. In Table 3.1, we report some summary statistics including the mean, median, mean squared error for the mean, and mean squared error for the median. For the simulations, we chose the parameters to be $\rho = 4$ and $r = 3$ for the first and $\rho = 2$ and $r = 2$ for the second, with sample sizes $n = 50$, $n = 100$, and $n = 500$ in both cases.

Figure 3.1 and 3.2 show good mixing of the chains while exploring the parameter space, with the chains centered around the true parameters. The posterior histograms

n	Parameter	Mean	Median	MSE Mean	MSE Median
500	$\rho = 4$	4.0701	4.0618	0.0737	0.0322
	$r = 3$	3	3	0	0
100	$\rho = 4$	4.4521	4.1753	2.8041	0.8463
	$r = 3$	2.7113	3	1.2051	1
50	$\rho = 4$	2.6813	2.4392	3.1156	2.7032
	$r = 3$	1.8554	2	2.2435	1
500	$\rho = 2$	2.3669	2.3629	0.1550	0.1317
	$r = 2$	2	2	0	0
100	$\rho = 2$	2.6871	2.5972	1.1042	0.4413
	$r = 2$	2.4027	2	0.8432	1
50	$\rho = 2$	2.2594	2.1735	0.6156	0.2243
	$r = 2$	2.3554	2	0.8402	1

Table 3.1: Summary statistics of the posterior distribution with varying values of the parameters ρ and r for sample sizes $n = 50$, $n = 100$, and $n = 500$ using simulated data from the two-parameter Yule-Simon distribution.

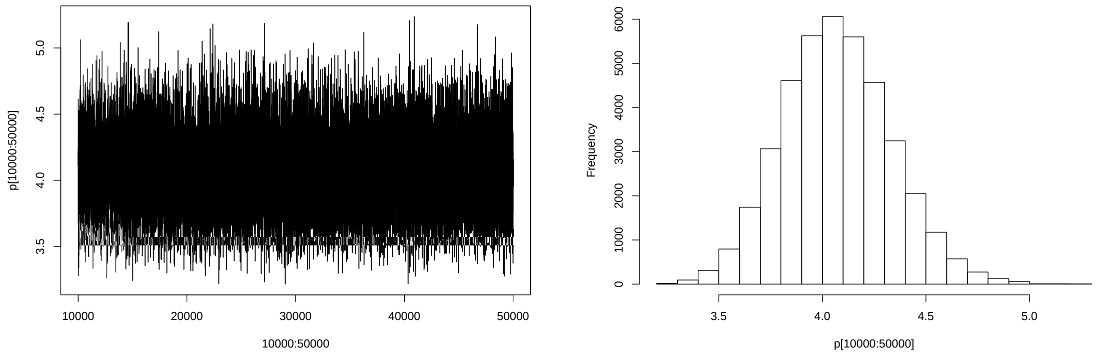

Figure 3.1: Posterior sample and histogram for the parameter $\rho = 4$ and when $r = 3$ for sample size $n = 500$.

show a wider variance than those for the one-parameter Yule-Simon distribution. The posterior histograms are, however, centered around the true parameter. It appears as though this algorithm is most efficient for larger sample sizes. When running the algorithm with smaller sample sizes, the estimates were not as accurate. This could mean that the proposal distribution needs to be reexamined.

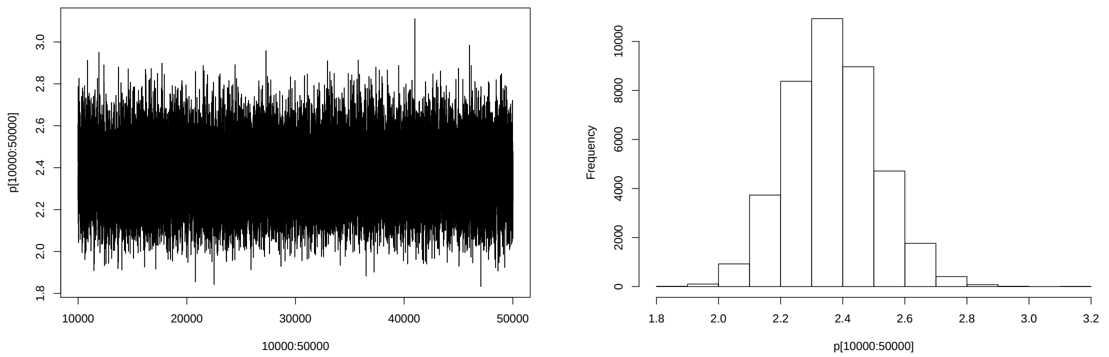

Figure 3.2: Posterior sample and histogram for the parameter $\rho = 2$ and when $r = 2$ for sample size $n = 500$.

Chapter 4

Discussion

The Yule-Simon distribution has many applications and has quite a bit of current literature. This paper aimed to review previous literature on Yule-Simon distribution, as well as extend that literature to include a two-parameter Yule-Simon distribution. In chapter 2, we introduced the Yule-Simon distribution and some preliminary information in order to take a Bayesian approach to the distribution. We then proceeded to simulate data to test that the proposed random walk Metropolis-Hastings algorithm was returning plausible results. Various graphical displays, such as posterior samples and posterior histograms were given in order to illustrate the results of our algorithms. We also tested an algorithm for count data regression. Applications to real-life contexts, such as number of National Parks visited in a year and length of stay in a hospital after a surgery, were made. We noted that the Yule-Simon distribution is a special case of the Waring distribution, which inspired the extension to a two-parameter Yule-Simon distribution. We derived an algorithm for the two-parameter Yule-Simon distribution and tested this on simulated data.

Future work to be done would include finding real data to apply to the two-parameter Yule-Simon distribution extension. It also seems as though the proposal distribution for the two-parameter Yule-Simon distribution could use some revision. We could also find more real data application to be applied to the one-parameter Yule-Simon distribution. Overall, the algorithms used for the one-parameter Yule-Simon distribution seem to be sound. They produce accurate and precise estimates of the parameters.

Acknowledgments

Foremost, I would like to thank my supervisor, Dr. Fabrizio Leisen for his support and guidance throughout my project.

Bibliography

[1] Gallardo, D. I., Gomex, H. W., Bolfarine, H., 2016. A new cure rate model based on the Yule-Simon distribution with application to a melanoma data set. Journal of Applied Statistics.

[2] Garcia Garcia, J. M., 2011. A fixed-point algorithm to estimate the yule-simon distribution parameter. Applied Mathematics and Computation 217, 85608566.

[3] Handcock, M. S., Morris, M., 2007. A simple model for complex networks with arbitrary degree distribution and clustering. Lecture Notes in Computer Science, 4503, 103.

[4] Johnson, N., Kotz, S., Kemp, A., 1992. Univariate discrete distributions. 2nd edn. Wiley series in probability and mathematical statistics.

[5] Leisen, F., Rossini, L., Villa, C., 2016. A note on the posterior inference for the Yule-Simon distribution. Journal of Statistical Computation and Simulation Volume 87, 2017 - Issue 6.

[6] Leisen, F., Rossini, L., Villa, C., 2016. Objective Bayesian Analysis of the YuleSimon Distribution with Applications. http://arxiv.org/abs/1604.05661.

[7] Simon, H. A., 1955. On a class of skew distribution functions. Biometrika 42, 425440.

[8] van Dyk, D., Meng, X. L., 2001. The art of data augmentation (with discussion). Journal of Computational and Graphical Statistics, 10, 1-111.

[9] Yule, G. U., 1925. A Mathematical theory of evolution, based on the conclusion of Dr. J.C. Willis. Philosophical Transactions of the Royal Society of London, Series B 213, 2187.

Appendix A: Simulation Code

This code simulates data from the Yule-Simon distribution. The true parameter is `rho` with number of Metropolis Hastings represented by `m` and sample size `n`.

```
m=50000
n=50
rho=4
a=.25
b=.05

W=rexp(n,rho)
k=rgeom(n,exp(-W))
k=k+1
t=vector(length=n)
p[1]=1

for (j in (2:m)){
  for (i in (1:n)) {
     t[i]=rbeta(1, p[j-1]+1, k[i])
  }
  w=-log(t)
  p[j]=rgamma(1, a+n, b+sum(w))
  j=j+1
}

mean(p[10000:50000])
median(p[10000:50000])
mean((p[10000:50000]-rho)^2)
median((p[10000:50000]-rho)^2)

plot(10000:50000, p[10000:50000], type='l')
hist(p[10000:50000], xlim= c(0,3))
```

Appendix B: Count Data Regression Code

This code simulates data for count data regression. The true parameters are `beta0` and `beta1` with sample size `n` number of simulations represented by `nrep` and number of Metropolis Hastings by `nMH`. The number of Metropolis Hastings should stay set at 2. The variables `nsave, nburn,` and `ntot` determine the number of iterations and the burn-in period.

```
library("optimbase", lib.loc="~/R/win-library/3.3")

n=500
mu_beta=c(0, 0)
sigma_beta=1
beta0=3.5
beta1=.5

x2=runif(n)
x2=transpose(x2)
rho=exp(beta0+beta1*x2)

w=rexp(n, rho)
x=rgeom(n, exp(-w))
w=transpose(w)
x=transpose(x)
y=x+1



nsave=9000
nburn=1000
ntot=nburn+nsave

nMH=2
```

```
nrep=5

wold=matrix(w, ncol=ntot, nrow=n)
told=matrix(exp(-w), ncol=ntot, nrow=n)
beta0old=matrix(beta0, ncol=1, nrow=nMH)
beta1old=matrix(beta1, ncol=1, nrow=nMH)
beta0new=matrix(beta0, ncol=nrep, nrow=ntot+1)
beta1new=matrix(beta1, ncol=nrep, nrow=ntot+1)

mean0=0
mean1=0
med0=0
med1=0
CIU0=0
CIL0=0
CIU1=0
CIL1=0
mse0=0
medse0=0
mse1=0
medse1=0

for (rep in 1:nrep){
for (iter in 1:ntot){
  for (j in 1:n){
    told[j,iter]=rbeta(1, exp(beta0new[iter, 1]+beta1new[iter, 1]*x2[j,])+1, y)
    wold[j,iter]=-log(told[j,iter])
  }
  for (i in 2:nMH){
    mustar_beta=c(beta0old[i-1,1], beta1old[i-1,1])
    tau=0.1
    beta0star=rnorm(1,mustar_beta[1],tau)
    beta1star=rnorm(1,mustar_beta[2],tau)
    betastar=c(beta0star, beta1star)
    betastar=transpose(betastar)

    postar=matrix(0, ncol=1, nrow=n)
    for (kk in 1:n){
      postar[kk,1]=exp(beta0star+beta1star*x2[kk,]*wold[kk,iter])
    }
    sustar=-sum(postar)
    lognum2=sustar-log(2*pi)+n*beta0star+beta1star*sum(x2)-.5*(beta0star^2+beta1star^2)/s

    poold=matrix(0, ncol=1, nrow=n)
    for (kk1 in 1:n){
      poold[kk1,1]=exp(beta0old[i-1,1]+beta1old[i-1,1]*x2[kk1,]*wold[kk1,iter])
    }
    suold=-sum(poold)
    logden2=suold-log(2*pi)+n*beta0old[i-1,1]+beta1old[i-1,1]*sum(x2)-.5*(beta0old[i-1,1]

    minbe=lognum2-logden2
    u=runif(1)

    if (log(u)<minbe){
      beta0old[i,1]=beta0star
      beta1old[i,1]=beta1star
    }else{
      beta0old[i,1]=beta0old[i-1,1]
      beta1old[i,1]=beta1old[i-1,1]
    }
  }
  beta0new[iter+1,rep]=beta0old[i,1]
  beta1new[iter+1,rep]=beta1old[i,1]

}
  mean0= mean0+mean(beta0new[nburn:ntot,rep])
  mean1= mean1+mean(beta1new[nburn:ntot,rep])
  med0=med0+median(beta0new[nburn:ntot,rep])
  med1=med1+median(beta1new[nburn:ntot,rep])

  CIU0=CIU0+mean(beta0new[nburn:ntot,rep])+1.96*sqrt(var(beta0new[nburn:ntot,rep])/n)
  CIL0=CIL0+mean(beta0new[nburn:ntot,rep])-1.96*sqrt(var(beta0new[nburn:ntot,rep])/n)
  CIU1=CIU1+mean(beta1new[nburn:ntot,rep])+1.96*sqrt(var(beta1new[nburn:ntot,rep])/n)
  CIL1=CIL1+mean(beta1new[nburn:ntot,rep])-1.96*sqrt(var(beta1new[nburn:ntot,rep])/n)

  mse0=mse0+sqrt(mean((beta0new[nburn:ntot,rep]-beta0)^2))
  medse0=medse0+sqrt(median((beta0new[nburn:ntot,rep]-beta0)^2))
  mse1=mse1+sqrt(mean((beta1new[nburn:ntot,rep]-beta1)^2))
  medse1=medse1+sqrt(median((beta1new[nburn:ntot,rep]-beta1)^2))
}
```

```
mean0/nrep
mean1/nrep
med0/nrep
med1/nrep
mse0/nrep
medse0/nrep
mse1/nrep
medse1/nrep
CIU0/nrep
CIL0/nrep
CIU1/nrep
CIL1/nrep

plot(1000:10000, beta0new[nburn:ntot,nrep], type='l')
plot(1000:10000, beta1new[nburn:ntot,nrep], type='l')
hist(beta0new[nburn:ntot,nrep], main=NULL)
hist(beta1new[nburn:ntot,nrep], main=NULL)
```

Appendix C: Two-Parameter Yule-Simon Code

This code simulates data from the two-parameter Yule-Simon distribution. The true parameters are `rho` and `r` with number of simulations represented by `sim` and number of Metropolis Hastings by `m`.

```
sim=1
m=50000
n=500
rho=2
r=2
a=.25
b=.05
theta=.5

W=rexp(n,rho)
lambda=rgamma(n, r, exp(-W)/(1-exp(-W)))
k=rpois(n, lambda)
#k=rgeom(n,exp(-W))
k=k+1

for (l in 1:sim){
  t=vector(length=n)
  p=vector(length=m)
  rnew=as.vector(r)
  A=vector(length=m)
  p[1]=rho

  for (j in (2:m)){
    t=rbeta(n, p[j-1]+r[j-1], k)
    w=-log(t)
    p[j]=rgamma(1, a+n, b+sum(w))

    rnew[j]=rgeom(1,3/4)
    sumw=0
    sumr=0
    sumrnew=0
    for (i in 1:n){
      sumw=sumw+w[i]
      sumr=sumr+log(choose(k[i]+r[j-1]-2, k[i]-1))
      sumrnew=sumrnew+log(choose(k[i]+rnew[j]-2, k[i]-1))
    }

    A_num=((rnew[j]-1)*log(1-theta)-(rnew[j]*sumw)+sumrnew-r[j-1])
    A_denom=((r[j-1]-1)*log(1-theta)-(r[j-1]*sumw)+sumr-rnew[j])
    A[j]=A_num-A_denom+log(4/3)

    u=runif(1, 0, 1)

    if (log(u) <= A[j]){
      r[j]=rnew[j]
    } else{
      r[j]=r[j-1]
    }
  }
}


mean(p[10000:50000])
mean(r[10000:50000])
median(p[10000:50000])
median(r[10000:50000])
mean((p[10000:50000]-rho)^2)
mean((r[10000:50000]-r[1])^2)
plot(10000:50000, p[10000:50000], type=’l’)
hist(p[10000:50000], main=NULL)
```